Alma Mater Studiorum · Università Di Bologna

Campus Di Cesena

SCUOLA DI INGEGNERIA E ARCHITETTURA
Corso di Laurea in Ingegneria Elettronica, Informatica e Telecomunicazioni

Tesi di Laurea in Sistemi Distribuiti

SICUREZZA, PROTEZIONE E INTEGRITÀ DEI SISTEMI CLOUD: MODELLI, METODI E TECNOLOGIE

Relatore:
Chiar.mo Prof.
ANDREA OMICINI

Correlatore:
Ing. STEFANO MARIANI

Presentata da:
FILIPPO ALBERTO BRANDOLINI

Sessione II
Anno Accademico 2013/2014

Indice

Elenco delle figure

Elenco delle tabelle

Elenco delle abbreviazioni

API	Application Programming Interface
DaaS	Data as a Service
DDoS	Distributed Denial of Service
DoS	Denial of Service
HaaS	Hardware as a Service
IaaS	Infrastructure as a Service
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
MITM	Man In The Middle
PaaS	Platform as a Service
SaaS	Software as a Service
SLA	Service Level Agreements
SSTP	Secure Socket Tunneling Protocol
STaaS	Storage as a Service
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network

Abstract

La natura distribuita del Cloud Computing, che comporta un'elevata condivisione delle risorse e una moltitudine di accessi ai sistemi informatici, permette agli intrusi di sfruttare questa tecnologia a scopi malevoli. Per contrastare le intrusioni e gli attacchi ai dati sensibili degli utenti, vengono implementati sistemi di rilevamento delle intrusioni e metodi di difesa in ambiente virtualizzato, allo scopo di garantire una sicurezza globale fondata sia sul concetto di prevenzione, sia su quello di cura: un efficace sistema di sicurezza deve infatti rilevare eventuali intrusioni e pericoli imminenti, fornendo una prima fase difensiva a priori, e, al contempo, evitare fallimenti totali, pur avendo subito danni, e mantenere alta la qualità del servizio, garantendo una seconda fase difensiva, a posteriori. Questa tesi illustra i molteplici metodi di funzionamento degli attacchi distribuiti e dell'hacking malevolo, con particolare riferimento ai pericoli di ultima generazione, e definisce le principali strategie e tecniche atte a garantire sicurezza, protezione e integrità dei dati all'interno di un sistema Cloud.

Keywords: cloud computing, security, integrity, hacking, threats, denial of service, intrusion detection, resiliency.

Capitolo 1

Introduzione al Cloud Computing

"Cloud computing is a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet" [1]. Il Cloud Computing, in italiano "nuvola informatica", è dunque un insieme di tecnologie che permettono l'utilizzo di risorse hardware e software distribuite per la memorizzazione e l'elaborazione di dati. Solitamente si presenta sotto forma di servizi offerti da un provider ai clienti, e ha segnato l'avvento di un'era tecnologica in cui gli utenti, pur non possedendo un computer, hanno accesso a componenti hardware e software gestiti da terzi [2]. In ambiente Cloud, le risorse condivise vengono tipicamente virtualizzate[1] in rete

[1]La virtualizzazione indica la possibilità di effettuare un'astrazione delle componenti hardware di un elaboratore, e realizzare un emulatore sul quale si è in grado di installare sistemi operativi e software, come se questi fosse un elaboratore fisico vero e proprio. L'insieme delle componenti hardware virtualizzate è denominato macchina virtuale.
Il principale vantaggio della virtualizzazione risiede nella condivisione delle risorse hardware, la quale rende possibile l'attivazione e il funzionamento, in contemporanea, di più sistemi operativi che sfruttano le medesime risorse (da qui emerge la necessità di software gestionali che amministrino le code di assegnazione e le eventuali contese per l'accesso a ta-

seguendo un'architettura client-server.

Ma come si è arrivati all'idea di Cloud Computing? [4] L'idea di informatica come servizio venne discussa per la prima volta negli anni '60 da McCarthy, e il concetto venne approfondito da Parkhill, che esaminò la natura dei servizi quali acqua, gas ed energia elettrica, per arrivare a comprendere, per analogia, le caratteristiche che l'informatica avrebbe dovuto avere se fosse stata anch'essa un servizio. Consideriamo, ad esempio, la fornitura di energia elettrica: per averne accesso, connettiamo i nostri dispositivi ad apposite prese elettriche, e paghiamo in base a quanta energia utilizziamo. Nei periodi estivi, le ore del giorno sono più lunghe di quelle notturne, pertanto abbiamo meno necessità di utilizzare energia elettrica per l'illuminazione artificiale o il riscaldamento. Al contrario, durante l'inverno, utilizziamo maggiormente illuminazione, riscaldamento e acqua calda, e ci aspettiamo che le nostre bollette riflettano il maggiore utilizzo di tali servizi. Il concetto vale anche per l'informatica come servizio: grazie al Cloud Computing, risorse hardware e software sono disponibili ovunque vi sia la possibilità di accedere a una rete, ed è possibile pagare solo ciò che si utilizza.

1.1 Servizi

Si possono distinguere alcune principali tipologie di servizi Cloud:

- *SaaS* (Software as a Service): fornisce accesso a risorse software localizzate su infrastrutture remote. Il cliente non acquista alcuna licenza specifica: il costo del software e dell'infrastruttura sono raggruppati in un singolo addebito proporzionale all'utilizzo [5].

li risorse). Nell'ambito di reti, la virtualizzazione permette di realizzare le VLAN (Virtual Local Area Network), gruppi di più reti locali configurate in modo da poter comunicare come se fossero connesse tramite gli stessi cavi, quando in realtà sono situate su differenti segmenti del dominio di rete [3]. L'utilizzo di reti virtuali, basato su connessioni logiche, anziché fisiche, è solitamente legato alla necessità di separare il traffico di rete dei dipartimenti aziendali, in modo tale da poter applicare, fra le altre cose, politiche di sicurezza informatica.

- *DaaS* (Data as a Service): fornisce accesso a dati memorizzati su dispositivi remoti. L'utente può accedervi, spesso attraverso autenticazione, come se fossero residenti sul suo disco locale.

- *STaaS* (Storage as a Service): permette all'utente di effettuare l'upload dei propri dati su dispositivi remoti, consentendo il backup di informazioni sensibili e la relativa sincronizzazione su più dispositivi.

- *HaaS* (Hardware as a Service): fornisce accesso a risorse hardware remote. L'utente invia i suoi dati a un computer remoto, e questi elabora i dati ricevuti e restituisce all'utente il risultato [6].

- *IaaS* (Infrastructure as a Service): fornisce accesso a veri e propri centri di dati remoti, ossia intere infrastrutture hardware. Un centro di dati è caratterizzato da un edificio sicuro in cui si trovano risorse hardware, con garanzie di fornitura elettrica ininterrotta per assicurare un funzionamento del servizio 24 ore su 24, sistemi di refrigerazione per evitare il riscaldamento dei server, controllo degli accessi e altri servizi dedicati. In questo caso, la virtualizzazione svolge un ruolo determinante, in quanto è probabile che i clienti utilizzino applicazioni di differenti sistemi operativi nell'accesso all'infrastruttura del provider, ed è pertanto necessaria la presenza di più macchine virtuali, gestite da un software definito hypervisor [2].

- *PaaS* (Platform as a Service): è un'implementazione del servizio IaaS, in cui il provider fornisce, oltre all'infrastruttura hardware, anche quella software (sistema operativo, applicazioni, strumenti di sviluppo dedicati) [2].

I servizi cloud possono anche essere categorizzati in base alla modalità di accesso, e in questo caso si suddividono in [2]:

- *Public Cloud*: l'infrastruttura informatica è pubblica, pertanto chiunque, gratuitamente o a pagamento, a seconda dei casi, può usufruire dei servizi offerti dal provider.

- *Private Cloud*: l'infrastruttura informatica è ad uso esclusivo di una singola azienda.

- *Community Cloud*: l'infrastruttura informatica è ad uso esclusivo di una specifica comunità di utenti che condividono interessi. Ad esempio, più università possono decidere di cooperare e interconnettere le proprie infrastrutture informatiche per creare un ambiente virtuale condiviso fra i propri membri.

- *Hybrid Cloud*: l'infrastruttura informatica è regolata da una combinazione di più modalità d'accesso. Ad esempio, un'azienda può decidere di tenere private alcune parti dell'infrastruttura e dei servizi software, e renderne disponibili altre in modo pubblico. O ancora, può capitare che un'azienda, pur utilizzando una private Cloud, debba affrontare un improvviso aumento del carico di rete, e decida di utilizzare una public Cloud per gestire il carico extra (questo scenario prende il nome di Cloud Bursting).

1.2 Vantaggi e svantaggi

Analizziamo ora pro e contro dell'utilizzo del Cloud Computing.
Vantaggi [2]:

- Realizzazione del concetto "pay for what you use" e conseguente riduzione delle spese aziendali, dato che non si investe né in grandi infrastrutture informatiche né in rimpiazzi per i computer divenuti obsoleti. Questo è sia utile alle compagnie in via di sviluppo, che non devono necessariamente investire grandi quantità di denaro in infrastrutture informatiche senza aver ancora realizzato i primi guadagni, sia alle grandi multinazionali, che possono massimizzare i profitti evitando anche il minimo spreco di capitale e risorse.

- Elevata scalabilità: possibilità di incrementare (o ridurre, a seconda delle necessità) le prestazioni del sistema, grazie all'introduzione di nuove risorse.

- Qualità del Servizio assicurata in base ai Service Level Agreements (SLA), accordi attraverso i quali si definiscono gli obblighi contrattuali del provider nei confronti dei suoi clienti, nell'erogazione dei servizi.

- Possibilità di archiviare dati aziendali in un'infrastruttura Cloud dedicata, in modo da evitare l'appesantimento della rete principale con dati a bassa frequenza d'utilizzo.

Svantaggi [2]:

- L'operatività di un servizio Cloud si basa su una comunicazione efficace e soprattutto ininterrotta con l'infrastruttura del provider: un eventuale guasto nelle comunicazioni causa la terminazione del servizio. Può risultare rischioso, pertanto, riporre l'intera funzionalità del proprio sistema aziendale nelle mani di un provider esterno. I principali metodi per rimediare a questo problema consistono nella predisposizione di un secondo canale di comunicazione con il provider – indipendente dal primo – da utilizzare in caso di necessità, o nel mantenimento di una parte del sistema informatico all'interno all'azienda.

- E' possibile, anzi, probabile, che il traffico dati di un cliente venga trasportato in rete assieme a quello di altri clienti. Ciò significa che pacchetti di dati scambiati attraverso punti di accesso sicuri potrebbero mescolarsi con pacchetti inviati secondo schemi di comunicazione poco affidabili, o persino appartenenti ad un competitore che ha scelto lo stesso Cloud provider, il quale, pur applicando una separazione dei traffici tramite reti virtuali, utilizza comunque gli stessi cavi fisici di connessione. La separazione, infatti, è soltanto virtuale: a livello fisico, le informazioni degli utenti sono mischiate [4]. Pertanto, ogni volta che

informazioni o programmi vengono inviati a una rete Cloud, c'è il rischio che siano intercettati da terzi. Per limitare i danni nel caso in cui si verifichi questa eventualità, è essenziale applicare una solida crittografia ai dati inviati in rete, ottenendo tuttavia un'inevitabile riduzione delle performance del sistema.

- La qualità del servizio offerto dal provider si può deteriorare nel tempo, e non si può escludere la possibilità che il provider cessi la propria attività a causa di fallimento (banca rotta). Le aziende che utilizzano i servizi Cloud devono avere un piano di riserva che permetta di trasferire rapidamente i propri dati e le proprie applicazioni ad un altro provider, in caso di necessità. La difficoltà principale che scaturisce da questo scenario è la mancanza di standardizzazione dei servizi Cloud offerti dai provider. A questo proposito, è opportuno che le aziende effettuino una ricerca approfondita fra i vari provider disponibili, allo scopo di trovare una coppia di provider che seguano standard simili, in modo tale da evitare un eventuale dispendioso lavoro di portabilità del sistema.

- Possono sorgere problemi legali nel caso in cui i server del provider siano localizzati in una nazione straniera, e dati o applicazioni dell'azienda risultino corrotti o rubati. E' opportuno prevedere questo scenario e specificare, negli accordi sul servizio con il provider, quali leggi applicare in casi simili, fra quelle del paese in cui ha siede l'azienda che offre il servizio, e quelle del paese in cui si trovano i suoi server.

- Un ultimo problema, sorto di recente, è la sorveglianza clandestina del traffico di rete da parte dei servizi segreti. Se l'infrastruttura del provider è suddivisa in più aree, localizzate in diversi paesi, il suo traffico dati potrebbe finire nel mirino delle Intelligence, e in tal caso i dati aziendali diventerebbero oggetto di indagini, benché questo comporti una violazione della privacy, poiché inoltrare comunicazioni di rete a più nodi sparsi per il globo può rappresentare un tentativo, da parte di un

hacker[2], di evitare il suo rintracciamento, nell'ambito di frodi o attività terroristiche. Persino un'efficace crittografia può essere bypassata, se si possiede un'enorme quantità di risorse hardware in grado di minimizzare i tempi di calcolo computazionale. Si può evitare questo problema preferendo provider residenti nel proprio paese o che comunque non si avvalgano di server localizzati all'estero.

Sulla base delle considerazioni fatte, le principali applicazioni adatte ad essere trasferite su un sistema cloud sono [2]:

- Applicazioni che richiedono una performance computazionale più elevata di quella a disposizione, o che richiedono un software specifico non disponibile all'interno dell'azienda. Ipotizziamo di essere alla guida di una software house di piccole dimensioni, che ha investito una certa somma di denaro in componentistica, a seguito di una stima del lavoro medio previsto, e supponiamo di dover far fronte a una richiesta di un cliente che non può essere soddisfatta poiché necessita di un software non disponibile in azienda. Generalmente, le opzioni decisionali sarebbero due, ovvero: investire un'ulteriore somma di denaro per l'acquisto del software necessario, nel caso in cui si valuti che possano verificarsi altre richieste simili in futuro, e che si possa in tal modo rientrare nell'investimento, oppure rifiutare la richiesta del cliente, se si valuta che la richiesta sia così singolare da non permettere un recupero della somma investita. La realtà Cloud offre all'azienda una terza opzione, ovvero quella di sfruttare un servizio SaaS di un provider e disporre così del programma extra per il solo periodo di tempo necessario a svolgere

[2]Per la precisione, quando ci si riferisce a criminali informatici e ad autori di attacchi distribuiti, sarebbe più opportuno parlare di "cracker", dato che il termine hacking indica l'insieme delle attività atte a conoscere e modificare un sistema informatico. Un esperto di sicurezza, pagato da un'impresa per testare l'affidabilità del suo sistema, può essere infatti chiamato hacker. Tuttavia, nel corso di questo lavoro, tranne che nel capitolo 2.2, in cui illustrerò le tipologie di criminali informatici, e sarò pertanto il più specifico possibile, utilizzerò per comodità il termine hacker per riferirmi all'autore di un attacco informatico, indipendentemente dalle cause che lo hanno spinto a realizzarlo.

il lavoro e soddisfare la richiesta del cliente, evitando così di investire in modo definitivo in nuove risorse di incerta utilità futura.

- Applicazioni di calcolo parallelo, che consistono prevalentemente nell'esecuzione di uno stesso programma, in modo concorrente, su un elevato numero di processori, allo scopo di aumentare la velocità computazionale. Questa strategia altro non è che l'applicazione della programmazione multiprocesso in ambito Cloud. Un classico esempio è la ricerca di siti web nei quali si riscontra uno specifico set di parole. Per ottimizzare le performance del programma, si può suddividere l'universo del world wide web in un certo numero di sottoinsiemi, e assegnare ogni sottoinsieme a un'istanza del programma. Ognuna di queste segna gli URL dei siti nei quali si riscontrano le parole chiave, e quando tutte le istanze hanno terminato la propria ricerca, i risultati singoli vengono combinati per fornire in output la totalità dei siti web che corrispondono ai criteri richiesti, permettendo di ottenere un aumento delle performance direttamente proporzionale al numero di istanze del programma in esecuzione su processori diversi, e una conseguente riduzione del tempo di risposta ad esso inversamente proporzionale. Il software Hadoop MapReduce [7] [8] è stato sviluppato allo scopo di permettere la creazione di applicazioni Cloud in grado di processare petabyte di dati (1 PB = 10^{15} byte) in modo parallelo e affidabile, garantendo tolleranza agli errori[3] e applicando la strategia di cui sopra.

- Applicazioni stagionali che richiedono un'elevata e rapida capacità computazionale per un breve periodo di tempo. Ad esempio, le compagnie di e-commerce hanno sicuramente un traffico dati assai più elevato nei periodi natalizi; la stessa Amazon, una delle maggiori compagnie

[3]Tolleranza agli errori (fault-tolerance): è uno degli aspetti fondamentali del concetto di affidabilità in ambito informatico. Rappresenta la capacità di un sistema di non subire avarie (interruzioni di servizio) anche in presenza di guasti: può accadere che un singolo componente subisca danni a livello hardware o software, mentre il resto del sistema continui a funzionare correttamente.

del settore a livello mondiale, registrava inizialmente periodi in cui la propria infrastruttura informatica era adeguatamente sfruttata, e altri in cui il suo utilizzo non superava il 10%. La soluzione al problema fu iniziare a vendere, in qualità di servizi Cloud, l'eccesso di tale infrastruttura nei periodi con minore carico.

- Applicazioni a bassa frequenza d'utilizzo e archivi di dati.

1.3 Analisi di mercato

La Cloud Technology ha un elevato potenziale commerciale. Stando alle ricerche sul mercato effettuate dalla IDC (International Data Corporation), la spesa per i servizi Cloud è cresciuta da 16 miliardi di dollari, nel 2008 [9], a circa 58 miliardi di dollari nel 2013, e si prevede che raggiunga i 191 miliardi di dollari entro il 2020 [10]. In Italia, il mercato di questo settore dell'Information Technology vale già 493 milioni di euro [11]. E' chiaro quindi che il mercato Cloud rappresenti non solo il presente dell'informatica, ma anche il suo futuro.

Un recente studio [12] ha stimato i costi aziendali necessari al mantenimento di un servizio di archiviazione Cloud, grazie alla realizzazione di un centro di dati e ad un'accurata analisi di costi diretti e indiretti, che spaziano dalle spese iniziali per l'hardware, a quelle per l'affitto dell'immobile, per l'energia e per la manutenzione dei componenti. Il risultato dello studio è la stima del costo necessario a immagazzinare un singolo byte di informazioni in un piccolo centro dati Cloud per un intero anno, ed ammonta a $71,51 \times 10^{3}$ US picocents, con 1 US picocent pari a $\$1 \times 10^{-14}$. In tabella 1.1 sono invece riportati i prezzi dei servizi di storage per l'utenza, offerti da tre fra i principali colossi del Cloud Computing – Google, Amazon e Microsoft – e aggiornati al settembre 2014.
Al di là del confronto fra i provider, si può dedurre l'enorme profitto derivante dalla vendita dei servizi Cloud accostando i dati di questa tabella ai risultati ottenuti dallo studio dei costi aziendali sopracitato [12]. Per calco-

Spazio richiesto	Costo per GB (Google)	Costo per GB (Amazon)	Costo per GB (Microsoft)
0-1 TB	$0.026	$0.03	$0.03
1-50 TB	$0.026	$0.0295	$0.0295
50-500 TB	$0.026	$0.029	$0.029
500-1000 TB	$0.026	$0.0285	$0.0285
1000-5000 TB	$0.026	$0.028	$0.028

Tabella 1.1: Prezzi di Cloud Storage [13]: Google [14], Amazon Web Services [15], Microsoft Azure [16]

lare il prezzo al quale viene venduta la possibilità di immagazzinare un byte di informazioni su rete Cloud, è sufficiente dividere il prezzo di vendita di 1 gigabyte per il numero di byte in esso contenuti, ossia un miliardo. Utilizzando il valore medio 0.028 $, si ottiene un prezzo di vendita al byte pari a 2.8×10^{-11} $, circa mille volte il costo di "produzione" (benché i dati siano relativi a scenari aziendali differenti).

Capitolo 2

Sicurezza

In questo capitolo, analizzerò i fondamenti di base della sicurezza informatica, illustrando le sue principali aree critiche, i metodi utilizzati per bypassare le protezioni di sistemi e reti, e le tecniche per garantire difesa da tali attacchi. Nel capitolo successivo, esaminerò invece i rischi e i pericoli inerenti alla nuvola informatica, esponendo le tecnologie e i modelli di difesa di ultima generazione.

2.1 Cybersecurity

La sicurezza informatica, in inglese "Cybersecurity" o "Information Technology Security", consiste nella difesa delle informazioni da accessi non autorizzati, guasti e alterazioni, nell'ambito di dispositivi informatici quali computer, smartphone, reti di dati locali e Internet. La protezione di un sistema informatico è ricercata attraverso strategie, metodi e software atti a garantire alcune regole fondamentali [17]:

- *Autenticazione*: l'accesso fisico e/o logico deve essere limitato ai soli utenti autorizzati.

- *Disponibilità*: ogni utente può usufruire di tutti e soli i servizi previsti per il suo account.

- *Integrità*: i dati devono sempre essere corretti e immuni da problemi fisici e logici.
- *Controllo degli accessi*: un utente non può vedere i dati privati di un altro utente.
- *Difesa*: il sistema deve essere protetto da attacchi malevoli in grado di compromettere la stabilità delle regole sopracitate.

L'attuazione di queste regole, relativamente allo sviluppo di un sistema di sicurezza, è un'operazione complessa e costosa, resa difficile dalla presenza di pericoli di natura fisica e logica. Tali pericoli possono trasformarsi in danni significativi, quali la perdita di interi centri di dati, a causa di un incendio, o il furto di informazioni private da parte di dipendenti o hacker esterni, con conseguenti risvolti negativi sulla stabilità finanziaria dell'azienda [18]. Vediamo quindi in cosa consistono esattamente i principali pericoli della sicurezza informatica.

2.2 Main Threats

Allo scopo di controllare i rischi di un sistema informativo, è necessario conoscere in modo approfondito le sue vulnerabilità e le minacce in grado di sfruttare tali vulnerabilità per compromettere l'integrità delle informazioni protette. La conoscenza dei pericoli permette infatti l'implementazione delle più efficaci misure di sicurezza [18]. Segue un'analisi relativa alle principali aree critiche dei sistemi.

2.2.1 Errors and Omissions

Errori e omissioni costituiscono un rischio non trascurabile per l'integrità dei dati e del sistema. Tali errori possono derivare dalla disattenzione di dipendenti e programmatori, influendo direttamente o indirettamente su problemi di sicurezza e generando vere e proprie vulnerabilità [18]. Citando uno

studio del 1989 relativo ai "bug" di programmazione [19]: *"As expenditures grow, so do concerns about the reliability, cost and accuracy of ever-larger and more complex software systems. These concerns are heightened as computers perform more critical tasks, where mistakes can cause financial turmoil, accidents, or in extreme cases, death"*. Naturalmente, dalla pubblicazione di questo studio, l'industria dei software ha registrato enormi miglioramenti qualitativi, tuttavia il concorrente aumento dell'utilizzo di tecnologie informatiche in ambito pubblico e privato fa sì che il rischio di errori in fase di programmazione sia tutt'ora potenzialmente catastrofico.

2.2.2 Fraud, Theft and Employee Sabotage

I sistemi informativi possono essere sfruttati a scopi fraudolenti. Nell'ambito di istituti bancari, un dipendente potrebbe prelevare esigue quantità di denaro da più depositi privati, contando sul fatto che discrepanze quasi impercettibili non vengano investigate [18]. Ma il problema non è confinato all'ambiente finanziario: qualsiasi sistema contenente informazioni preziose relative a persone, o conoscenze di produzione, è a rischio. L'Unità dei Crimini Informatici del Dipartimento di Giustizia americano ritiene che gli addetti ai lavori costituiscano la più grande minaccia ai sistemi informatici [20], poiché, avendo accesso diretto al sistema (e conoscendo quindi i suoi difetti e le relative vulnerabilità) sono nella posizione migliore per chi intende commettere un crimine. Le ragioni alla base di frodi e sabotaggi interni sono spesso riconducibili al desiderio di vendetta o ad un'insofferenza sul posto di lavoro: *"As long as people feel cheated, bored, harassed, endangered, or betrayed at work, sabotage will be used as a direct method of achieving job satisfaction – the kind that never has to get the bosses' approval"* [21]. Alcuni scenari di sabotaggio di un sistema consistono in:

- Distruzione di risorse hardware
- Cancellazione o alterazione di dati
- Inserimento di dati volutamente errati nel database informativo

2.2.3 Loss of Physical and Infrastructure Support

Interruzione di corrente, picchi o cali di tensione, allagamenti e incendi possono causare la perdita definitiva di infrastrutture e informazioni, con conseguente inattività del sistema e dei servizi. E' importante effettuare accurate analisi dei rischi e implementare metodi di contenimento dei danni (ulteriori approfondimenti nel capitolo 5).

2.2.4 Malicious Hackers

Il termine Malicious Hackers, alias *Crackers*, si riferisce a coloro che accedono a un computer o a un sistema informatico senza averne l'autorizzazione. Rispetto ai sabotatori interni al sistema, sono tipicamente *outsiders*, raggruppabili in sette principali categorie [22]:

- *Cyber Criminals*: hanno come unico obiettivo il furto di denaro, *hic et nunc*, realizzato attraverso l'appropriazione di codici bancari, numeri di carte di credito, password e identità.

- *Spammers / Adware Spreaders*: il loro scopo è diffondere a dismisura avvisi pubblicitari e promuovere prodotti commerciali. Spesso sono finanziati da compagnie che desiderano aumentare le proprie vendite.

- *Advanced Persistent Threat Agents*: gli agenti APT sono gruppi organizzati che mirano all'appropriazione di proprietà intellettuali di altre aziende. A differenza dei Cyber Criminals, non sono interessati al guadagno rapido, bensì investono in progetti illegali a lungo termine, quali lo sfruttamento delle idee rubate per la realizzazione di profitti in altri paesi, o la vendita di tali informazioni al miglior offerente.

- *Corporate Spies*: come gli agenti APT, hanno lo scopo di rubare proprietà intellettuali dei competitori, tuttavia non sono organizzati in gruppi e si focalizzano su un guadagno finanziario più a breve termine, spesso allo scopo di avvantaggiare compagnie rivali o un governo nazionale (e in questo caso si parla di spionaggio governativo).

- *Hacktivists*: molti hacker sono motivati da ragioni sociali, politiche o personali, e mirano a screditare un ente o un'azienda bersaglio. Possono facilmente diventare spie aziendali, se ritengono che possa servire a indebolire ulteriormente la vittima.

- *Cyber Warriors*: hanno come obiettivo l'indebolimento della capacità militare di un avversario. I "guerrieri informatici" possono operare in qualità di agenti APT o spie aziendali, ma unicamente in funzione di uno specifico obiettivo militare.

- *Rogue hackers*: rappresentano i dilettanti del mestiere, il cui modus operandi è irregolare e non finalizzato a uno scopo ben preciso.

2.2.5 Malicious Code

I malicious code sono programmi informatici creati allo scopo di compromettere l'integrità di un sistema e, più generale, di aiutare l'autore a raggiungere i suoi obiettivi fraudolenti. Possono essere suddivisi in più categorie:

- *Virus*: codici maligni che si diffondono nel dispositivo informatico duplicandosi all'interno di più programmi, o in precisi indirizzi di memoria, e sono creati in modo tale da attivarsi all'apertura del file che li contiene [23] [24] o in seguito a una condizione prestabilita. Si trasmettono da un nodo del sistema a un altro tramite lo spostamento dei fili infetti ad opera degli utenti ignari. Segue un esempio di virus espresso in pseudo-linguaggio C:

```
1 /* il virus sceglie a caso un file EXE del sistema e vi
2    si duplica all'interno. Se l'inizio del file contiene
3    la firma del virus, "virus_signature", significa che
4    il file e' gia' stato infettato: in tal caso, il virus
5    seleziona un altro file. */
6    void infect_executable(){
```

```
7         do {
8                 file = get_random_executable_file;
9         } while (virus_signature not in first_line);
10        add_virus_to_file;
11 }
12
13 /* effettua il danno */
14    void do_damage(){
15          ...
16 }
17
18 /* controlla se e' verificata una precisa condizione */
19    d'innesco */
20    void trigger_pulled(){
21          ...
22 }
23
24 /* il main chiama la funzione che replica il virus
25    all'interno degli altri file eseguibili del sistema,
26    ed esegue il suo codice malevolo ogni volta che la
27    condizione prestabilita e' verificata. */
28    main(){
29          infect_executable();
30          if (trigger_pulled()){
31                  do_damage();
32          ...
33          }
34 }
```

- *Worm*: programmi che sfruttano falle nella sicurezza di rete per diffondersi in altri dispositivi [24]. Solitamente, il loro codice maligno ha lo scopo di cancellare file dal dispositivo, appesantire il sistema e sfruttare l'indirizzo email dell'utente per auto-inviarsi in rete.

- *Batch File*: non sono propriamente malware, bensì semplici file di testo contenenti sequenze di operazioni interpretabili dal prompt dei comandi. Tuttavia, comandi legittimi quali cancellazione di directory e

formattazione di partizioni possono essere utilizzati, all'insaputa dell'utente, a scopi fraudolenti, rendendo in tal modo i file batch potenziali programmi malevoli. Il seguente codice cancella l'intera directory C: del sistema, senza chiedere conferma *("Are you sure? Y/N")* e in modo "silenzioso", tramite la disabilitazione dell'interfaccia.

```
1 @echo off
2 cd..
3 cd..
4 DEL *.* /Q /F /S
```

- *Backdoor*: codici in grado di aprire porte di rete per consentire, all'insaputa dell'utente vittima, un accesso al suo sistema in remoto.

- *Spyware*: programmi utilizzati per la raccolta di informazioni, fra le quali password e chiavi crittografiche, dal sistema bersaglio.

- *Keylogger*: programmi simili agli spyware, registrano tutto ciò che un utente digita su tastiera permettendo in tal modo il furto di password o dati sensibili [24].

- *Trojan horse*: software che, oltre a contenere funzionalità lecite e previste, allo scopo di attirare l'attenzione dell'utente, presentano anche istruzioni dannose eseguite a sua insaputa [24]. In sostanza, trattasi di codici maligni camuffati da programmi legittimi [25]. Il termine deriva dalla leggenda dell'Antica Grecia, in cui gli Achei utilizzarono il celebre cavallo di legno per ingannare i Troiani e introdurre in città guerrieri nascosti. Il seguente codice avvia un'istanza del web browser Mozilla Firefox, (ammettendo che questo sia installato nel pc), e, al contempo, esegue il medesimo processo di cancellazione visto nell'esempio precedente.

```
1 @echo off
```

```
2 "C:\Program Files\Mozilla Firefox\firefox.exe"
3 cd..
4 cd..
5 DEL *.* /Q /F /S
```

- *Dialer*: programmi in grado di connettere la linea telefonica della vittima a numeri a tariffazione speciale, allo scopo di trarne illecito profitto a insaputa dell'utente [24].
- *Hijacker*: programmi, in grado di contaminare applicazioni di navigazione in rete, che causano l'apertura automatica di pagine web indesiderate [24].
- *Rootkit*: programmi che nascondono, sia all'utente sia a programmi difensivi quali gli antivirus, la presenza di determinati file o impostazioni del sistema [24].
- *Rabbit*: codici maligni che esauriscono le risorse del computer tramite autoduplicazione, in memoria o su disco, a grande velocità [24]. Un classico esempio di Rabbit che sfrutta i comandi Batch di Windows:

```
1 @echo off
2 :s
3 start "" %0
4 goto s
```

e un ulteriore esempio scritto in C:

```
1 #include <unistd.h>
2 int main(void){
3     while(1){
4         fork();
5     }
6 }
```

- *Adware*: programmi che si insediano nel sistema ed eseguono ordini provenienti da un pc remoto, spesso offuscato grazie all'uso combinato di rootkit [24], principalmente allo scopo di mostrare annunci pubblicitari, reindirizzare le richieste di ricerche web verso determinati siti e raccogliere dati di marketing sull'utente [26].

- *Exploit*: codici che permettono l'acquisizione dei privilegi amministrativi di un computer, sfruttando bug o vulnerabilità del sistema.

- *Shellcode*: programmi che ottengono l'accesso alla shell di un dispositivo ed eseguono comandi arbitrari. Sono spesso la conseguenza di un exploit, con il quale si ottengono i privilegi di amministratore necessari alla successiva esecuzione di comandi critici da shell.

2.2.6 Ulteriori tecniche di Hacking

In aggiunta ai malware e agli altri codici malevoli, un criminale informatico può sfruttare numerose strategie e tecniche in grado di semplificare le sue operazioni illecite. Analizziamole nel dettaglio:

- *Buffer Overflow*: l'hacker sfrutta una condizione di errore che si verifica quando si tenta di scrivere in un buffer di memoria più dati di quanti esso ne possa contenere [27]. In questi casi, viene sovrascritta parte della zona di memoria adiacente al buffer, con conseguente possibilità di corrompere dati di impostazione del sistema e ottenerne il controllo. I linguaggi di programmazione più vulnerabili agli attacchi buffer overflow sono il C e il C++, dato che non prevedono controlli relativi alle dimensioni degli array.

- *Denial of Service*: letteralmente "negazione del servizio". L'hacker può rallentare o bloccare le attività di un server o, più in generale, di una rete, e negare agli utenti l'accesso alle informazioni e ai servizi [28] [29]. Tipicamente, un attacco DoS è realizzato attraverso l'invio di un'enorme quantità di richieste al dispositivo vittima, il quale non è più

in grado di gestire le richieste legittime degli utenti. Gli attacchi DoS rappresentano uno dei principali pericoli per la sicurezza informatica (ulteriori approfondimenti nella sezione 3.1.5, da p. 37 a p. 47).

- *Ingegneria Sociale*: attacchi che sfruttano l'ingenuità degli utenti, anziché le vulnerabilità del sistema, per ottenere informazioni private quali password e codici bancari [30]. Un "ingegnere sociale" è quindi un truffatore informatico, abile nell'arte dell'inganno. Un esempio classico di ingegneria sociale è una lettera di posta elettronica in cui l'hacker finge di rappresentare il servizio clienti di eBay e chiede all'utente di reimpostare la sua password a causa di problemi tecnici ai server. L'utente, ingenuamente, visita il link fornito (pagina opportunamente allestita per assomigliare al sito originale di eBay) e digita i suoi dati di login nell'apposito form, regalando di fatto i suoi dati sensibili all'autore dell'attacco.

- *Spoofing*: tecnica di supporto che permette falsificazione dell'identità (utilizzata dagli hacker per evitare di essere rintracciati), e falsificazione di URL (per intercettare dati sensibili degli utenti o facilitare il lancio di attacchi distribuiti).

2.3 Strumenti di difesa

Vediamo ora alcuni elementi di base atti a proteggere i sistemi dagli attacchi sopracitati e a tutelare le regole fondamentali di cui alle pp. 11-12 (ulteriori aree di sicurezza e implementazioni avanzate dei seguenti metodi saranno introdotte nel capitolo 3, in ambito Cloud):

- *Antimalware*: programmi in grado di rilevare file infetti da malware all'interno del sistema ed eliminarli prima che l'utente possa aprirli e compromettere la sicurezza del suo computer. Si basano su una estesa conoscenza dei codici maligni in circolazione: se il programma riscontra una corrispondenza con uno o più malware presenti nel suo database,

avvia la disinfezione del sistema. Possono essere predisposti per la ricerca di una particolare tipologia di malware (e.g.: Antivirus[1], Antispyware). Alcuni produttori di antimalware: Kaspersky [31], Bitdefender [32], Avira [33].

- *Firewall*: componente di difesa perimetrale di una rete in grado di controllare il traffico in entrata e in uscita, tramite il filtraggio dei pacchetti dati, fornire protezione da intrusioni esterne e impedire a programmi nascosti all'interno del computer di accedere ad Internet ad insaputa dell'utente [34]. Una regola di filtraggio può essere così formalmente definita:

$$R_i : \{C_i\} \rightarrow D_i \tag{2.1}$$

dove i indica la posizione della regola all'interno del set di configurazione, D_i è un'espressione booleana relativa alle azioni di filtraggio del traffico associate, e $\{C_i\}$ è un set di attributi che viene combinato con espressioni condizionali E_n tali che $\{C_i\} = E_1 \cap E_2 \cap \ldots \cap E_n$, con n pari al numero di espressioni condizionali relative alla regola di filtraggio considerata. Dunque, una lista ordinata di regole di filtraggio specifica le azioni da eseguire sul flusso, sulla base delle condizioni verificate [35]. Le tecnologie firewall di ultima generazione implementano schemi DPI *(Deep Packet Inspection)*, una forma di filtraggio del traffico che, oltre a effettuare l'analisi dell'header dei pacchetti, controlla anche il loro payload, allo scopo di rilevare eventuali divergenze dai protocolli [36], le quali potrebbero indicare la natura malevola del traffico analizzato, o la presenza di virus e spam. L'ispezione dei pacchetti è spesso effettuata tramite port mirroring, ovvero l'invio di una copia dei pacchetti selezionati ad una rete ausiliaria di controllo (ulteriori approfondimenti alle pp. 43-47).

[1]Per semplicità, il termine Antivirus viene spesso utilizzato come sinonimo di Antimalware.

- *Crittografia*: processo di trasformazione di un testo semplice in testo codificato, definito *cipher text*, mediante un algoritmo risolvibile unicamente con l'utilizzo di una chiave segreta. Un testo cifrato può essere quindi interpretato solo da coloro che conoscono la chiave, la quale permette di ritrasformare il testo cifrato in testo semplice [37]. Esistono due principali strategie crittografiche: i sistemi a chiave segreta *(secret key / symmetric systems)* e i sistemi a chiave pubblica *(public key / asymmetric systems*). Nel caso di sistemi a chiave segreta, due (o più) interlocutori condividono la stessa chiave, e la utilizzano per la codifica e la decodifica delle informazioni; nei sistemi a chiave pubblica, ogni interlocutore possiede una coppia di chiavi, delle quali una è pubblica, e può anche essere comunicata tramite canali non sicuri, e l'altra è privata, e deve rimanere confidenziale [18]. Il rovescio della medaglia, relativamente all'utilizzo della crittografia, mostra una riduzione delle performance del sistema, causata dai tempi di attesa per la codifica e la decodifica dei dati. Risulta utile, pertanto, effettuare una valutazione dei rischi che determini quali parti del sistema sia opportuno criptare, e quali no (o, in alternativa, per quali parti del sistema sia necessario uno schema a chiave pubblica, più complesso, e per quali sia sufficiente uno schema a chiave segreta, meno sicuro ma più rapido). Maggiore è il danno causato da un'eventuale fuoriuscita dell'informazione, maggiore è il grado di crittografia necessario per la sua protezione: si genera così un inevitabile trade-off tra sicurezza dei dati e prestazioni del sistema [4].

  Attualmente uno dei principali algoritmi di crittografia è l'Advanced Encryption Standard (AES), che sfrutta una chiave segreta binaria di lunghezza variabile (128, 192 o 256 bit) [38]. Un esempio di software di crittografia open-source è TrueCrypt, che permette di nascondere all'interno di un file contenitore un disco criptato virtuale, il quale può essere montato come disco rigido vero e proprio. Il programma supporta i più efficaci algoritmi di crittografia: l'AES, il Serpent (che, utilizzan-

do una rete a sostituzione e permutazione a 32 passaggi, anziché quella classica a 16, ottiene un grado di sicurezza persino maggiore di quello offerto dall'AES, a scapito delle performance [39]) e il Twofish (che si basa su un'altra struttura di cifratura, la Rete di Feistel, e ottiene risultati simili a quelli dell'AES [39]). È possibile utilizzare i tre algoritmi in cascata, aumentando notevolmente le complessità di codifica dell'informazione.

- *Steganografia*: tecnica che permette di nascondere informazioni sensibili all'interno di altri dati. A differenza della crittografia, che protegge un dato osservabile, la steganografia ha lo scopo di occultare la presenza del dato stesso. Fra i vari tipi di dati utilizzati come contenitori dell'informazione nascosta (immagini, audio, video, testi), il formato immagine digitale JPEG è il più comune [40], e la tecnica steganografica più diffusa è la Steganografia LSB [41] (anche conosciuta come "rumore di fondo"). Tale tecnica sfrutta il fatto che l'aspetto di un'immagine digitale non cambia se i colori vengono modificati in modo impercettibile. Si può pertanto correggere il bit meno significativo di ogni pixel dell'immagine, senza che questa risulti diversa. L'algoritmo LSB sovrascrive proprio i bit meno significativi dall'immagine "contenitore" con i dati da nascondere, seguendo un ordinamento dettato da una chiave segreta. Pertanto, l'hacker interessato a ottenere il dato occultato, non solo non ha motivo di pensare che l'immagine contenga il dato ricercato, ma deve anche conoscere la chiave steganografica. Un esempio di software gratuito per la steganografia è OpenPuff [42].

- *Intrusion Detection System*: sistemi per il rilevamento delle intrusioni, in grado di identificare accessi non autorizzati al computer e alla rete, tramite analisi comportamentali degli utenti e anomalie a livello di sistema. Uno dei principali fattori discriminanti tra firewall e IDS è la Deep Packet Inspection (p. 21), che prevede l'ispezione del *payload* dei pacchetti, ossia i messaggi veri e propri. La DPI è opzionale nel caso

di firewall, mentre è prerogativa di ogni IDS (ulteriori approfondimenti nel capitolo 4).

- *Autenticazione multifattore*: sistemi di sicurezza che richiedono più di una forma di autenticazione per verificare la legittimità dell'accesso. Rispetto al classico modello di autenticazione, basato su singole password, gli schemi MFA aumentano il grado di identificazione dell'utente. Nel caso di autenticazione a 3 livelli, le verifiche di sicurezza sono rappresentate da "qualcosa che solo l'utente sa" (e.g.: password, PIN), "qualcosa che solo l'utente possiede" (e.g.: codici segreti inviati tramite SMS o ricavabili dall'estratto conto bancario, tessere elettroniche) e "qualcosa che solo l'utente è" [43] (e.g.: impronte digitali, scansione della retina, timbro vocale). In data 15 settembre 2014 è stato registrato il completamento del sistema di identificazione biometrica dell'FBI, chiamato *Next Generation Identification System* [44] [45] (NGI), sul quale il bureau investigativo ha lavorato sin dal 2011. Il sistema è in grado di gestire numerosi livelli di identificazione biometrica (voce, retina, impronte digitali, segni particolari) e implementa un meccanismo di riconoscimento facciale che coordina videocamere e infrastrutture di supporto a livello nazionale.

Capitolo 3

Sicurezza dei sistemi Cloud

La sicurezza dei sistemi Cloud rappresenta la nuova frontiera della sicurezza informatica, in cui strategie, protocolli e modelli devono essere implementati anche a livello virtualizzato, ed essere costantemente aggiornati, per far fronte ai pericoli più recenti. Prima di entrare nel dettaglio, tuttavia, è opportuno specificare l'importanza del concetto di affidabilità di un provider. Nel Cloud Computing, l'architettura e la gestione del servizio sono invisibili agli utenti (e da qui la parola "Cloud", nuvola, per descrivere un'entità assai distante dalla nostra portata fisica) [9]; pertanto, abbandonando i classici sistemi fisici di controllo e di accesso ai dati, gli utenti si interrogano circa la sicurezza e l'integrità del sistema. Per rendere più chiaro il concetto di affidamento a un provider Cloud, si possono prendere in considerazione, per analogia, le istituzioni finanziarie, quali le banche, presso cui il cliente deposita una somma di denaro in un conto personale, e cessa di avere il possesso fisico della somma, pur essendone ancora il proprietario. Ne deriva la convinzione del cliente di potersi affidare all'integrità finanziaria e tecnologica della banca per la protezione del suo patrimonio [9]. Allo stesso modo, in ambito informatico, il cliente si rivolge ad aziende fidate per la gestione dei suoi dati sensibili, ed è di fondamentale importanza, nel caso in cui anche il cliente sia un'azienda professionale, chiedere al provider, prima di sottoscrivere con questi un contratto, di poter conoscere i metodi di sicurezza applicati, allo

scopo di delineare il suo profilo di integrità e solidità. Le informazioni utili sono principalmente l'architettura del sistema (per capire quante e quali strategie vengono attuate a protezione dei dati sensibili), i modelli di crittografia (per sapere se vengono rispettati gli standard), la gestione delle patch e degli scenari di disservizio (per comprendere quali effetti avranno sul servizio eventuali aggiornamenti del sistema, e sapere come il provider pensa di gestire situazioni in cui il servizio non sia disponibile), la localizzazione geografica delle infrastrutture (per conoscere le leggi vigenti nel territorio in cui si trovano le informazioni), la portabilità del sistema (per assicurarsi la possibilità, in futuro, di trasportare applicazioni su altri sistemi senza eccessive perdite di tempo e lavoro) e la segregazione (per sapere se l'infrastruttura o alcune applicazioni del sistema sono condivise con altri utenti, o completamente riservate).

3.1 Cloud Top Threats

Scendiamo ora nel dettaglio, analizzando le più rilevanti insidie del Cloud Computing. La Cloud Security Alliance[1], nel documento "Top Threats to Cloud Computing" [47], elenca le principali categorie di pericoli della nuvola informatica, allo scopo di assistere aziende e organizzazioni nella gestione dei rischi che riguardano l'ambiente Cloud. Per ogni categoria di rischio, vengono suggerite le principali strategie di prevenzione. Il documento viene costantemente aggiornato, per riflettere i pareri degli esperti di sicurezza riguardo ai rischi emergenti. Il più recente aggiornamento risale a febbraio 2013, e presenta nove categorie di pericoli, elencate in ordine decrescente di criticità. Nelle seguenti sezioni, illustrerò caratteristiche, tecniche offensive e principali contromisure per ognuna delle aree critiche, preservando il loro ordine di classificazione.

[1] Organizzazione no-profit creata allo scopo di promuovere l'utilizzo delle migliori tecniche per la sicurezza del Cloud Computing [46].

3.1.1 Data Breaches

La principale minaccia in ambito Cloud consiste in violazioni della sicurezza in cui informazioni confidenziali o dati sensibili vengono visualizzate, trasmesse o copiate da un individuo non autorizzato a compiere queste operazioni [48]. Tali violazioni riguardano principalmente dati finanziari, come codici bancari e di carte di credito, informazioni per l'identificazione o la localizzazione di una persona (PII, Personally Identifiable Information) e proprietà intellettuali. Se il sistema Cloud non è efficacemente implementato, eventuali falle nell'applicazione di un solo cliente possono permettere all'hacker di ottenere, oltre ai dati di quel cliente, anche quelli di altri clienti del sistema. L'*autotrunking*, ad esempio, è un protocollo proprietario dei sistemi Cisco[2] che consente alle porte di rete di gestire il traffico dati di più VLAN, semplificandone la gestione. Tuttavia, tale configurazione può rendere il sistema più vulnerabile, dato che la compromissione di una singola porta espone i dati sensibili di tutte le reti virtuali connesse [49]. Sempre nell'ambito della virtualizzazione, i ricercatori della RSA[3] e delle Università del North Carolina e del Wisconsin, hanno spiegato come poter sfruttare un side channel timing attack[4] per estrarre chiavi crittografiche private contenute in macchine virtuali [51].
Contromisure:

- *Secure Disposal* [52]: evitare backup di informazioni e ridondanze, e stabilire politiche e procedure per la rimozione dei dati da tutti i dispositivi di memorizzazione al di fuori della rete Cloud.

- *Non-Production Data* [52]: i dati di produzione, ovvero le informazioni che registrano le conoscenze aziendali riguardo ai processi

[2]Cisco: è una delle aziende leader nel campo della sicurezza e delle reti informatiche.

[3]Divisione della Sicurezza della EMC, una delle maggiori aziende nel campo dell'Information Technology.

[4]Attacco informatico in cui l'autore tenta di compromettere un sistema di crittografia analizzando il tempo impiegato dal sistema per l'esecuzione dei suoi algoritmi [50].

di produzione, non devono essere replicati o utilizzati in ambienti di non-produzione.

- *Information Leakage* [52]: implementare meccanismi di sicurezza per prevenire la fuoriuscita di dati. Fra le principali possibilità:

  - *Chaffing and Winnowing*: tecnica che consiste nell'aggiunta di pacchetti di dati falsi al messaggio vero e proprio, in modo che risulti impossibile distinguere le informazioni fittizie da quelle reali, a meno che non si conosca la chiave di lettura [53]. Spesso questa tecnica è utilizzata nell'ambito di reti di comunicazione che inviano i pacchetti dati in ordine casuale, allo scopo di aumentare ulteriormente la protezione dell'informazione. I pacchetti di dati del mittente contengono pertanto l'indice del pacchetto, l'informazione e un codice di autenticazione del messaggio (MAC), l'elemento discriminante per la determinazione della legittimità del pacchetto. Il destinatario, che possiede la chiave di lettura, conosce il criterio in base al quale separare i MAC autentici da quelli fittizi, e può estrarre il messaggio originale.
    Il Chaffing and Winnowing, non essendo categorizzabile come metodo crittografico, è particolarmente utile nei paesi in cui è vietata per legge l'esportazione di informazioni crittografiche, o la crittografia stessa delle comunicazioni [53].
  - *Steganografia*: p. 23.

- *Risk Assessments* [52]: effettuare, a intervalli regolari, valutazioni dei rischi riguardanti la gestione dei dati.

- *Encryption*: applicare tecniche di crittografia per la protezione dei dati sensibili (p. 19).

- *Encryption Key Management* [52]: stabilire politiche e procedure per garantire un'efficace gestione delle chiavi crittografiche. In uno scenario ideale, la chiave di crittografia dovrebbe essere sostituita ad ogni messaggio inviato, in modo che solo tale messaggio risulti comprensibile nel caso in cui la chiave venga rubata. Sfortunatamente, uno scenario simile comporta un notevole appesantimento della comunicazione, pertanto è necessario effettuare un'analisi dei rischi e valutare il trade-off tra sicurezza e performance, allo scopo di ottenere un buon grado di sicurezza senza compromettere l'efficacia del sistema.

- *User ID Credentials* [52]: rafforzare le credenziali d'accesso e i controlli sulle password per applicazioni, database e infrastrutture; verificare l'identità di un utente prima di permettere il reset della sua password, la quale deve essere cambiata almeno ogni 90 giorni e deve avere una lunghezza minima di 7 caratteri alfanumerici. Se l'utente, dopo aver effettuato l'accesso, diventa inattivo per più di 15 minuti, dovrà inserire nuovamente la password (vincolo comune nei siti di online banking).

- *Production/Non-Production Environments* [52]: separare gli ambienti di produzione da quelli di non-produzione, allo scopo di prevenire o ridurre accessi non autorizzati – o modifiche – al set informativo di conoscenze aziendali.

- *Remote User Multi-Factor Authentication* [52]: richiedere l'autenticazione multi-fattore per ogni accesso remoto (p. 24).

- *VLAN Hopping prevention*: Nell'ambito delle reti virtuali, e con particolare riferimento ai sistemi Cisco, spesso l'hacker simula il comportamento del protocollo di trunking per dirottare i traffici gestiti dalla trunk port a un host controllato. La soluzione più efficace consiste nel disabilitare la funzione di trunking, tuttavia è

possibile mantenere la trunk port e al contempo bloccare l'attacco configurando gli switch in modo tale che non possano "negoziare" scambi di traffico [49], tramite il comando:

```
switchport nonegotiate
```

3.1.2 Data Loss

Il secondo principale rischio del Cloud Computing consiste in guasti che causano la distruzione di informazioni immagazzinate, trasmesse o in fase di elaborazione. Possono verificarsi a livello software (e.g.: crash / bug del programma, errori accidentali) o a livello hardware (e.g.: interruzione di corrente, compromissione dei server a causa di bancarotta del provider o di disastro naturale). Sfortunatamente, i principali metodi per prevenire la perdita di informazioni, quali la creazione di backup di una o più parti del database, esacerbano i rischi di data breaches causati da violazioni esterne [47].
Contromisure:

- *Retention Policy & Risk Assessments* [52]: generare periodicamente backup dei dati più sensibili del sistema, valutando, tramite analisi dei rischi, per quali parti del sistema sia necessario effettuare backup più frequenti, e per quali invece siano sufficienti salvataggi più sporadici, nel caso in cui la relativa perdita di informazioni produca un danno minore.

- *Environmental Risks* [52]: garantire protezione fisica all'infrastruttura contro danni naturali (e.g.: incendi, allagamenti, terremoti) tramite l'utilizzo di sensori in grado di rilevare dati ambientali quali temperatura, umidità e flusso d'aria, mezzi di isolamento termico come porte tagliafuoco, e monitor per controllare lo stato delle proprie infrastrutture in remoto.

- *Equipment Location* [52]: per limitare i danni in caso di disastri ambientali, i server più importanti devono essere collocati in luoghi distanti da aree ad alto rischio ambientale, e sostenuti da infrastrutture ridondanti collocate a ragionevole distanza.

- *Resiliency Management Program, Impact Analysis & Business Continuity Planning* [52]: stabilire procedure di ripresa per minimizzare l'impatto di un evento negativo sull'azienda e facilitare la salvaguardia del patrimonio informativo, attraverso un programma che combini controlli preventivi e di recupero. Identificare i settori più critici del sistema, le sue dipendenze, sia interne sia da terze parti, e determinare le priorità per il recupero. Stabilire quindi un piano di continuità aziendale per documentare le strategie che permettano all'azienda di continuare ad esercitare il suo business, anche a fronte di avversità.

3.1.3 Account or Service Traffic Hijacking

Attacchi in cui l'hacker ruba – tramite phishing, truffa o sfruttamento delle vulnerabilità dell'applicazione – i dati di accesso dell'account di un utente, e li utilizza per effettuare operazioni malevole o non autorizzate [54]. Molti utenti scelgono la stessa password per più account, con conseguente possibilità, da parte dell'hacker, di ottenere l'accesso a più servizi: si amplifica così l'impatto negativo di questo genere di attacchi [47]. L'hacker può intercettare le attività della vittima, manipolare o corrompere i suoi dati e ottenere i codici bancari per sottrarre denaro. Nell'ambito dei siti web dinamici, una delle principali vulnerabilità della sicurezza è il Cross-site scripting (XSS), che permette agli attaccanti di inserire codice maligno in siti web affidabili, sfruttando i form[5] offerti agli utenti. Stando a un rapporto di Cenzic[6] del 2014, circa il 25% delle

[5]Form: interfaccia web che consente a un client di digitare dati e inviarli al server.

[6]Azienda che opera nel campo della sicurezza informatica, con particolare focus su settori Cloud, Mobile e Web.

violazioni web sono dovute a XSS, che detiene il primato di "top web vulnerability" [55].
Contromisure:

- *Identities Management* [4]: vista la possibilità, per un utente, di richiedere molteplici servizi allo stesso provider, sarebbe inefficiente gestire i dati di accesso di ogni servizio separatamente, e lasciare che il cliente utilizzi un login diverso per ogni servizio invocato. Il pericolo maggiore derivante da questo scenario è che l'utente, per comodità, inizierebbe a utilizzare il salvataggio automatico delle password sul suo web browser, evitando così di dover ridigitare ogni volta i dati di accesso, e favorendo gli hacker nei tentativi di phishing. La soluzione a questo potenziale problema è il *Single Sign-On* (SSO), uno schema di accesso in cui le identità degli utenti sono gestite in modo incrociato attraverso più sistemi e molteplici servizi: l'utente effettua l'accesso una volta sola, ed è automaticamente autorizzato ad usufruire di tutti i servizi per i quali ha ottenuto il permesso. Un esempio di SSO è *Shibboleth*, un modello open-source di autenticazione unica tra più organizzazioni o più servizi di una stessa organizzazione [56].

- *User Access Policy, Restriction, Authorization & Revocation* [52]: implementare politiche per la gestione, la concessione e la revoca dei permessi di accesso alle applicazioni, al database e all'infrastruttura di rete. Le modalità di accesso, sia standard sia privilegiate, devono essere inizialmente testate e approvate dall'amministrazione prima di essere assegnate agli utenti.

- *User Access Reviews* [52]: tutti i livelli di accesso devono essere revisionati dall'amministrazione a intervalli regolari, allo scopo di monitorare i privilegi concessi agli utenti e rilevare eventuali anomalie nel sistema.

- *Incident Management* [52]: stabilire procedure da attuare in caso di incidenti di sicurezza (e.g.: intrusioni, abusi di privilegi). E' opportuno documentare l'incidente e fornire all'autorità giudiziaria un resoconto completo, sostenuto dai registri degli eventi.

- *User ID Credentials & Remote User Multi-Factor Authentication*: valgono le stesse politiche esposte a p. 29.

- *Audit Logging / Intrusion Detection* [52]: compilare e conservare registri di revisione (audit logs) che registrino le attività degli utenti con accessi privilegiati, i tentativi di accesso non autorizzati e le eccezioni del sistema. Implementare strumenti di rilevamento delle intrusioni (ulteriori approfondimenti nel capitolo 4).

- *Web Protection Libraries* [57]: utilizzare librerie apposite per la codifica e decodifica dei testi inseriti in input form. Un esempio di libreria gratuita è *AntiXSS* di Microsoft, che fornisce un insieme di funzioni HTML, XML, CSS e JavaScript utili per la prevenzione di attacchi cross-site scripting.

3.1.4 Insecure Interfaces and APIs

Le *Cloud APIs* sono interfacce il cui scopo è fornire ai clienti una comoda gestione dei servizi e la possibilità di realizzare applicazioni Cloud. Essendo le interfacce strumenti di interoperabilità fra più utenti e programmi, rappresentano punti critici del sistema; pertanto, se compromesse, rischiano di mettere a repentaglio la sicurezza di tutte le applicazioni ad esse collegate, e, di conseguenza, di tutti gli utenti che le utilizzano. Per garantire a un cliente la sicurezza del suo canale di comunicazione, si utilizzano i certificati, documenti elettronici il cui scopo è garantire l'autenticità di un servizio. Esistono tre tipi di certificati [58]:

- *CA/Root Certificate*: identificano un ente, pubblico o privato, abilitato a rilasciare un certificato digitale, e garantiscono l'autenticità di un Site Certificate o di un Intermediate CA Certificate. Sono gli unici certificati incorporati nei browser di navigazione, e rappresentano il massimo grado di certificazione.

- *Intermediate/Branch CA Certificate*: come i primi, identificano un ente, pubblico o privato, abilitato a rilasciare un certificato digitale, e garantiscono l'autenticità di un Site Certificate o di un ulteriore Intermediate CA Certificate. Sono utilizzati per la realizzazione di catene di certificazione.

- *Site/Leaf Certificate*: identificano un particolare URL e garantiscono l'autenticità di una chiave di sicurezza.

Sfortunatamente, gli hacker sono in grado di falsificare i certificati, e, sfruttando tecniche di Ingegneria Sociale (p. 20) ingannano gli utenti, convincendoli che i dati trasmessi siano protetti, mentre in realtà vengono inoltrati a un host segreto. La domanda è: "come fa l'hacker a intercettare le informazioni segrete di un client e, al contempo, simulare il comportamento del server, in modo che l'utente riceva la risposta attesa e non si accorga che la comunicazione è compromessa?". La risposta risiede nell'uso di un attacco *"Man-in-the-Middle"*, definito giustappunto come tecnica offensiva in cui l'attaccante reindirizza la comunicazione fra due utenti al proprio host, senza che questi se ne accorgano [59], e può contaminarla a suo piacimento. Contromisure:

- *Application Security* [52]: le applicazioni devono essere progettate seguendo gli standard di sicurezza industrialmente accettati (e.g.: verifiche OWASP[7] per le applicazioni web).

[7]OWASP (Open Web Application Security Project) è un progetto open-community che presenta quattro livelli di verifica per categorizzare le applicazioni Web in base alla loro capacità di soddisfare determinati requisiti di sicurezza [60]. Tali criteri di verifica rap-

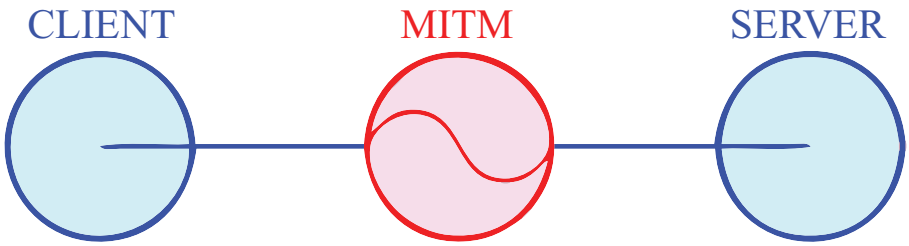

Figura 3.1: Rappresentazione semplificata di un attacco MITM

- *Transport security*: utilizzare un protocollo EAP (*Extensible Authentication Protocol*), quali MD5 (*Message Digest)* o TLS (*Transport Layer Security*) per garantire un canale di comunicazione sicuro fra client e server, nell'accesso alle API del sistema.

- *Man-in-the-Middle attacks prevention*: consideriamo il classico scenario in cui l'hacker intercetta le comunicazioni fra un client e il server. Entrambi i nodi pensano di comunicare direttamente e in sicurezza, grazie al protocollo SSTP (*Secure Socket Tunneling Protocol*), ma in realtà l'hacker sta svolgendo il ruolo di intermediario fra i due, captando le richieste dell'uno e le risposte dell'altro. La figura 3.2 [61] mostra il diagramma delle interazioni fra client, hacker e server nel caso di un attacco MITM in cui il nodo illegittimo è rappresentato da un *access point* fasullo. Per contrastare questo genere di attacchi, è opportuno che il server richieda al client di inserire un attributo speciale (*crypto binding attribute*) nel *Call Connected message*, ossia il pacchetto di dati che rappresenta il passaggio conclusivo della negoziazione SSTP. Tale attributo è ottenuto mediante l'utilizzo di una chiave contenuta all'interno del client stesso. In altre parole, la chiave EAP è ricavata dalla combinazione di due chiavi: una chiave pubblica, che

presentano, per le software house, linee guida sulle quali basarsi per consolidare l'integrità delle applicazioni sviluppate.

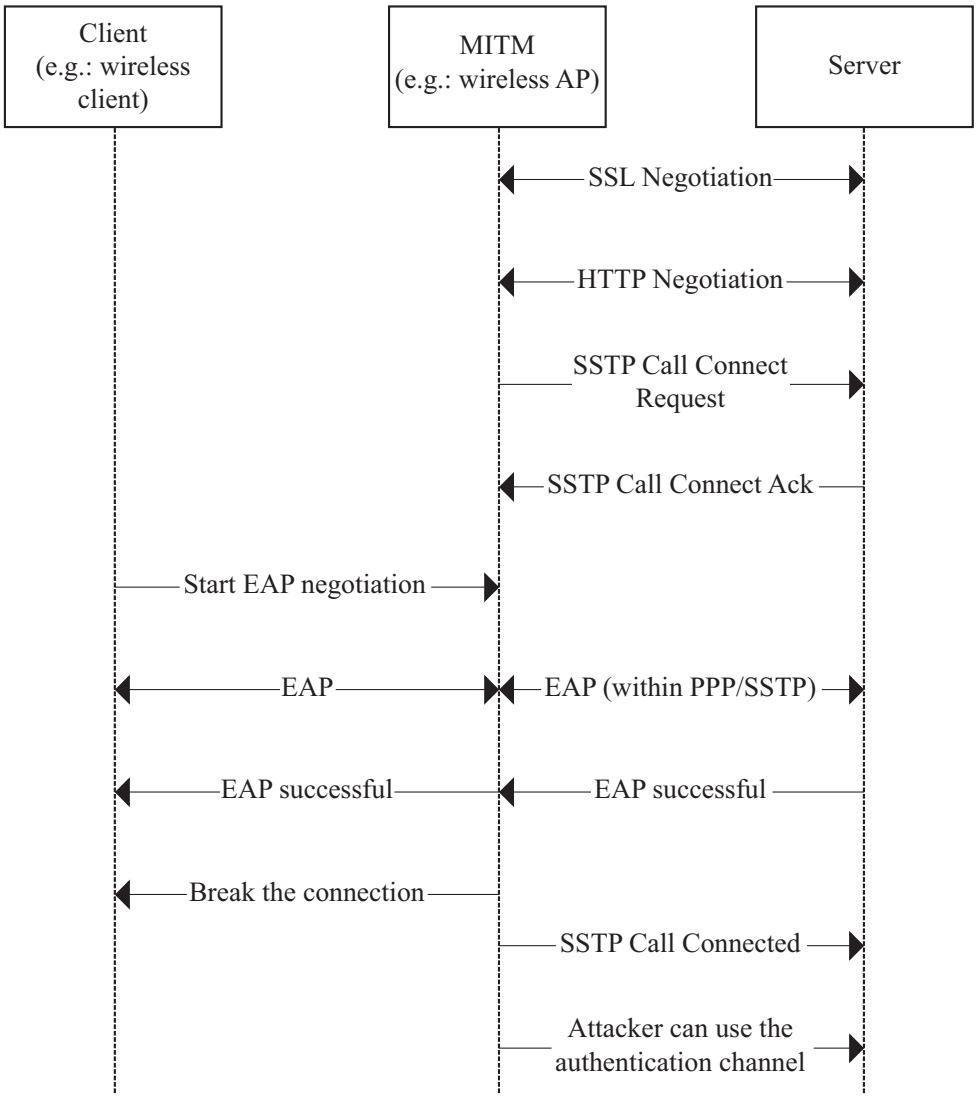

Figura 3.2: Esempio di scenario MITM

viene inviata in rete al destinatario, e una chiave privata, conservata segretamente all'interno del dispositivo. Sfruttando la chiave privata, il mittente genera una firma digitale e la invia al destinatario, il quale, conoscendo la chiave pubblica, può verificarne la legittimità. In questo scenario, l'hacker non può completare la negoziazione SSTP con il server, poiché, non conoscendo la chiave privata del client, non è in grado di replicare la firma digitale necessaria al riconoscimento. Nel caso in cui l'hacker riesca a intercettare la firma digitale inviata al server, non otterrà comunque alcun beneficio, poiché le firme digitali non possono essere riutilizzate, bensì sono rigenerate ad ogni nuova negoziazione, mediante l'utilizzo di una nuova chiave pubblica [62]. Il diagramma delle interazioni in figura 3.3 [61] evidenzia lo scenario risolutivo.

- *Certificate Check*: prima di accettare un certificato, controllare che esso identifichi lo stesso indirizzo al quale si sta tentando di accedere, e che non sia scaduto. Verificare inoltre che il contenuto del campo *basic constraints*, utilizzato per specificare lo scopo del certificato, sia coerente con il servizio garantito [58].

- *User Access Policy, Restriction, Authorization & Revocation*: valgono le stesse politiche esposte a p. 32.

3.1.5 Denial of Service

Un attacco di tipo Denial of Service (DoS) rappresenta il tentativo di precludere a un utente l'utilizzo delle risorse del proprio elaboratore [63]. Se l'attacco è inviato da due o più persone o bot, esso prende il nome di Distributed Denial of Service (DDoS), attualmente uno dei più utilizzati attacchi informatici alle reti Cloud. Tipicamente mediante l'uso di Trojan Horses, il mittente dell'attacco ottiene il controllo di

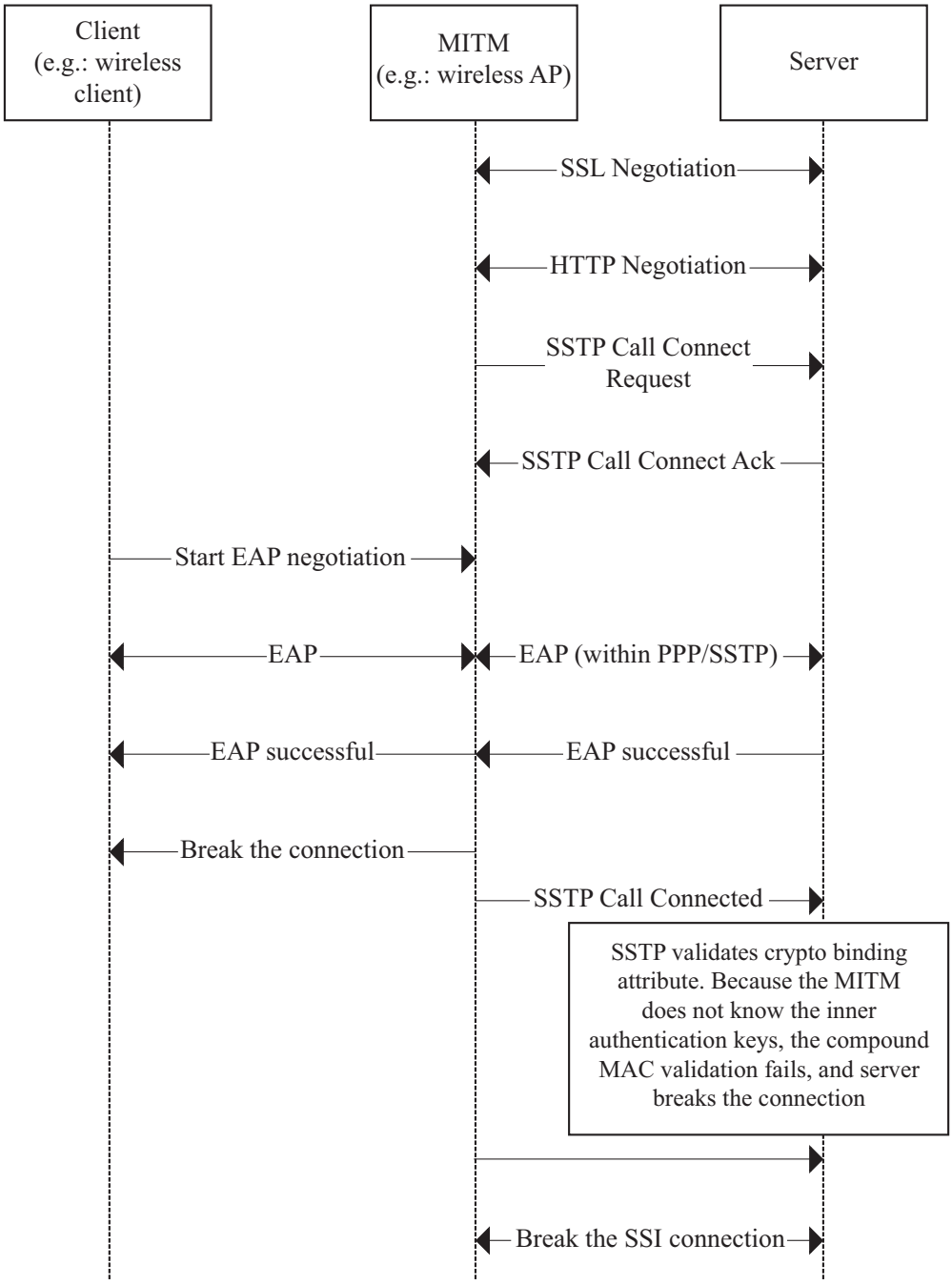

Figura 3.3: Esempio di scenario MITM con soluzione crypto binding

svariati computer, definiti vittime o zombie[8], i quali, inviando ai server della rete il maggior numero possibile di richieste, li sovraccaricano a tal punto da impedire al sistema di gestire le richieste legittime degli utenti, compromettendo il servizio.
Il DoS è il principale metodo offensivo del Cyberterrorismo, in cui le vittime sono enti e organizzazioni che hanno attirato propositi di vendetta di singoli o gruppi organizzati. In altri casi, le motivazioni che spingono un hacker a lanciare un attacco DoS sono di ordine pratico: nella maggior parte dei sistemi Windows NT[9] è necessario riavviare il sistema per rendere effettivi alcuni cambiamenti di impostazioni. L'attaccante, che, con altri metodi, è riuscito ad acquisire i privilegi di amministratore, ma ha bisogno di renderli effettivi, utilizza un attacco DoS per tentare di mandare in crash il sistema e costringere l'amministratore a forzarne il riavvio [65].
Esistono diversi tipi di DoS, ma solo alcuni di essi rappresentano tutt'oggi un pericolo concreto:

- *SYN Flood*: si basa sul protocollo di comunicazione TCP, caratterizzato da una connessione bilaterale stabilita nel seguente modo: il client invia un pacchetto SYN al server; questi risponde con un pacchetto SYN-ACK che rappresenta il riconoscimento (acknowledgement) della richiesta del client e l'apertura della comunicazione da server a client (half-open connection); infine, il client invia al server un ACK che rappresenta il riconoscimento dell'accettazione della sua richiesta, stabilendo la connessione in senso inverso, da client a server, e realizzando così il canale di comunicazione full-duplex. L'attacco DoS di tipo SYN flood sfrutta un comportamento anomalo nell'attuazione della comu-

[8]Zombie: in ambito informatico, indica un computer, connesso a Internet, compromesso da un hacker attraverso virus o trojan horse e utilizzato per l'attuazione di attacchi informatici. La maggior parte dei proprietari di computer zombie non sono al corrente del fatto che i propri dispositivi siano compromessi e sfruttati a scopi fraudolenti da terzi [64].

[9]Famiglia di release di Microsoft Windows, che consiste in: XP, Vista, Seven, 8.

nicazione TCP per sovraccaricare la rete. Un host effettua un enorme numero di richieste al server tramite l'invio di pacchetti SYN. Per ognuna di queste richieste, il server restituisce un pacchetto SYN-ACK, apre una connessione monolaterale e attende il pacchetto finale ACK del client per completare la creazione della comunicazione bilaterale. Tuttavia, l'attacco è strutturato in modo tale che nessuna risposta finale sia inviata al server: tutte le connessioni inizializzate dal server causano quindi saturazione del traffico, riducendo o azzerando il numero di connessioni che il server stesso è in grado di effettuare, e impedendo la gestione delle richieste legittime degli utenti del sistema [66].

- *Ping Flood*: l'hacker travolge il dispositivo vittima con un elevato numero di pacchetti ICMP[10] di tipo ECHO_REQUEST, comunemente utilizzati per il ping[11]. Se la rete non è opportunamente configurata per difendersi da questo tipo di attacco, il dispositivo vittima risponderà con un altrettanto elevato numero di pacchetti ECHO_RESPONSE, consumando la sua banda in uscita. Alternativamente, l'hacker può impostare, tramite spoofing, l'IP del computer bersaglio come indirizzo di origine dei pacchetti, in modo che siano gli altri dispositivi della rete a inviare i pacchetti ECHO_RESPONSE alla vittima, con conseguente consumo della sua banda in entrata.

- *Peer-to-peer attack*: l'hacker forza la disconnessione di un elevato numero di clients dalla rete peer-to-peer alla quale sono connessi,

[10]Internet Control Message Protocol: protocollo utilizzato da dispositivi di rete, quali i router, per inviare notifiche e segnalazioni di errori.

[11]Ping: strumento utilizzato per la misurazione del tempo impiegato da un pacchetto ICMP a raggiungere un dispositivo in rete e a tornare al dispositivo di origine. Invia un pacchetto ECHO_REQUEST al dispositivo bersaglio, allo scopo di ottenere da questi un pacchetto ECHO_RESPONSE. Tale processo è utile per verificare la presenza e la raggiungibilità di dispositivi connessi in rete e per misurare le latenze di trasmissione [67].

e la loro conseguente connessione al dispositivo vittima, causandone il degrado delle performance o il sovraccarico immediato. Per contrastare questo genere di attacco, è necessario l'utilizzo di algoritmi in grado di rilevare le botnet[12] e bloccare il traffico da esse generato. BotHunter [68] può individuare botnet centralizzate o P2P mediante l'analisi di comportamenti anomali quali il port scanning[13] non autorizzato e la presenza di dispositivi in ascolto presso i canali di comunicazione della rete. BotTrack [70] rileva botnet P2P utilizzando l'algoritmo PageRank[14] per creare un grafo delle dipendenze degli host, quindi identifica le botnet osservando i dispositivi con il maggior numero di collegamenti reciproci e caratterizzati da comportamenti simili. Approcci di ultima generazione, quali il modello proposto da Jiang, H. e Shao, X. [71] pongono il focus sulle "Command & Control communications", ossia le comunicazioni fra i molteplici nodi di una botnet e il server origine del botmaster, che rappresenta il "single point of failure"[15] del sistema: individuare ed interrompere tali comunicazioni rende l'intera botnet completamente inattiva. Benché le botnet più elaborate sfruttino politiche di ridondanza quali il backup dei server C&C e numerose strategie controffensive (e.g.: reindirizzare i bot a un diverso server di controllo), l'analisi delle comunicazioni C&C resta comunque uno dei metodi più efficaci contro gli attacchi P2P, offrendo un elevato rilevamento del

[12]Botnet: rete costituita da dispositivi controllati da una singola entità definita botmaster. Questi individua eventuali falle nella sicurezza del sistema, quindi, attraverso malware, infetta componenti dell'infrastruttura allo scopo di utilizzarli in remoto per operazioni illecite.

[13]Port Scanning: processo di scansione delle porte TCP e UDP di un host, per verificare quali sono accessibili e quali invece sono bloccate [69].

[14]PageRank: algoritmo di analisi che assegna un valore numerico ad ogni elemento del sistema, allo scopo di quantificare la sua rilevanza all'interno dell'insieme globale.

[15]Single point of failure: punto critico del sistema che, se compromesso, può causare la disfunzione dell'intero sistema.

traffico malevolo e un ridotto tasso di falsi positivi.

- *Permanent DoS*: ha lo scopo di sabotare le risorse hardware del dispositivo bersaglio, tramite la compromissione delle sue interfacce e la sostituzione dei firmware originali con versioni corrotte [72]. Esempi di DoS permanenti consistono in:
  - modificare le *ACPI* [16] dei ventilatori interni e della batteria, allo scopo di causare il surriscaldamento dell'host o il guasto dei dispositivi stessi;
  - impostare il tasso di refresh dello schermo a un valore estremamente elevato, per sovraccaricare i monitor a tubo catodico;
  - reiterare ininterrottamente la sovrascrittura di ogni dispositivo di memorizzazione, riducendone la vita operativa.
- *HTTP POST DoS*: recentemente, gli attacchi DoS sfruttano vulnerabilità del livello 7 del modello OSI, l'Application Layer. Nell'ambito di un attacco HTTP DoS, l'hacker invia un pacchetto HTTP POST al server, affinché questo accetti di ricevere un messaggio, dopodiché, non appena la connessione è stata stabilita, imposta una velocità di upload estremamente bassa (e.g.: 1 byte ogni 60 secondi). Poiché la richiesta iniziale è legittima, il server non può sapere che il mittente ha appositamente minimizzato la sua velocità di invio, pertanto attende l'arrivo dell'intero messaggio [73]. Un invio multiplo di HTTP POST può quindi compromettere la capacità del server di gestire le richieste di utenti legittimi.

Contromisure: la principale difficoltà nel realizzare un efficace sistema di difesa contro gli attacchi DoS risiede nella necessità di identificare il

[16] Advanced Configuration Power Interface: interfacce per il controllo di un dispositivo e per la gestione della sua alimentazione.

traffico illegittimo e separarlo da quello legittimo. I Firewall, potendo bloccare l'accesso alla rete a determinati indirizzi o porte, costituiscono il perimetro più esterno del sistema di difesa. È possibile, ad esempio, bloccare qualsiasi richiesta proveniente da un indirizzo IP ostile, impedendo così all'autore dell'attacco di eseguire operazioni illecite. Tuttavia, nella maggior parte dei casi, gli attacchi DoS sono sufficientemente complessi da non permettere un'immediata individuazione degli indirizzi IP che hanno causato il flusso di traffico malevolo. Inoltre, il blocco delle porte può risultare controproducente, poiché gli attacchi DoS più sofisticati sfruttano le medesime porte utilizzate dai clienti del sistema, pertanto bloccare una porta specifica potrebbe impedire ai server di gestire il traffico legittimo degli utenti. Per questi motivi, il Firewall deve essere opportunamente implementato affinché possa rappresentare un efficace componente dei sistemi di sicurezza distribuiti, specialmente in ambito Cloud. Una delle strategie principali è l'intensificazione del rate limiting, ossia il controllo sul flusso del traffico inviato e ricevuto:

- Gli attacchi di tipo SYN flood possono essere prevenuti mediante tecniche di *Delayed Binding* [74] (a.k.a. TCP Splicing), che rappresentano lo slittamento della connessione tra client e server, ritardando la creazione del canale di comunicazione fino a quando tutti i pacchetti di dati previsti nei passaggi del protocollo TCP sono stati ricevuti da entrambe le parti. In questo modo, il SYN flood viene bloccato, in quanto il server non apre alcuna half-open connection fino al ricevimento del pacchetto finale dal client, e questo avviene solo nel caso di richieste legittime che rispettano il protocollo standard TCP.

- Il *Bogon Filtering* è una tipologia di filtraggio del traffico che ricerca pacchetti specificanti, come origine o destinazione, indirizzi oscuri o appartenenti a porzioni di spazio IP non utilizzate. Gli indirizzi oscuri (dark addresses o indirizzi marziani) sono indirizzi ad uso speciale riservati alla IANA (Internet Assigned Numbers

Authority), l'autorità responsabile dell'assegnazione degli indirizzi internet, mentre gli altri rappresentano gli indirizzi non ancora allocati. Pertanto, un eventuale riscontro di tali indirizzi nei pacchetti dati ricevuti comporta, nella maggioranza dei casi, un tentativo di DoS, in cui l'autore intende nascondere la propria identità e far risultare che l'attacco sia originato da una fonte differente o fittizia, impedendo così la sua geolocalizzazione. Ciò è possibile grazie alle tecniche di spoofing: tipicamente il programma sceglie indirizzi di origine e destinazione in modo casuale dall'intero spazio IP, tuttavia i meccanismi più recenti e sofisticati conservano una lista degli indirizzi oscuri e di quelli non ancora allocati, in modo da evitare che gli IP generati ricadano in uno dei due insiemi [75].

- Grazie alle *Clean Pipes*, tutto il traffico di rete viene fatto passare attraverso un "centro di lavaggio" in cui l'utilizzo di vari metodi, quali proxy[17] e tunnel[18], permette un ulteriore separazione del traffico maligno da quello legittimo, e consentono solamente a quest'ultimo di superare il centro di controllo e arrivare al server. Fra le più grandi compagnie informatiche realizzatrici di Clean Pipes vi sono Verisign, Arbor Networks e AT&T.

- Per contrastare gli attacchi DoS che sfruttano il livello 7 del modello OSI, tra i quali il diffuso HTTP POST attack, di cui a p.42, si può specificare un intervallo di tempo massimo al termine del quale il server interrompe la comunicazione con il client, indipendentemente dalla legittimità della richiesta. I server Apache[19], ad

[17]Proxy: elemento di rete che si interpone fra un client e un server. Il client, anziché inviare le richieste direttamente al server, le invia al proxy, il quale ha il compito di inoltrarle al server, e restituire infine la risposta al client [76].

[18]Tunnel: *"an intermediary program which is acting as a blind relay between two connections"* [76]

[19]Apache: è una delle piattaforme Web Server più diffuse al mondo.

esempio, consentono la gestione di questa impostazione tramite l'estensione *TimeOut* [77]. La direttiva:

```
TimeOut 45
```

interrompe ogni connessione in cui il client impiega più di 45 secondi per completare l'invio del pacchetto.

- *Deep Packet Inspection*: p. 21.

In aggiunta al filtraggio del traffico, la gestione dello scheduling di rete[20] può rappresentare un eccellente metodo di difesa dagli attacchi DDoS. L'idea centrale è l'utilizzo di una *priority queue*, una coda caratterizzata da un livello di priorità associato ad ogni suo elemento. Gli elementi a priorità più elevata hanno la precedenza sugli altri: si determina così uno scenario di prelazione (*preemption*), in cui l'elaborazione della richiesta corrente può essere interrotta a favore di una seconda richiesta pervenuta successivamente, ma con priorità superiore. Distinguendo il traffico malevolo da quello legittimo, si può dirottare il primo verso la coda a priorità bassa (di tipo *Tail Drop*[21]) e il secondo verso quella a priorità alta, permettendo ai clienti di usufruire del servizio anche se il sistema è sotto attacco. Lin, C.-H. et al. [78] propongono uno schema *Double Check Priority Queue*, strutturato nel seguente modo:

1. *Source analysis*: al ricevimento di un pacchetto, il server controlla se ha già ricevuto altri dati dalla stessa fonte. In caso negativo, la probabilità che il pacchetto appartenga a un flusso di traffico malevolo è molto bassa, pertanto il server indirizza il pacchetto verso la coda ad alta priorità, destinata agli utenti legittimi, e

[20]Network scheduler: programma di rete che gestisce il traffico in entrata e in uscita, sfruttando apposite code per amministrare le contese dei pacchetti di dati che richiedono la propria elaborazione in modo concorrente.

[21]Tail drop: algoritmo di scheduling che prevede il rifiuto automatico di ogni pacchetto in ingresso, nel caso in cui la coda sia piena.

crea una struttura dati in cui inserisce indirizzo del mittente, tempo di arrivo e altri dati di rilevazione relativi al pacchetto. In caso affermativo, invece, gli scenari possibili sono due: se la fonte è risultata sospetta, in precedenti iterazioni dell'algoritmo, il pacchetto viene direttamente inviato alla coda a bassa priorità; in caso contrario, si procede con la seconda fase del controllo: l'analisi degli intervalli.

2. *Packet Intervals Analysis*: lo scheduler di rete calcola la soglia (*threshold*) al di sopra della quale il traffico è considerato malevolo, e le medie armoniche[22] relative ai tempi di arrivo degli ultimi pacchetti. La formula proposta per il calcolo della threshold è:

$$T = C\frac{b}{n} \tag{3.1}$$

dove T è tale soglia, b è la banda di rete, n è il numero di utenti correnti e C è una costante regolabile in un range di valori (da 1,5 a 10).
Stabilita la soglia di traffico del sistema, l'algoritmo calcola le medie armoniche:

$$H_{t12}(t) = \frac{2}{\frac{1}{t_1} + \frac{1}{t_2}} \tag{3.2}$$

$$H_{t23}(t) = \frac{2}{\frac{1}{t_2} + \frac{1}{t_3}} \tag{3.3}$$

quindi ricava la differenza tra le due medie:

$$H_{diff} = H_{t23}(t) - H_{t12}(t) \tag{3.4}$$

e utilizza il valore ottenuto per determinare l'*incoming packets rate*, il tasso di pacchetti in ingresso, dato dal reciproco della

[22] La media armonica di n termini è definita come il reciproco della media aritmetica dei reciproci degli n termini. È notevolmente influenzata dagli outlier piccoli, piuttosto che da quelli grandi, pertanto è adatta allo scenario descritto, in cui la presenza di traffico malevolo è direttamente collegata al rilevamento di intervalli di tempo minimi fra un pacchetto di dati e quello successivo.

media armonica. Infine, il tasso rilevato viene confrontato con il valore di *threshold*. Se il tasso è maggiore, significa che il sistema sta gestendo un traffico assai più elevato della norma, e si trova probabilmente sotto attacco: il pacchetto viene quindi indirizzato verso la coda a bassa priorità. In caso contrario, il sistema è stabile e il pacchetto viene inoltrato alla coda ad alta priorità.

Ultime, ma non meno importanti, alcune strategie preventive proposte dalla CSA in ambito gestionale [52]:

- *Capacity/Resource Planning*: effettuare proiezioni e stime di eventuali necessità di risorse extra, per ridurre il rischio di inadempimento contrattuale in caso di sovraccarichi, valutando attentamente il trade-off tra margine di sicurezza e dispendio di risorse.
- *Equipment Power Failures*: implementare meccanismi di ridondanza per garantire la disponibilità delle informazioni anche in caso di interruzioni del servizio relative a una parte dell'infrastruttura.
- *Application Security*: valgono le stesse politiche esposte a p. 34.

3.1.6 Malicious Insiders

Bisogna considerare l'eventualità che i rischi non sopraggiungano dall'esterno del sistema, bensì nascano direttamente all'interno dello stesso. Il CERT[23] definisce un "insider threat" nel seguente modo: *"A malicious insider threat to an organization is a current or former employee, contractor, or other business partner who has or had authorized access to an organization's network, system, or data and intentionally exceeded or misused that access in a manner that negatively affected the*

[23]CERT: Divisione della Sicurezza della SEI, importante azienda nel campo della cybersecurity, avente come missione principale la coordinazione delle risposte ai problemi di sicurezza informatica [79].

confidentiality, integrity, or availability of the organization's information or information systems" [80].
I motivi che spingono un dipendente ad arrecare danni immateriali all'azienda in cui lavora sono principalmente di tipo economico: vendere informazioni private può essere molto remunerativo, così come rivelare parti del patrimonio informativo aziendale a un'azienda competitrice.
Contromisure:

- *Third Party Audits*: provider esterni a cui si affida l'azienda devono essere periodicamente sottoposti ad analisi, per assicurarsi che mantengano conformità con i protocolli di sicurezza, di confidenzialità e di qualità del servizio, come stabilito da contratto [52] (se un'azienda X si affida a un provider esterno Y per la fornitura di uno o più servizi ai propri clienti, è necessario che la qualità del servizio offerta da Y sia conforme a quella degli SLA dell'azienda X).

- *Information Leakage*: valgono le stesse strategie esposte a p. 28.

- *User Access, Unauthorized Persons Entry & Off-Site Authorization* [52]: l'accesso fisico ai patrimoni informativi aziendali da parte di utenti e del personale di supporto deve essere soggetto a restrizioni. I punti di ingresso e di uscita dell'edificio e altre aree a cui il personale non autorizzato può accedere (e.g.: aree di servizio) devono essere monitorate e isolate dall'infrastruttura di conservazione dei dati sensibili. Inoltre, qualsiasi trasferimento di risorse hardware o software in altri edifici deve ottenere un'autorizzazione prima di poter essere effettuato.

- *Human Resources Background Screening* [52]: tutti i candidati e le terze parti devono essere soggetti a verifica del proprio background, con una profondità di controllo proporzionale alla classe di privilegi di cui potranno disporre.

- *Roles, Responsibilities & Segregation of Duties* [52]: è opportuno documentare i ruoli e le responsabilità di impiegati, contraenti e terze parti in relazione alla gestione del patrimonio informativo e della sicurezza. Stabilire politiche e procedure di separazione dei compiti per evitare che la responsabilità di un intero settore aziendale ricada nelle mani di un singolo dipendente.

- *Encryption*: si applicano tecniche di crittografia per la protezione dei dati sensibili (pp. 22-23).

- *USB Devices Disabling/Restriction*: sebbene il firewall, le impostazioni di rete e i sistemi di monitoragio blocchino il traffico non autorizzato in uscita, per impedire fughe di dati, cosa impedisce a un dipendente di trasferire fisicamente i dati desiderati su un dispositivo di memorizzazione rimovibile, quale una chiave USB? In riferimento a Windows Seven, si può disabilitare il riconoscimento di memorie USB non ancora installate nel sistema accedendo alla directory:

  `%windir%\inf`

  e modificando i permessi degli utenti relativi ai file *usbstor.inf* e *usbstor.pnf*.
  Per la disabilitazione di memorie USB già installate nel sistema, è necessario invece modificare il registro *Start* contenuto nella directory

  `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\`
  `Services\UsbStor`

  e impostare il suo valore a `4`.
  In alternativa, si può impedire la sola scrittura su memorie USB, permettendo gli accessi in lettura. In tal caso, è sufficiente creare un nuovo registro di tipo "DWORD (32 bit)" nella directory:

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\StorageDevicePolicies`

rinominarlo *WriteProtect*, e impostare il suo valore a `1`.

3.1.7 Abuse of Cloud Services

Come già accennato a p. 4, uno dei principali benefici del Cloud Computing consiste nella possibilità, per medie o piccole imprese, di avere accesso a enormi quantità di capacità computazionale, in modo relativamente semplice. Tuttavia, questa opportunità può essere sfruttata dagli hacker a scopi malevoli, ad esempio per decodificare una chiave crittografica rubata – operazione che altrimenti, con risorse hardware limitate, richiederebbe anni – per inizializzare un attacco DDoS, o per la diffusione di malware e spam [47]. La domanda da porsi, in questo caso, è "in che modo si possono identificare gli utenti che abusano dei servizi offerti?". Sfortunatamente, al di là delle operazioni di monitoraggio e dei protocolli per la verifica delle identità, non si può disporre di metodi accurati per prevedere abusi dei servizi Cloud. E' opportuno, tuttavia, definire dettagliatamente gli utilizzi accettabili del patrimonio informativo e dell'infrastruttura concessa come servizio, e stabilire procedure legali da intraprendere in caso di violazioni dei protocolli.

3.1.8 Insufficient Due Diligence

L'espressione inglese "due diligence" rappresenta *"il procedimento attraverso il quale il potenziale acquirente valuta in via preventiva le condizioni economiche della società che intende acquisire, nonché i rischi eventuali e potenziali correlati a determinate operazioni"* [81]. Contestualizzato nel caso specifico del Cloud Computing, il termine indica anche l'insieme di attività svolte da un'azienda interessata al trasferimento sulla nuvola, allo scopo di conoscere gli eventuali rischi dell'o-

perazione, le potenzialità future dell'investimento e giungere quindi a una valutazione finale. Trascurando tali analisi, sopraggiunge il rischio di una mancata corrispondenza fra le aspettative del cliente e i servizi effettivamente offerti dal provider (e.g.: scarsa gestione degli incidenti, assenza di garanzie crittografiche, limitato monitoraggio della sicurezza [47]), sia nel caso in cui l'azienda decida di entrare nell'ambiente Cloud in qualità di cliente, sia nel caso in cui vi si inserisca come venditore.
Contromisure:

- *Risk Assessments (2)* [52]: Effettuare periodicamente analisi dei rischi, determinando la probabilità che tali rischi si trasformino in danni concreti, e valutare il loro impatto sul sistema, attraverso metodi qualitativi e quantitativi.

- *Industry Knowledge / Benchmarking* [52]: effettuare analisi comparative fra le conoscenze aziendali sulla sicurezza e quelle di specialisti di sicurezza esterni e associazioni professionali.

- *Capacity/Resource Planning*: valgono le stesse politiche esposte a p. 47.

- *Resiliency Management Program, Impact Analysis & Business Continuity Planning*: valgono le stesse politiche esposte a p. 31.

3.1.9 Shared Technology Vulnerabilities

La natura distribuita dei sistemi Cloud, con la conseguente condivisione di risorse e tecnologie, aggrava l'impatto delle violazioni di sicurezza sul sistema, estendendo i rischi a tutti i sistemi ad esso connessi: la compromissione di un singolo elemento condiviso può minare l'integrità dell'intero sistema Cloud, e, di conseguenza, la salvaguardia dei dati di tutti i clienti che utilizzano i suoi servizi [47].
Contromisure:

- *Network Segmentation & Segregation* [52] [82]: segmentazione e segregazione della rete sono forse i controlli più efficaci per l'attenuazione dei danni causati dalle intrusioni, e in particolare dalla loro fase successiva: la propagazione (a.k.a. *lateral movement*). La segmentazione consiste nel partizionare la rete in più sottoreti, mentre la segregazione prevede l'applicazione di politiche che stabiliscano a quali dispositivi del sistema è consentito comunicare con altri dispositivi, e, in tal caso, con quali dispositivi. I principali metodi adoperati sono l'utilizzo di VLAN, firewall (sia a livello fisico sia a livello applicativo), filtri del traffico, reti unidirezionali (*data diodes*[24]), proxy non trasparenti[25], servizi di autenticazione/autorizzazione e principio del privilegio minimo[26].

- *Encryption*: si applicano tecniche di crittografia per la protezione dei dati sensibili (pp. 22-23).

- *Patch Management*: i sistemi Cloud necessitano costanti analisi e manutenzioni a tempo di esecuzione. Non appena vengono scoperte nuove vulnerabilità, è necessario applicare dinamicamente una patch che risolva il problema [4], modificando una o più parti

[24]Data diodes (unidirectional security gateways): dispositivi di rete che consentono il trasferimento dei dati in un'unica direzione. Nell'ambito dei sistemi di sicurezza, sono utilizzati come connessioni fra due o più reti di diversa classe [83]. In tal modo, si evita che la compromissione di una singola area del sistema permetta all'hacker di ottenere anche il controllo delle aree più classificate (viceversa, la compromissione di un'area top-class permette la contaminazione di tutti i settori inferiori).

[25]Un proxy è definito "trasparente" se non modifica le richieste e risposte inoltrate, altrimenti è detto "non trasparente", e può applicare filtraggi e protocolli stabiliti dall'amministratore del sistema, per determinare quali richieste inoltrare e quali rifiutare, o limitare l'ampiezza di banda concessa al client [76].

[26]Principle of least privilege: *"Every program and every user of the system should operate using the least set of privileges necessary to complete the job. Primarily, this principle limits the damage that can result from an accident or error. It also reduces the number of potential interactions among privileged programs to the minimum for correct operation, so that unintentional, unwanted, or improper uses of privilege are less likely to occur"* [84].

strutturali/comportamentali del sistema. È opportuno, pertanto, stabilire politiche per l'applicazione di patch in modo tempestivo. Solarwind Patch Management [85] è un software professionale, utilizzato da colossi mondiali quali Microsoft, Cisco, Visa, Mastercard, AT&T e JP Morgan, che permette la gestione di decine di migliaia di dispositivi, consentendo di monitorare lo status dell'intera infrastruttura aziendale, da una o più unità di controllo, e avviare scansioni, installazioni e disinstallazioni sincronizzate di patch e applicazioni.

Capitolo 4

Intrusion Detection Systems

Il rilevamento delle intrusioni è il processo di monitoraggio e di analisi degli eventi di un sistema informatico, allo scopo di individuare segni di intrusione [86], ossia tentativi di elusione dei meccanismi di sicurezza e di compromissione della confidenzialità, dell'integrità o della disponibilità di un componente del sistema. Gli IDS sono composti da uno o più sensori per la ricezione di informazioni dalla rete e dagli host, un motore che analizza i dati prelevati dai sensori, un database contenente i set di regole sulla base delle quali il motore di analisi elabora le informazioni, e una console per il monitoraggio delle anomalie del sistema. Le tipologie di informazioni utili al rilevamento delle intrusioni sono:

- Informazioni di configurazione relative allo stato corrente del sistema.
- Informazioni di revisione, come l'elenco delle attività del sistema (e.g.: il *registro degli eventi* di Microsoft, il *syslogd* di Linux).
- Informazioni specifiche a seconda delle conoscenze di base degli attacchi informatici.

Il sistema compila una lista completa delle azioni effettuate dagli utenti; quindi invia un allarme se una o più di esse rappresentano segni di intrusione.

L'efficienza di un IDS viene misurata in base ad alcuni parametri, tra cui:

- *Accuratezza* [87]: in inglese Accuracy, è la capacità di un IDS di riconoscere correttamente i segni di intrusione e limitare il numero di falsi positivi. Il rilevamento delle intrusioni ha infatti quattro possibili esiti: "true negatives" e "true positives" corrispondono al corretto funzionamento dell'IDS e identificano, rispettivamente, eventi legittimi e segni di intrusione; "false positives" e "false negatives" corrispondono invece al funzionamento impreciso dell'IDS, e identificano, rispettivamente, eventi legittimi scambiati per segni di intrusione, e segni di intrusione scambiati per eventi legittimi. Le seguenti equazioni calcolano il tasso di TN, TP, FN, FP e quantificano l'accuratezza dell'IDS [88] [89]:

$$TP_{rate} = \frac{DetectedAttacks}{ObservableAttacks} = \frac{TP}{TP + FN} \tag{4.1}$$

$$TN_{rate} = \frac{TrueAlerts}{TotalAlerts} = \frac{TN}{TN + FP} \tag{4.2}$$

$$FP_{rate} = \frac{FP}{TN + FP} = 1 - TN_{rate} \tag{4.3}$$

$$FN_{rate} = \frac{FN}{TP + FN} \tag{4.4}$$

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{4.5}$$

- *Performance* [89]: è la velocità di processazione degli eventi rilevati. Una scarsa performance impedisce rilevazioni in tempo reale, e permette all'hacker di compromettere il sistema di rilevamento e cancellare le proprie tracce, prima che queste siano analizzate.

- *Elasticità* [90]: un efficace IDS deve essere in grado di resistere agli attacchi – DoS in primis – e tollerare errori parziali.

Una prima categorizzazione degli IDS può essere effettuata in base all'approccio utilizzato per il rilevamento delle intrusioni [89]:

- *Knowledge/signature-based*: si sfruttano le conoscenze disponibili riguardo agli attacchi informatici e alle relative modalità di attuazione: quando il sistema rileva un set di azioni che corrisponde a un modello di attacco conosciuto, lancia un allarme. Sono ammesse quindi tutte e sole le azioni non esplicitamente riconosciute come parti di attacchi. Gli IDS knowledge-based possono raggiungere alti livelli di precisione, e generano un numero esiguo di falsi positivi, ma funzionano correttamente solo se il database informativo viene aggiornato regolarmente, per permettere al sistema di stare al passo con le offensive più recenti.

- *Behaviour/anomaly-based*: si assume che un'intrusione possa essere rilevata osservando le deviazioni rispetto al comportamento previsto del sistema e degli utenti. Nella prima fase di distribuzione, l'IDS "impara" a riconoscere le azioni legittime degli utenti e il corretto funzionamento del sistema, registrando un insieme di modelli comportamentali ammessi; nella seconda fase, lancia un allarme ogni volta che il modello comportamentale controllato non corrisponde a nessuno di quelli previsti. Il vantaggio degli IDS basati sul comportamento è dato dalla possibilità di contribuire al rilevamento di nuove tipologie di attacchi e di individuare abusi di privilegi, tuttavia si ottiene un tasso di falsi positivi piuttosto elevato, poiché, nella fase iniziale di apprendimento, il sistema potrebbe fallire nel tentativo di riconoscere tutti i casi comportamentali legittimi. In secondo luogo, il comportamento del sistema monitorato può cambiare nel tempo, forzando di fatto nuovi periodi di riapprendimento in cui l'IDS non è pienamente operativo. Infine, se il sistema è sotto attacco nella fase in cui avviene l'analisi dei modelli comportamentali accettabili, l'IDS non riuscirà a riconoscere parti del traffico malevolo come intrusioni poiché queste risulteranno incluse nel registro dei comportamenti leciti.

Una seconda suddivisione degli IDS è effettuata in base a quali dispositivi informatici sono analizzati durante il rilevamento delle intrusioni:

- *Host-based IDS*: sono installati su un host per monitorare gli eventi che si verificano al suo interno. Presentano un limitato tasso di falsi positivi, e sono indicati per l'analisi dei dati criptati. Un esempio di IDS open source basato su host è OSSEC [91], che esegue analisi dei registri, controllo dell'integrità dei file, monitoraggio delle politiche di sicurezza del sistema e ricerca di rootkit, garantendo reazione in tempo reale (ulteriori approfondimenti alle pp. 62-65).

- *Network-based IDS*: analizzano i pacchetti dati di rete, risultando particolarmente efficaci nel rilevare tentativi di intrusione. Tuttavia, solitamente hanno un alto tasso di falsi positivi, e non possono analizzare informazioni criptate. Le principali attività anomale ricercate sono: accessi non autorizzati, abuso di privilegi, intercettazione del traffico e indicatori di DoS. Un esempio di IDS open source basato su network è Snort [92], che implementa numerosi metodi di ispezione delle anomalie e dei protocolli di rete (ulteriori approfondimenti alle pp. 60-61).

- *Hybrid IDS*: sono IDS personalizzati, composti da sottosistemi network-based e sottosistemi host-based, che forniscono sia analisi del traffico di rete, sia analisi di uno o più host. Spesso sono affiancati da software gestionali in grado di monitorare i sottosistemi e gestire le interazioni. Un esempio di IDS ibrido open-source è Prelude [93] mentre ACARM-ng [94] è un manager di supporto che permette di riunire gli eventi simili in gruppi di attività fraudolente, riducendo la quantità di messaggi che necessitano supervisione manuale, allo scopo di facilitare l'analisi delle intrusioni e aumentare la reattività del sistema.

4.1 Data Mining

Il *data mining (a.k.a. Knowledge Discovery in Data, KDD)* è l'insieme delle pratiche che analizzano i sistemi e i database informativi allo scopo di estrarre modelli, schemi e tendenze, mediante l'utilizzo di algoritmi matematici che segmentano il set informativo e valutano le probabilità di eventi futuri [95]. Nadiammai, G.V. e Hemalatha, M. [96] propongono l'applicazione degli algoritmi di data mining ai sistemi di rilevamento delle intrusioni, individuando, quale miglior schema di apprendimento automatico, il *Random Forest*. Tale algoritmo, di tipo *ensemble learning*[1], consiste nell'utilizzare più alberi decisionali non correlati, e calcolare la distribuzione di frequenza degli output dei singoli algoritmi. Altri algoritmi di uso frequente sono il *k-Means*, uno schema iterativo che partiziona il sistema in k gruppi di elementi *(training vectors)*, il *kNN (k-Nearest Neighbors)*, in cui si valuta la classe di un elemento in base ai k elementi più vicini, e l'*SVM (Support Vector Machine)*, che stabilisce la migliore funzione di classificazione degli elementi, tracciando l'iperpiano con più margine da entrambe le classi [98].

4.2 Intrusion Prevention Systems

A causa del tasso di falsi positivi non trascurabile, solitamente gli IDS sono passivi, ovvero si limitano a registrare la presenza di anomalie all'interno del sistema, e lasciare che sia l'amministratore ad effettuare operazioni di pulizia laddove siano ritenute necessarie. Tuttavia, se il tasso di falsi positivi è particolarmente limitato, l'IDS può essere implementato in modo attivo, e in tal caso prende il nome di Intrusion Prevention System (IPS): in questo scenario, il sistema di difesa, oltre a rilevare le intrusioni, agisce proattivamente, ad esempio interrompendo una connessione o riprogrammando il firewall, nel tentativo di escludere l'hacker dal sistema o respingere il suo attacco [99].

[1] Combinare diversi algoritmi di apprendimento per ottenere migliori performance di predizione rispetto a quelle derivanti dall'uso separato dei singoli algoritmi [97].

Per passare dalla teoria alla pratica, vedremo ora le principali caratteristiche e alcune funzionalità di due IPS già citati: Snort e OSSEC.

4.2.1 Snort

Snort è un knowledge-based NIPS *(Network Intrusion Prevention System)*. Il dispositivo su cui viene installato il software deve essere configurato per il port mirroring, e la scheda di rete deve essere in modalità promiscua, permettendo al programma di intercettare e leggere ogni pacchetto di rete. La prima cosa da fare è abilitare la modalità di rilevamento delle intrusioni (dato che Snort può agire anche come semplice *packet logger*) [100]:

```
./snort -d -h 192.168.1.0/24 -l ./log -c snort.conf
```

dove `snort.conf` è il nome del file di configurazione del software, contenente le regole di analisi e il set di reazioni predefinite. Per aumentare ulteriormente il grado di sicurezza della rete, si può abilitare lo scarto *bad packets*, ovvero i pacchetti la cui *checksum*[2] risulti sospetta, tramite il comando:

`config enable_ttcp_drops`.

La modalità di generazione degli allarmi è configurabile grazie al comando `-A`. Le impostazioni possibili sono illustrate in tabella 3.1.

Relativamente all'aspetto delle notifiche, vediamo come interpretare un messaggio di allerta. Un esempio di avviso generato dal software è dato dal seguente log [100]:

```
[**] [116:56:1] (snort_decoder): T/TCP Detected [**]
```

Il primo numero è l'ID del "generatore", ovvero il componente che ha genera-

[2]Checksum: sequenza di bit associata a un pacchetto, utilizzata per verificare che il messaggio inviato non abbia subito alterazioni durante la trasmissione nel canale di comunicazione.

Opzione	Descrizione
-A fast	Scrive gli avvisi di allerta su disco, utilizzando un formato semplice contenente timestamp, messaggio descrittivo, indirizzi e porte di origine e di destinazione.
-A full	Scrive gli avvisi di allerta su disco, registrando tutte le informazioni possibili *(modalità standard)*.
-A unsock	Invia gli avvisi di allerta a un socket, per permettere al software di comunicare con un altro programma.
-A none	Disabilita gli avvisi di allerta.
-A console	Visualizza gli avvisi di allerta, in modalità fast, direttamente sulla console.
-A cmg	Visualizza gli avvisi di allerta, in modalità full, direttamente sulla console.

Tabella 4.1: Modalità di generazione degli avvisi di allerta in Snort [100]

to l'allerta. Il secondo valore indica il tipo di evento rilevato, mentre il terzo numero è l'ID di revisione relativo all'evento.
Per quanto riguarda le prestazioni, Snort offre la possibilità di configurare l'algoritmo di ricerca dei pacchetti: a fronte di una elevata quantità di memoria RAM, si può "richiedere" al programma una maggiore efficacia in termini di rapidità nell'analisi. La direttiva è:

```
config detection:  [search-method <method>]
```

dove `method` può assumere i valori `lowmem` *(very "low memory, moderate performance")*, `ac-bnfa` *("low memory, high performance")* e `ac` *("high memory, best performance")*.

4.2.2 OSSEC

OSSEC è un HIPS (Host Intrusion Prevention System) in grado di interpretare i registri del sistema, dei database e dei dispositivi di sicurezza installati sull'host (e.g.: Firewall, Antivirus, ricerca di rootkit), unificare le analisi effettuate in un unico log, e inviarlo alla console di monitoraggio, con possibilità di reagire in modo proattivo.
Il software prevede che un dispositivo svolga il ruolo di server/manager del sistema (conservando i database informativi, i registri degli eventi, le regole di reazione e le opzioni di configurazione), e che sugli altri host siano installati i tool di rilevamento del software, definiti agenti, in grado di collezionare informazioni utili e inviarle al server per l'analisi. Per il funzionamento del software, è necessario abilitare il traffico UDP in ingresso sulla porta 1514, utilizzata per la comunicazione fra il server e gli altri host.
Una delle funzioni principali di OSSEC è l'*Integrity Checking (syscheck)* [101], che controlla periodicamente eventuali modifiche ai registri di sistema, valutando le relative checksum. Quando un agente effettua la scansione del sistema, invia tutte le checksum rilevate al server, il quale confronta i nuovi dati con quelli conservati nel database, e invia una notifica di allerta nel caso in cui una o più checksum non dovessero corrispondere, indicando il rilevamento di modifiche dei file.
La ricerca di rootkit è un'altra operazione di base effettuata dal software. Come il syscheck, viene eseguita a intervalli regolari (personalizzabili) e, operando in modalità knowledge-based (sulla base di un database di rootkit conosciuti) effettua una scansione del sistema in cerca di [101]:

- *processi nascosti*: grazie all'utilizzo delle funzioni *getsid()* e *kill()* si controlla se esistono pid utilizzati. In caso positivo, se il comando per mostrare i processi attivi (*ps* in Linux, *tasklist* in Windows) non è in grado di rilevarli, significa che nel sistema è presente un rootkit a livello kernel;

- *porte segrete*: con la funzione bind() si controlla ogni porta TCP e UDP

del sistema. Se il binding relativo a una porta fallisce (il che significa che tale porta è attualmente utilizzata), ma *netstat*, il comando che permette di visualizzare le connessioni attive del computer, non è in grado di rilevarla, è probabile la presenza di un rootkit all'interno del sistema;

- *interfacce di rete*: se una o più interfacce sono in modalità promiscua, e il comando *ifconfig* non lo evidenzia, è probabile che un rootkit stia nascondendo la presenza di uno o più dispositivi che intercettano tutti i pacchetti di rete;
- in generale, modifiche non desiderate alle impostazioni di sistema.

Un'altra interessante estensione del software permette di realizzare la disabilitazione distribuita delle memorie USB, per impedire ai dipendenti di appropriarsi in modo illecito dei contenuti del patrimonio informativo aziendale. Le soluzioni di cui a p. 49 sono quindi sconsigliabili nel caso in cui il sistema sia composto da un elevato numero di elaboratori, ed è preferibile utilizzare un software di monitoraggio distribuito, quale, per l'appunto, un IDS. La soluzione seguente [101] fa riferimento a un sistema operativo Windows.
Innanzitutto, è necessario monitorare il registro *USBSTOR*, contenente la lista completa dei dispositivi USB installati sull'elaboratore, mediante il comando *reg*:

```
1 <agent_config os="windows">
2     <localfile>
3         <log_format>full_command</log_format>
4         <command>reg QUERY HKLM\SYSTEM\CurrentControlSet\
              Enum\USBSTOR</command>
5     </localfile>
6 </agent_config>
```

In seguito, si sfrutta la funzione `check_diff`, che verifica se l'output di un comando sia cambiato rispetto allo stato precedente, e la si applica al comando di monitoraggio del registro USBSTOR.

```
1 <rule id="140125" level="7">
2     <if_sid>530</if_sid>
3     <match>ossec: output: 'reg QUERY</match>
4     <check_diff />
5     <description>New USB device connected</description>
6 </rule>
```

Ogni eventuale tentativo di utilizzare un dispositivo USB modificherà il registro stesso, con conseguente innesco della funzione `check_diff`, la quale provvederà a generare un messaggio di allerta contenente i dettagli relativi al dispositivo utilizzato e al profilo dell'utente.
Un esempio di avviso, in cui si osserva una nuova entry nella lista dei dispositivi installati:

```
** Alert 1268687754.35062: mail  - local,syslog,
2010 Mar 15 18:15:54 (xx-netbook) any->reg QUERY
   HKLMSYSTEMCurrentControlSetEnumUSBSTOR
Rule: 140125 (level 7) -> 'New USB device connected'
Src IP: (none)
User: (none)
ossec: output: 'reg QUERY
   HKLMSYSTEMCurrentControlSetEnumUSBSTOR':! REG.EXE
   VERSION 3.0

HKEY[...]USBSTOR
HKEY[...]USBSTORDisk&Ven_&Prod_USB_Flash_Memory&Rev_5.00
HKEY[...]USBSTORDisk&Ven_Generic&Prod_Flash_Disk&Rev_8.0
HKEY[...]USBSTORDisk&Ven_Hitachi&Prod_HTS543225L9&Rev_11
HKEY[...]USBSTORDisk&Ven_LEXAR&Prod_JD_FIREFLY&Rev_1100
HKEY[...]USBSTORDisk&Ven_SAMSUNG&Prod_HM160JC&Rev_0000
HKEY[...]USBSTORDisk&Ven_Sony&Prod_DSC&Rev_1.00
HKEY[...]USBSTORDisk&Ven_TomTom&Prod_ONE_XXL_IQ_Rts
HKEY[...]USBSTORDisk&Ven_USB_2.0&Prod_USB_FDrive&Rev_0.0

Previous output:
```

```
ossec: output: 'reg QUERY
   HKLMSYSTEMCurrentControlSetEnumUSBSTOR':
! REG.EXE VERSION 3.0
HKEY[...]USBSTOR
HKEY[...]USBSTORDisk&Ven_&Prod_USB_Flash_Memory&Rev_5.00
HKEY[...]USBSTORDisk&Ven_Generic&Prod_Flash_Disk&Rev_8.07
HKEY[...]USBSTORDisk&Ven_Hitachi&Prod_HTS543225L9&Rev_11
HKEY[...]USBSTORDisk&Ven_SAMSUNG&Prod_HM160JC&Rev_0000
HKEY[...]USBSTORDisk&Ven_Sony&Prod_DSC&Rev_1.00
HKEY[...]USBSTORDisk&Ven_TomTom&Prod_ONE_XXL_IQ_Rts
HKEY[...]USBSTORDisk&Ven_USB_2.0&Prod_USB_FDrive&Rev_0.0
```

Capitolo 5

Disaster Recovery Systems

Le tecniche di *Disaster Recovery* assicurano la salvaguardia delle informazioni e la continuità dei servizi in situazioni critiche, grazie alla realizzazione di un sistema di memorizzazione ad alta affidabilità [102]. Le tecnologie di recupero di ultima generazione possono essere suddivise in:

- *Storage Layer Disaster Recovery*:
  - implementate sulla base di specifici dispositivi di memorizzazione (e.g.: IBM PPRC [103], Hitachi TrueCopy [104]). Ogni scrittura sul volume principale viene eseguita anche su un volume secondario, disponibile in rete tramite accesso remoto.
  - basate su virtualizzazione: il sistema di memorizzazione è interamente virtualizzato all'interno di un lotto di memoria, il quale può essere copiato e replicato a piacimento (e.g.: FalconStor NSS [105]).
- *Application Layer Disaster Recovery*: tecnologie che permettono il salvataggio di applicazioni e database (e.g.: Oracle Data Guard [106]), tramite il backup delle operazioni SQL e altri metodi di scansione dei registri applicativi.

Il processo di Disaster Recovery è diviso in due stadi: il recupero delle informazioni e la ricostruzione del servizio. Fino ad alcuni anni fa, il secondo

processo non era solito iniziare finché il primo non fosse completamente terminato, allo scopo di garantire piena operatività del sistema e la medesima qualità del servizio. Di recente, e con particolare riferimento alla strategia proposta da Zheng, W. e Fang, B. [102], è avanzata l'idea di eseguire entrambe le fasi del recupero in modo parallelo. Applicando granularità fine all'intero processo, si può suddividere la fase di ricostruzione in N parti, $S_1, S_2, \ldots S_n$. Risulta evidente che l'abilitazione di una singola parte del servizio non necessiti il recupero del set informativo totale, bensì di una sua frazione. L'esecuzione di S_i, quindi, dipende solo dal recupero di D_i, con $D_i \in D_{ALL}$, ossia l'insieme totale delle informazioni da recuperare. Rispetto al processo di recupero tradizionale, quello parallelo accelera la ricostruzione del servizio, ma può causare una degradazione momentanea del servizio, dato che il sistema viene riassemblato gradualmente. Allo scopo di misurare l'efficacia della strategia di recupero si possono considerare due fattori:

- il tempo necessario a completare il recupero delle informazioni, senza alcuna fornitura del servizio (*Object Recovery Time*, RTO);
- il tempo necessario a ristabilire il livello di servizio (*Back To Normal*, BTN).

La scelta migliore è valutare il trade-off tra BTN e qualità del servizio, e implementare una strategia equilibrata. In figura 5.1 è illustrato un esempio comparativo tra recupero tradizionale e recupero parallelo, nel caso in cui $N = 3$, mentre in figura 5.2 è riportato un algoritmo euristico [102] che permette di minimizzare i tempi di ripristino del servizio e quello di recupero, e al contempo assicurare sincronizzazione fra il recupero di una frazione del sistema e il relativo ripristino.

5.1 Erasure Coding

In aggiunta alla replicazione pura, un'altra forma di ridondanza è definita *Erasure Coding*, e consiste nel dividere un oggetto in n frammenti, e

Obiettivo: il tempo di esecuzione di $S_1, S_2, \ldots S_n$ e il tempo di recupero di D_{BTN} devono essere minimizzati.
Vincolo: D_i, da cui dipende S_i, deve essere recuperato prima dell'esecuzione di S_i.

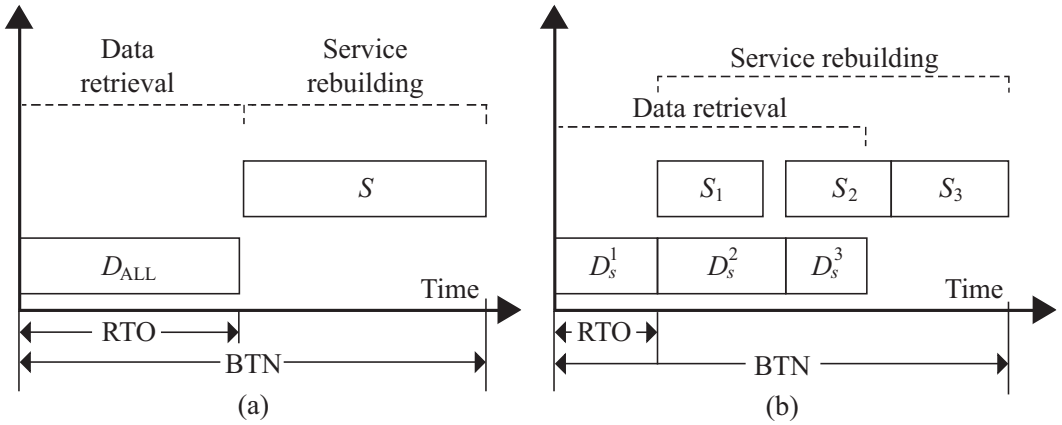

Figura 5.1: Recupero tradizionale (a) vs recupero parallelo (b).

Data retrieval

1. Initializazion: retrieve D_1
2. for $i = 2$ to N do
   retrieve $D_i - \sum_{k=1}^{i-1} D_k$
3. retrieve $D_{\text{BTN}} - \sum_{k=1}^{N} D_k$

Service rebuilding

1. S_1 executes when D_1 is ready
2. for $i = 2$ to N do
   S_i executes when S_{i-1} is finished and D_i is ready

Figura 5.2: Fast Recovery Algorithm

ricodificarli in m frammenti, con $m > n$, in grado di fornire informazioni aggiuntive utili al recupero dell'oggetto iniziale. Questa tecnica è più efficente della replicazione, poiché i frammenti combinati *(parity blocks)* permettono di raggiungere lo stesso livello di disponibilità dell'informazione con minore ridondanza.
Consideriamo un file composto da 2 frammenti – A e B – e supponiamo di effettuare due backup su server remoti: il primo con ridondanza semplice (replicazione), il secondo con schema Erasure Coding. Avremo quindi un server contenente 4 frammenti – A, A, B, B – e un server contenente i due frammenti originali – A, B – più due frammenti extra, che, ipotizziamo, siano A+B e A+2B, ovvero frammenti composti dalla somma logica dei codici binari dei frammenti originali. Ora, sempre per ipotesi, immaginiamo di dover far fronte a una perdita di informazioni: il file memorizzato sul server principale è perduto, e i due server di ripristino subiscono una perdita pari al 50% dei dati (2 frammenti su 4). Per quanto riguarda il server con ridondanza semplice, si ha esattamente il 50% di probabilità di non riuscire a recuperare il file (perdendo la coppia di A o la coppia di B). Nel caso del server in cui è stata applicata la strategia Erasure Code, la probabilità di fallire nel tentativo di recupero del file è pari a zero, poiché qualsiasi coppia di frammenti permette di risalire ai frammenti originali A e B che compongono il file. Lo scenario è definito *MDS (Maximum Distance Separable)* di tipo (4,2), e indica che, sulla base di quattro frammenti dati, è possibile risalire al file originale sfruttandone soltanto due. In figura 5.3 è illustrato un esempio pratico di schema MDS [107], nel quale, per comodità, il concetto di somma logica è sostituito da quello di somma algebrica. Il simbolo EC in figura indica l'Erasure Code, ossia l'algoritmo che stabilisce quali vincoli siano più efficaci e determina in che modo sia più opportuno ridistribuire i frammenti di dati. Il codice più utilizzato è il *Reed-Solomon*, che applica l'algebra lineare dei polinomi alla codifica degli oggetti. Lo schema personalizzato di Facebook *(HDFS RAID)* utilizza due codici di cancellatura:

- un XOR combinato, che consiste nella creazione di 1 parity block ogni

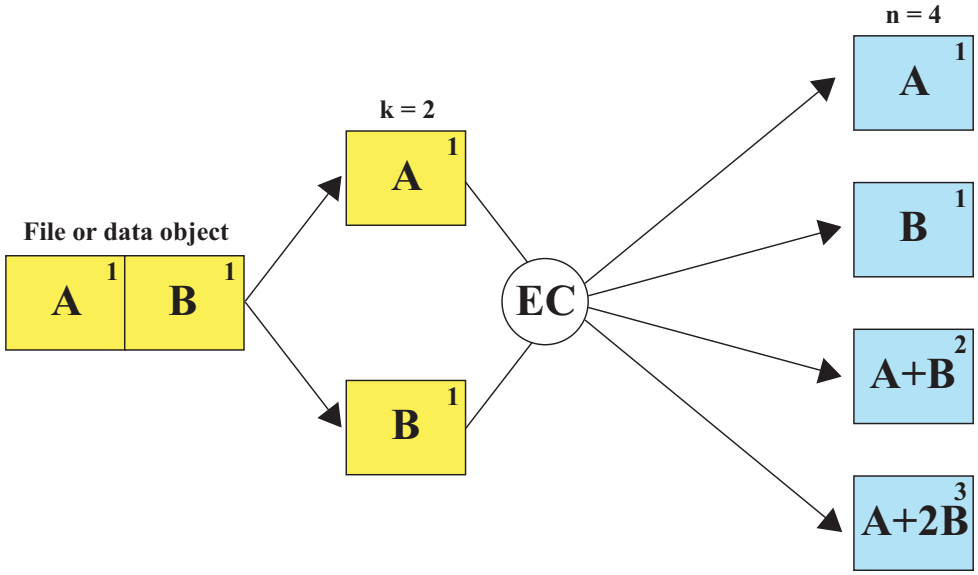

Figura 5.3: Esempio di ridondanza Erasure Code.

10 frammenti, e una replicazione a fattore 2x; è in grado di tollerare un errore parziale di 3 o 4 frammenti, a seconda dei casi, con un costo di memorizzazione aggiuntivo del 120%.

- un Reed-Solomon di tipo (14,10) in grado di tollerare un errore parziale di 4 frammenti, con un costo di memorizzazione aggiuntivo del 40%.

In figura 5.4, i costi relativi allo schema RS di Facebook sono confrontati con quelli di un modello di replicazione 3x, con tolleranza massima agli errori pari a 2 frammenti.

Il limite dei codici RS risiede nell'elevata quantità di traffico di rete necessaria al recupero delle informazioni, dato che, per ogni frammento corrotto, è necessario inviarne dieci (su quattordici) al server di backup, affinché questo possa ricostruire quello perduto. Per questo motivo, è necessario valutare il trade-off tra efficacia di recupero e performance della rete, determinando quali dati richiedano una codifica RS e per quali invece sia sufficiente una replicazione standard, più dispendiosa in termini di costi di memorizzazione, ma più leggera in termini di carico di rete.

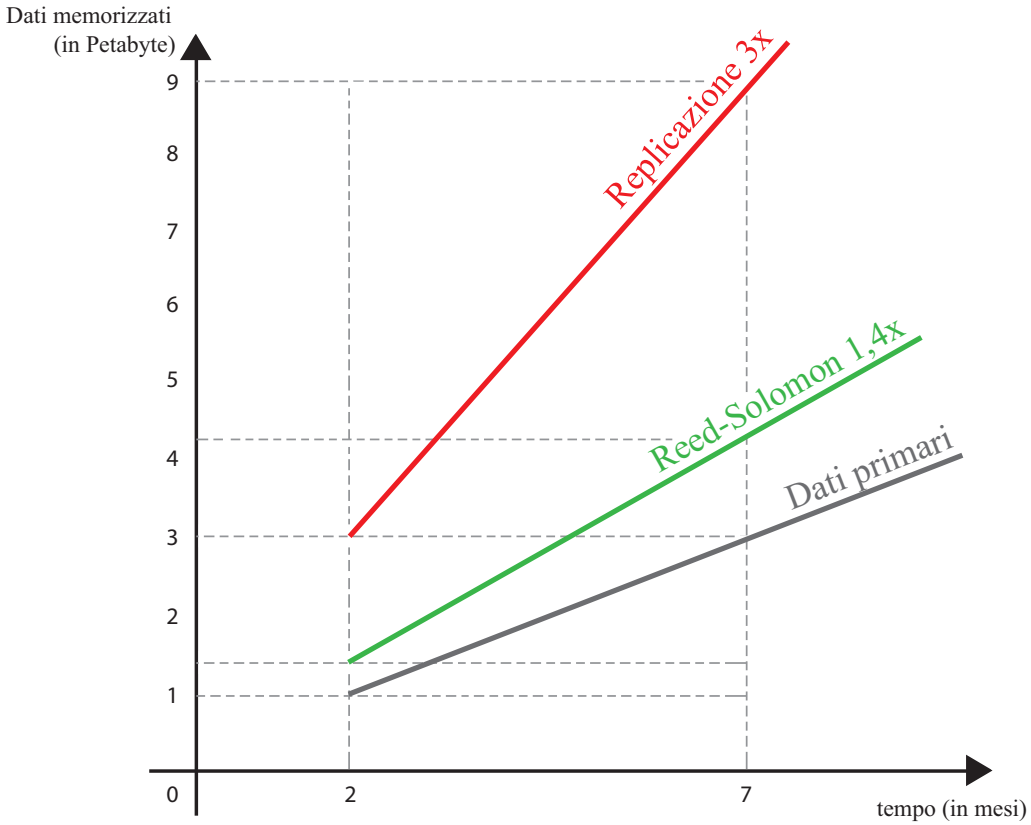

Figura 5.4: Confronto dei costi: replicazione vs Erasure Code

Capitolo 6

Conclusioni

Nel corso di questo lavoro si è evidenziata l'importanza della sicurezza nell'ambito del Cloud Computing, il quale, oltre ad aver ereditato alcune sfide dalla multiprogrammazione e dai sistemi distribuiti, deve continuamente evolversi per affrontare le minacce di ultima generazione (Denial of Service in primis) e offrire ai clienti un servizio solido e affidabile. A questo scopo, si implementano sistemi di rilevamento delle intrusioni, attraverso i quali è possibile riconoscere anomalie e minacce, e favorire la protezione della confidenzialità e dell'integrità delle informazioni memorizzate. Al contempo, per realizzare un valido sistema di sicurezza, è necessario valutare tutti i trade-off generati dai singoli componenti, e scegliere una soluzione equilibrata fra protezione, performance e costi, dato che un servizio scadente, seppur attendibile, è sgradito quanto uno vulnerabile.

Un'altra fra le principali sfide nell'ambito del Cloud Computing è relativa alla portabilità. "At present there is no standard which is universally followed by vendors to promote inter-operability. It is expected that the technology will mature and lead to a computer utility in which providers adhere to a common standard" [108]. Il fatto che l'utente sia legato a un particolare provider, e che questi spesso non rispetti gli standard strutturali e funzionali universalmente promossi, è una preoccupazione non trascurabile per chi desidera utilizzare la nuvola informatica: una maggiore diffusione di

standardizzazione e regole condivise allevierebbe le paure di molti clienti.

In ambito di resilienza, la missione attuale consiste nella ricerca di nuovi algoritmi di ridondanza che permettano di ridurre ulteriormente i costi di memorizzazione, o che, in alternativa, aumentino la capacità di recupero dei sistemi in situazioni critiche, senza sovraccaricare il traffico di rete.

Infine, è presumibile che il futuro della sicurezza virtuale sia legato ad un maggiore utilizzo dell'autenticazione multifattore, con particolare riferimento alle tecnologie destinate all'utenza, il cui principale metodo per proteggere i dati sensibili è l'utilizzo delle password, non più sufficienti a contrastare lo stato dell'arte dell'hacking malevolo.

Riferimenti

[1] Foster, I., Zhao, Y., Raicu, I., Lu, S., 2008: Cloud Computing and Grid Computing 360-Degree Compared, GCE (Grid Computing Environments) Workshop '08, Austin, TX, 12-16 November 2008.

[2] Rajaraman, V., 2014: Cloud Computing, Resonance – Journal of Science Education, March 2014, pp. 242-258.

[3] Cisco, Introduction to Virtual LANs.
`http://www.cisco.com/c/en/us/tech/lan-switching/virtual-lans-vlan-trunking-protocol-vlans-vtp/index.html`
Accessed 2 August 2014.

[4] Hill, R., Hirsch, L., Lake, P., Moshiri, S., 2013: Guide to Cloud Computing, Principles and Practice, Springer, pp. 3-4, 227-232.

[5] Sun, W., Zhang, K., Chen, S.K., Zhang, X., Liang, H., 2007: Software as a Service: An Integration Perspective, Service-Oriented Computing – ICSOC 2007, 5th International Conference, Vienna, Austria, 17-20 September 2007, p. 558.

[6] Magoules, F., 2009: Fundamentals of Grid Computing: Theory, Algorithms and Technologies, Chapman & Hall, pp. 131-132.

[7] Dean, J., Ghemawat, S., 2004: MapReduce: Simplified Data Processing in Large Clusters, OSDI '04, 6th Symposium on Operating System Design and Implementation, San Francisco, CA, December 2004.
`http://research.google.com/archive/mapreduce.html`

[8] IBM, What is MapReduce?
http://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/
Accessed 13 August 2014.

[9] Furht, B., Escalante, A., 2010: Handbook of Cloud Computing, Springer, pp. 12-24.

[10] Bartels, A., Rymer J.R., Staten, J., 2014: The Public Cloud Market Is Now In Hypergrowth: Sizing The Public Cloud Market, 2014 to 2020, Forrester, April 24, 2014.

[11] Il Sole 24 Ore, L'Italia crede nel Cloud Computing.
http://www.ilsole24ore.com/art/tecnologie/2013-06-26/litalia-crede-cloud-computing-105505.shtml
Accessed July 16, 2014.

[12] Dutta, A.K., Hasan, R.: How Much Does Storage Really Cost? Towards a Full Cost Accounting Model for Data Storage, published on Economics of Grids, Clouds, Systems and Services by Altmann, J., Vanmechelen, K. And Rana, O.F. (Eds.), GECON 2013.

[13] Filecloud Blog, Google Cloud Services vs Amazon Web Services (AWS) vs Microsoft Azure, Which suits you better?
http://www.getfilecloud.com/blog/2014/05/google-cloud-services-vs-amazon-web-services-aws-vs-microsoft-azure-which-suits-you-better/
Accessed September 6, 2014.

[14] Google Cloud Platform, Cloud Storage.
https://cloud.google.com/products/cloud-storage/
Accessed September 6, 2014.

[15] Amazon S3 Pricing.
`http://aws.amazon.com/s3/pricing/`
Accessed September 6, 2014.

[16] Microsoft Azure, Storage Pricing Details.
`http://azure.microsoft.com/en-us/pricing/details/storage/`
Accessed September 6, 2014.

[17] Pfleeger, C.P., Pfleeger, S.L., 2008: Sicurezza in informatica, Seconda edizione italiana, Pearson, pp. 355-362.

[18] U.S. Department of Commerce, NIST Special Pubblication 800-12, 1995: An Introduction to Computer Security: The NIST Handbook.
`http://csrc.nist.gov/publications/nistpubs/800-12/handbook.pdf`

[19] House Committee on Science, Space and Technology, Subcommittee on Investigations and Oversight, Bugs in the Program, 1989: Problems in Federal Government Computer Software Development and Regulation, 101st Congress, 1st session, August 3, 1989, p. 2.

[20] Letter from Charney, S., Chief, Computer Crime Unit, U.S. Department of Justice, to Guttman, B., NIST, July 29, 1993.

[21] Sprouse, M., 1992: Sabotage in the American Workplace: Anecdotes of Dissatisfaction, Mischief and Revenge, Pressure Drop Press, p. 7.

[22] Grimes, R.A., 2010: "Your guide to the seven types of malicious hackers", Infoworld.
`http://www.infoworld.com/d/security-central/your-guide-the-seven-types-malicious-hackers-636`

[23] Aycock, J., 2006: Computer Viruses and Malware, Springer, pp. 11-27.

[24] Sella, M., 2010: I nuovi illeciti. Danni patrimoniali e non patrimoniali, Utet Giuridica, p. 50.

[25] Edwards, D.: Definition of "Trojan Horse", The New Hacker's Dictionary.
http://www.outpost9.com/reference/jargon/jargon_35.html

[26] Kaspersky Lab, Internet Security Center, Minacce Internet, "Cos'è l'adware?"
http://www.kaspersky.com/it/internet-security-center/threats/adware
Accessed August 15, 2014.

[27] OWASP, Buffer Overflow.
https://www.owasp.org/index.php/Buffer_Overflow
Accessed August 15, 2014.

[28] US-CERT, United States Computer Emergency Readiness Team, Security Tip (ST04-015), Understanding Denial-of-Service Attacks, November 4, 2009, revised February 6, 2013.
https://www.us-cert.gov/ncas/tips/ST04-015
Accessed August 15, 2014.

[29] Ministero della Difesa, Riconoscere un attacco DoS.
http://www.difesa.it/SMD_/STAFF/REPARTI/II/CERT/TIPS_TRICKS/Pagine/DoS.aspx
Accessed August, 15, 2014.

[30] Microsoft Safety and Security Center, "Che cos'è l'ingegneria sociale?".
http://www.microsoft.com/it-it/security/resources/socialengineering-whatis.aspx
Accessed August 15, 2014.

[31] Kaspersky Lab, Kaspersky Anti-Virus.
http://www.kaspersky.com/anti-virus

[32] Bitdefender Antivirus.
http://www.bitdefender.com/solutions/free.html

[33] Avira Antivirus.
`http://www.avira.com/en/avira-free-antivirus`

[34] Microsoft, What is a firewall?.
`http://windows.microsoft.com/en-us/windows/what-is-firewall#1TC=windows-7`
Accessed August 15, 2014

[35] Castiglione, A., De Santis, A., Fiore, U., Palmieri, F., 2010: An Enhanced Firewall Scheme for Dynamic and Adaptive Containment of Emerging Security Threats, International Conference 2010 on Broadband, Wireless Computing, Communication and Applications (BWCCA), IEEE, pp. 475-481.

[36] Huang, K., Zhang, DF., 2010: An index-split Bloom filter for deep packet inspection, published on Science China, Information Sciences, Vol. 54, No. 1, p.23.

[37] TechTarget, SearchSoftwareQuality, cryptography.
`http://searchsoftwarequality.techtarget.com/definition/cryptography`
Accessed August 15, 2014.

[38] AES Crypt.
`http://www.aescrypt.com/`
Accessed September 2, 2014.

[39] Schneier, B., Whiting, D., 2000: A Performance Comparison of the Five AES Finalists.

[40] Sachnev, V., Kim, H.-J., 2011: Ternary Data Hiding Technique for JPEG Steganography, Digital Watermarking: 9^{th} Internation Workshop, IWDW 2010, Springer, pp. 202-210.

[41] CeMiSS, Centro Militare di Studi Strategici, Rapporto di Ricerca 2011: Cyber Security e Cyber Intelligence. La sicurezza dei Contingenti Militari contro le nuove minacce.
http://www.difesa.it/SMD_/CASD/IM/CeMiSS/Documents/Ricerche/2012/Stepi/cybersec_20111109_0846_Stefano_Mele.pdf
Accessed September 8, 2014.

[42] EmbeddedSW.net, Security Technology, Cryptography & Obfuscation, OpenPuff.
http://embeddedsw.net/OpenPuff_Steganography_Home.html
Accessed September 8, 2014.

[43] TechTarget, SearchSecurity, MFA (Multifactor authentication).
http://searchsecurity.techtarget.com/definition/multifactor-authentication-MFA
Accessed August 16, 2014.

[44] Federal Bureau of Investigation, Next Generation Identification System.
http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/ngi
Accessed September 17, 2014.

[45] Federal Bureau of Investigation, FBI announces full operational capability of the next generation identification system.
http://www.fbi.gov/news/pressrel/press-releases/fbi-announces-full-operational-capability-of-the-next-generation-identification-system
Accessed September 17, 2014.

[46] Messmer, E., 2009: "Cloud Security Alliance formed to promote best practices", Computerworld, March 31, 2009.
http://www.computerworld.com/s/article/9130884/Cloud_Security_Alliance_formed_to_promote_best_practices
Accessed March 14, 2013.

[47] Cloud Security Alliance, "The Notorious Nine Cloud Computing Top Threats in 2013", February 2013.
https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf
Accessed August 7, 2014.

[48] Saint Louis University, Information Security Incident Management Policy, InfoSec 1.13, February 16, 2012.
http://www.slu.edu/Documents/its/SLUInfoSecurity%201.13%20Information%20Security%20Incident%20Management%20v1.1_0216_2012.pdf
Accessed August 17, 2014.

[49] Hucaby, D., McQuerry, S., 2002: Cisco Field Manual: Catalyst Switch Configuration, Cisco Press.

[50] Stallings, W., Brown, L., 2012: Computer Security: Principles and Practice, Second Edition, Prentice Hall.

[51] Zhang, Y., Reiter, M.K., Juels, A., Ristenpart, T.: Cross-VM Side Channels and Their Use to Extract Private Keys, CCS' 12, October 16–18, 2012, Raleigh, North Carolina, USA.
http://www.cs.unc.edu/~yinqian/papers/crossvm.pdf

[52] CSA Cloud Controls Matrix.
https://downloads.cloudsecurityalliance.org/initiatives/ccm/CSA_CCM_v3.0.xlsx
Accessed August 15, 2014.

[53] Kitada, W., Hanaoka, G., Matsuura, K., Imai H., 2009: Unconditionally Secure Chaffing-and-Winnowing for Multiple Use, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Springer.

[54] Techopedia, Definition of Account Hijacking.
http://www.techopedia.com/definition/24632/account-hijacking
Accessed August 16, 2014.

[55] Cenzic, Application Vulnerability Report 2014, p. 4.
http://www.cenzic.com/downloads/Cenzic_Vulnerability_Report_2014.pdf
Accessed August 16, 2014.

[56] Shibboleth.
https://shibboleth.net/

[57] CodePlex, Project Hosting for Open Source Software, Microsoft Web Protection Library, AntiXSS.
http://wpl.codeplex.com/
Accessed September 13, 2014.

[58] Rosenfeld, M. a.k.a. Marlinspike, M., 2009: HTTPS stripping attacks, Black Hat DC 2009.
http://www.thoughtcrime.org/software/sslstrip/

[59] Microsoft Library, 2009: Man-in-the-Middle Attack.
http://technet.microsoft.com/en-us/library/dd572659
Accessed September 13, 2014.

[60] OWASP, 2014: Application Security Verification Standard 2014.
https://www.owasp.org/images/5/58/OWASP_ASVS_Version_2.pdf

[61] Microsoft Developer Network, Man in the Middle.
http://msdn.microsoft.com/en-us/library/cc247407.aspx
Accessed September 14, 2014.

[62] PuTTY User Manual, Using public keys for SSH authentication.
http://the.earth.li/ sgtatham/putty/0.60/htmldoc/

Chapter8.html
Accessed September 14, 2014.

[63] Robeuck, T.A., 2005: Network security: DoS vs DDoS attacks, Computer Crime Research Center.
http://www.crime-research.org/articles/
network-security-dos-ddos-attacks/
Accessed August 18, 2014.

[64] National Cyber Skills Centre, Glossary Terms, Zombie Computer.
http://www.cyberskillscentre.com/glossary/page/10/
Accessed August 18, 2014.

[65] Università degli Studi di Salerno, Dipartimento di Informatica, Corso sulla Security.
http://www.di.unisa.it/~ads/corso-security/www/CORSO-9900/
ddos/ddos.htm
Accessed July 26, 2014.

[66] CERT, Software Engineering Institute, Carnegie Mellon University, November 29, 2000: TCP SYN Flooding and IP Spoofing.
https://www.cert.org/historical/advisories/CA-1996-21.cfm
Accessed August 4, 2014.

[67] IBM, IBM Knowledge Center, ping Command.
http://www-01.ibm.com/support/knowledgecenter/ssw_aix
_53/com.ibm.aix.cmds/doc/aixcmds4/ping.htm?cp=ssw_aix_53%
2F1-2-0-15-22
Accessed September 9, 2014.

[68] Gu, G., Porras, P., Yegneswaran, V., Fong, M., Lee, W., 2007, BotHunter: Detecting Malware Infection Through IDS-driven Dialog Correlation, 16th USENIX Security Symposium.

[69] Whitaker, A., Newman, D.P., 2005: Penetration Testing and Network Defense, Cisco Press.

[70] François, J., Wang, S., State, R., Engel, T., 2011: BotTrack: Tracking Botnets Using NetFlow and PageRank, published in Networking 2011, Springer.

[71] Jiang, H., Shao, X., 2012: Detecting P2P botnets by discovering flow dependency in C&C traffic, Peer-to-Peer Networking and Application, December 2014, Volume 7, Issue 4, pp. 320-331.

[72] Jackson Higgins, K., 2008: Permanent Denial-of-Service Attack Sabotages Hardware.
`http://www.darkreading.com/permanent-denial-of-service-attack-sabotages-hardware/d/d-id/1129499?`
Accessed September 9, 2014.

[73] Chee, W.O., Brennan, T., 2010: HTTP POST, OWASP AppSec DC 2010.
`https://www.owasp.org/images/4/43/Layer_7_DDOS.pdf`
Accessed September 15, 2014.

[74] Syme, M., Goldie, P., 2003: Optimizing Network Performance with Content Switching: Server, Firewall and Cache Load Balancing, Prentice Hall, pp. 115-116.

[75] Tanase, M., March 11, 2003: "IP Spoofing: an Introduction", The Security Blog.
`http://66.14.166.45/sf_whitepapers/tcpip/IP%20Spoofing%20-%20An%20Introduction.pdf`
Accessed August 17, 2014.

[76] Fielding, et al.: Hypertext Transfer Protocol – HTTP/1.1, RFC 2616.
`http://www.w3.org/Protocols/rfc2616/rfc2616-sec1.html#sec1`

[77] Apache Docs, TimeOut Directive.
`http://httpd.apache.org/docs/2.2/mod/core.html#timeout`

[78] Lin, C.-H., Liu, J.-C., Chen, G.-H., Chen, Y.-H., Huang, C.-H., 2011: Double Check Priority Queue to Alleviate Malicious Packet Flows from Distributed DoS attacks.

[79] CERT, About Us.
`http://www.cert.org/about/`
Accessed August 10, 2014.

[80] CERT, Insider Threat.
`http://www.cert.org/insider-threat/`
Accessed August 10, 2014.

[81] Enciclopedia Treccani, Definition of "Due diligence".
`http://www.treccani.it/enciclopedia/due-diligence/`

[82] Australian Government, Department of Defence, 2012: Information security advice for all levels of government, Network Segmentation and Segregation.
`http://www.asd.gov.au/publications/csocprotect/Network_Segmentation_Segregation.pdf`

[83] Slay, J., Turnbull, B., 2004: "The Uses and Limitations of Unidirectional Network Bridges in a Secure Electronic Commerce Environment", INC 2004 Conference.

[84] Saltzer, J.H., Schroeder, M.D., 1975: The Protection of Information in Computer Systems.

[85] Solarwind, Patch Manager.
`http://www.solarwinds.com/patch-manager.aspx`

[86] Bace, R., Mell, P., 2001: NIST Special Publication on Intrusion Detection Systems, National Institute of Standards and Technology.

[87] Porras, P.A., Valdes, A., 1998: Live Traffic Analysis of TCP/IP Gateways, NDSS (Network and Distributed System Security) Symposium 1998.

[88] Blasco, J., Orfila, A., Ribagorda, A., 2010: Improving Network Intrusion Detection by Means of Domain-Aware Genetic Programming, ARES (International Conference on Availability, Reliability and Security) 2010, pp. 327-332.

[89] Hesham, A.I.M., Baiardi, F., Hariri, S., 2013: Ph.D. Thesis: Cloud Computing Security, An Intrusion Detection System for Cloud Computing Systems.

[90] Debar, H., Dacier, M., Wespi, A., 1999: "Towards a taxonomy of intrusion-detection systems", Computer Networks, Volume 31, Issue 8, April 23, 1999, pp. 805-822.

[91] OSSEC.
`http://www.ossec.net/`
Accessed September 24, 2014.

[92] Snort.
`https://www.snort.org/`
Accessed September 24, 2014.

[93] Prelude.
`https://www.prelude-ids.org/`
Accessed September 24, 2014.

[94] ACARM-ng, Alert Correlation, Assessment and Reaction Module – next generation.
`http://www.acarm.wcss.wroc.pl/`
Accessed September 24, 2014.

[95] Oracle, Data Mining Concepts 11g Release 1 (11.1).
`http://docs.oracle.com/cd/B28359_01/datamine.111/b28129/`

process.htm#DMCON002
Accessed September 24, 2014.

[96] Nadiammai, G.V., Hemalatha, M., 2013: Performance Analysis of Tree Based Classification Algorithms for Intrusion Detection Systems, Mining Intelligence and Knowledge Exploration, Springer, pp. 82-89.

[97] Polikar, R., 2006: "Ensemble based systems in decision making", IEEE Circuits and Systems Magazine Volume 6, Issue 3, pp. 21–45.

[98] Wu, X. et al., 2007: Top 10 algorithms in data mining, published on Knowledge and Information Systems, Volume 14, Issue 1, Springer, pp. 1-37.

[99] Wyld, D.C. Et al., 2011: Difference between Intrusion Detection System (IDS) and Intrusion Prevention System (IPS), Advances in Network Security and Applications, 4th International Conference, CNSA 2011, Springer, pp. 497-501.

[100] Roesch, M., Green, C., Sourcefire, Inc., Cisco, 2014: SNORT Users Manual 2.9.6.
http://manual.snort.org/

[101] OSSEC 2.8 Documentation, Manual.
https://ossec-docs.readthedocs.org/en/latest/manual

[102] Zheng, W., Fang, B., 2009: Structure-independent disaster recovery: Concept, architecture and implementations. Sci China Ser F-Inf Sci, 2009, Volume 52, Issue 5: 813–823, doi: 10.1007/s11432-009-0095-8.

[103] IBM PPRC.
http://www-01.ibm.com/support/knowledgecenter/SSPHQG_6.1.0/com.ibm.hacmp.pprc/ha_pprc_mirroriing.htm
Accessed September 24, 2014.

[104] Hitachi TrueCopy.
http://www.hds.com/products/storage-software/truecopy-remote-replication.html
Accessed September 24, 2014.

[105] FalconStor NSS.
http://falconstor.com/page/577/network-storage-server-maximize-storage-utilization-in-heterogeneous-structures
Accessed September 24, 2014.

[106] Oracle Data Guard.
http://www.oracle.com/technetwork/database/features/availability/dataguardoverview-083155.html
Accessed September 24, 2014.

[107] Rielaborazione di una slide del Prof. Dimakis, A.G. (University of Texas, Austin).

[108] Rajaraman, V., 2006: Toward a Computing Utility, Annals of the Indian National Academy of Engineering, December 2006, Volume III, pp. 1–10.