Alma Mater Studiorum - Università di Bologna

CAMPUS DI CESENA
SCUOLA DI SCIENZE

Corso di Laurea in Ingegneria e Scienze Informatiche

Applicazioni client-server sicure per utenti in mobilità

Relazione finale in
Reti di Telecomunicazione

Relatore:
Chia.mo
Franco Callegati

Presentata da:
Samuele Medici

Correlatore:
Dott.
Alessandro Bulgarelli

Sessione IV
Anno Accademico 2017-2018

Indice

1 Introduzione

Lo scopo di questa tesi è quello di analizzare approfonditamente un'architettura per garantire una comunicazione sicura tra client e server.

In particolare questa analisi si sofferma maggiormente su tutti gli aspetti per la comunicazione di un server che espone servizi all'esterno per client che non sono all'interno della intranet aziendale.

Con mobilità definiamo la possibilità dell'utente finale di utilizzare dispositivi diversi, quali smartphone, tablet, pc e desktop e ottenere un'esperienza consistente e simile tra i diversi dispositivi.
Nel corso della dissertazione si faranno distinzioni tra i dispositivi sopra elencati, pur tenendo del fatto che rispetto all'architettura e al flusso la distinzione tra essi risulta di poca importanza.

Definiamo come architettura sicura, un flusso client-server e server-client che garantisce una comunicazione che gode di tre principali proprietà: riservatezza, integrità e disponibilità.

BPER Banca ha mostrato particolare interesse a questa tematica per applicazioni future, concentrandosi su tutti gli aspetti che garantiscono, in modo quasi completo, ad un elevato livello di sicurezza.

Il mio contributo è stato quello di analizzare dettagliatamente una rete F5, una soluzione fornita da VMWare che attualmente BPER Banca sta esaminando per possibili sviluppi futuri, svolgere alcuni test da remoto per verificare la disponibilità di tale servizio e trovare le possibili vulnerabilità di tale sistema.

Il primo capitolo si concentra sullo stato dell'arte della sicurezza delle comunicazioni.

Lo studio e la materia è molto vasta, e gli argomenti sono stati limitati a quelli valutati più attinenti all'ambito di questo studio.
Viene fatta quindi un'analisi dei possibili dispositivi maggiormente utilizzati, dei possibili fattori di rischio collegati a tale impiego e dei relativi meccanismi di sicurezza. Nel capitolo vengono esaminati gli attacchi che possono minare l'affidabilità del flusso di comunicazione, concludendo con una descrizione dei principali meccanismi di difesa per contrastare.

Il secondo capitolo è costituito da un'analisi delle tecnologie che in questa analisi sono state reputate importanti per il conseguimento di un sistema sicuro di comunicazione. Vengono analizzate le reti MPLS, che costituisce la intranet per BPER Banca, tecnologia VPN e Reverse Proxy.

Il terzo capitolo ha lo scopo di illustrare ogni singolo componente costituito dall'architettura analizzata e valutare gli aspetti positivi e negativi di tale implementazione.

Nell'ultimo capitolo viene fatta un'analisi finale su tutti gli aspetti presi in esame e considerazioni legate a sviluppi futuri.

2 Stato dell'arte

In questo capitolo capitolo vengono approfondite le varie entità che fanno parte del flusso di comunicazione client e server.
In particolar modo verranno analizzati i dispositivi utilizzati per avviare la connessione da remoto, relativi fattori di rischio e come possiamo definire un dispositivo sicuro.
Vengono poi evidenziati le minacce che possono verificarsi durante il flusso, in particolare modo verranno accentuate le minacce più frequenti.
Per ultimo vengono analizzati i meccanismi di difesa messi in atto per contrastare tali minacce, e le pratiche più comuni per ottenere un canale di comunicazione stabile e sicuro.

2.1 Dispositivi analizzati

In questa sezione sono illustrati i dispositivi per l'accesso remoto e in particolare sono evidenziate le principali caratteristiche che rendono tale dispositivo sicuro.

Per facilità di sviluppo della dissertazione, in questa sezione verranno considerati solo i dispositivi più comuni per una connessione client server, per cui viene fatta una distinzione netta tra device Desktop, che comprendono tutti i Personal Computer, portatili, laptop e computer fissi, e dispositivi Android e iOS.

Per i Desktop sono stati considerati i due sistemi più diffusi, quali OSx e Windows.

2.1.1 Fattori di rischio

Questa sezione ha lo scopo di mostrare i principali rischi, cause e scenari di attacco legate all'utilizzo di qualsiasi dispositivo fisico a cui non sono state apportate modifiche in sicurezza.

2.1.1.1 Memorizzazione di dati

Un primo fattore di rischio specialmente legato ai dispositivi mobili è la memorizzazione di dati in modo non sicuro o perdita di dati sensibili.
Tali rischi possono risultare in:

- ottenimento di un dispositivo perso o rubato e accesso ai relativi dati
- accesso ai dati da parte di malware o altre applicazioni installate sul dispositivo
- furto di identità
- perdita di dati

Questo rischio comprende i maggiori database utilizzati (SQL, SQLite per i dispositivi mobili) file di log, XML file (inclusi anche i file di manifest), memorie SD esterne, e collegamenti a storage cloud.
Un esempio semplice è la memorizzazione di credenziali
La maggior parte della vulnerabilità sono originate da parte del sistema operativo in utilizzo, i framework utilizzati per lo sviluppo, compilatori, hardware e se il dispositivo permette di ottenere diritti sui file installati.

2.1.1.2 Canale di comunicazione

Sempre più frequenti sono l'utilizzo di applicazioni su device mobili che richiedono un canale di comunicazione sicuro.
Quando l'applicativo trasmette i dati viaggiano attraverso la rete, e agenti esterni possono sfruttare vulnerabilità sul canale di comunicazione per intercettare dati sensibili.

Tale rischio comprende tutti gli aspetti che comprendono trasportare i dati da un punto A ad un punto B. Comprende comunicazioni mobile-to-mobile, client-server e quindi anche tutti i protocolli utilizzati per le comunicazioni: TCP/IP, WiFi, Bluetooth, NFC, GSM, 3/4G, SMS.

Tutti i rischi collegati ad un canale di comunicazione non sicuro sono: integrità dei dati, confidenzialità dei dati e integrità dell'origine della comunicazione. Se i dati possono essere alterati durante il transito, senza possibilità di verificare un cambiamento, la confidenzialità del canale è messa a repentaglio.

Inoltre una serie di problematiche possono verificarsi anche dalla erronea configurazione di strumenti atti a rendere sicuro il canale di comunicazione come SSL / TLS: utilizzo di certificati provvisti da un ente non di fiducia.

2.1.1.3 Qualità del codice dell'applicativo

Questa sezione mostra alcuni rischi legati al codice, e per facilità verranno considerate allo stesso modo applicazioni mobili e web.

L'utilizzo di applicazioni o di servizi web costringe i fruitori a dover controllare in modo sempre più ardito come questi applicativi vengono costruiti fin dal primo momento.
Un codice di scarsa qualità o il mancato utilizzo di pattern e linee guida per lo sviluppo sicuro corrisponde spesso a vulnerabilità che possono arrecare seri danni all'azienda.
L'impatto che può avere sull'azienda può essere molto vario in quanto dipende dalla natura della vulnerabilità.
Spesso può portare a:

- furto di informazioni
- danno all'azienda
- furto di proprietà intellettuale

Altri aspetto tecnici sono legati soprattutto al degrado dell'applicazione in termini di performance e utilizzo della memoria.

Questa sezione riguarda tutte le vulnerabilità a tutti i livelli di codice, dal client al server.
Comprende quindi vulnerabilità più semplici come *Buffer Overflow,* formattazione di stringhe, input che permettono injection di codice (DOM-based XSS in una *WebView*).

In fase di design e di sviluppo dell'applicazione è di vitale importanza seguire linee guida per lo sviluppo, scrivere codice leggibili e documentato, controllare che tutte le librerie che vengono adoperate siano ben documentate e non vi siano bug di sicurezza.

E’ inoltre opportuno utilizzare alcune strategie per uno sviluppo efficiente come l’utilizzo di tecniche di *Continuous integration* e *Continuous delivery* che garantiscono un efficace sistema di sviluppo che garantisce una diminuzione consistente di bachi presenti nell’applicativo.

2.1.1.4 Reverse Engineering

Un ulteriore rischio è legato al Reverse Engineering.
Questo tipo di attacco avviene quando un agente riesce ad accedere e analizzare a parti del codice per poterlo utilizzare contro l’organizzazione che lo utilizza normalmente. La maggior parte delle applicazioni sono suscettibili a questo fattore dovuto alla natura del codice. La maggior parte dei linguaggi utilizzati oggi sono ricchi di metadati che aiutano in fase di sviluppo e allo stesso tempo aiuta un agente terzo a capire come l’applicazione funziona.
Un’applicazione è soggetta a Reverse Engineering se è possibile fare un’analisi funzionale, ottenere in parte o nella sua interezza il codice sorgente da file binari oppure capire il contenuto di tabelle binarie.
L’impatto sull’organizzazione comprende:

- furto di proprietà intellettuale
- danno all’organizzazione
- furto di identità
- compromissione di sistemi server

Per prevenire questo tipologia di attacco è possibile utilizzare due principali strumenti: strumenti di offuscazione o cercare di capire se il dispositivo sul quale gira l’applicazione è in grado di compiere un simile attacco.

Lo strumento di offuscamento ha lo scopo di nascondere codice, metadati, metodi e punti specifici dell’applicazione tramite tecniche di crittografia.

2.1.2 Proprietà per un dispositivo sicuro

In questa sezione vengono mostrate le più importanti proprietà che rendono il dispositivo sicuro.

Un dispositivo sicuro deve disporre di componenti hardware che contrastino attacchi sia applicativi che fisici. Quando utilizzato per proteggere dati e correttezza del dispositivo, l'hardware fornisce un solido punto di partenza di fiducia dove un software ricco di funzionalità può essere utilizzato in sicurezza.

La prima proprietà da verificare è se il dispositivo possiede una base hardware su cui porre fiducia, ossia se l'hardware predispone la possibilità di fornire un'identità unica che è inseparabile dall'hardware stesso.
Un secondo punto da considerare è la presenza di un hardware protetto che include tutti le chiavi private dell'utente, su cui non vengono eseguiti nessun processo computazionale.
Un successivo punto per un dispositivo sicuro è l'autenticazione che avviene tramite presenza di certificati installati sul dispositivo. L'autenticazione tramite credenziali spesso risulta debole e quindi la presenza di certificati porta ad avere benefici riguardo l'autenticazione.
Una forte compartimentazione dei vari hardware all'interno del dispositivo e relative barriere, impediscono che il fallimento di un componente possa sfociare in una reazione a catena su vari hardware.

I punti sopra menzionati costituiscono il punto di partenza per un'analisi dei dispositivi. Successivamente verranno elencati le tipologie di dispositivo prese in analisi e per ognuna verranno prese in considerazioni questi punti.

2.1.3 Tipologie di dispositivo

In questa sezione saranno evidenziate le principali differenze tra applicazioni web e app sviluppate in codice nativo e saranno inoltre mostrate le diversità tra dispositivo mobili, quindi tablet e smartphone, e pc, ossia laptop e desktop.

Verrà fatta una distinzione aggiuntiva tra i vari dispositivi mobili, in quanto al momento sul mercato non condividono uno stesso approccio alla sicurezza.

2.1.3.1 Desktop

In questa sezione vengono analizzati i dispositivi computer laptop e desktop, i principali sistemi operativi e le applicazioni.

Il mondo dei personal computer può essere banalmente suddiviso nei due principali sistemi operativi più utilizzati: Windows e Mac OSX.
Per fare un'analisi approfondita e significativa dei due sistemi occorre considerare le due versioni più recenti e più aggiornate.

Principalmente questi due sistemi utilizzano applicazioni scritte in codice compilabile dai sistemi operativi, come C, C#, Objective-C o compilato tramite macchine virtuali come Java, Scala, .NET, oppure tramite Web Browser navigano su Internet. La differenza sostanziale tra le due opzioni è che tutto il codice web utilizzato è gestito all'interno del Browser, e che password, dati personali, cache, comunicazioni li gestisce quasi completamente. Nel caso delle applicazioni invece, ogni gestione è dipendente a seconda del codice.

Lo sviluppo per un'applicazione web è nettamente più rapido, più facilmente testabile e ha una serie di gestioni per una usabilità più alta, come *Single Sign-On* (SSO) che permette l'immissione dei dati per il login una volta sola.
D'altro canto dal punto di vista della sicurezza è molto più complesso.
La maggior parte degli attacchi avviene tramite siti web: manomissioni, immissione di codice malevolo, utilizzo di librerie di terze parti non firmate.

La facilità con cui è possibili penetrare un sito web è allarmante e spesso chi vuole ottenere un prodotto sicuro opta per una soluzione che non viene eseguita su browser ma sulla macchina stessa.

Lo sviluppo di applicazione in linguaggio nativo, come C/Objective-C o attraverso virtual machine come Java, Scala, C#, invece risulta essere più complesso.
Permette però di avere una maggior padronanza di quello che sarà il prodotto ultimo e vi sono una serie di tecniche di offuscamento o di prevenzione della manomissione del codice che permettono all'applicativo di garantire un sistema sicuro. E' necessario evidenziare che però le applicazioni vengono installate e eseguite su sistemi di cui non si ha nessun tipo di controllo e gli utenti possono procedere a mettere a dura prova tali applicazioni, isolando o provando a utilizzare codici malevoli per danneggiare l'applicazione.

Riassumendo, i sistemi pc sono esposti a moltissime minacce e rischi, dato che, permettono di poter installare software da web non firmato o certificato, non potendo gestire il contenuto di esso.

Per continuare questa analisi è necessario definire che cos'è una vulnerabilità che può essere rilevata in tali dispositivi.
Consideriamo come vulnerabilità di un sistema una minaccia legata a componenti di sistemi che possono essere compromessi, applicazioni con bachi che possono essere sfruttati e in generale difetti che possono danneggiare il sistema parzialmente o nella sua interezza.

Tendenzialmente si crede che i sistemi Mac siano esenti da attacchi o bachi nel sistema. Questa affermazione non può essere più falsa.

Come riportato da *CVEDetails* [1], database più grande che raccoglie tutte le vulnerabilità dei sistemi, si sono rilevati dei numeri molto alti di vulnerabilità nei due sistemi.

[1] CVEDetails, https://www.cvedetails.com

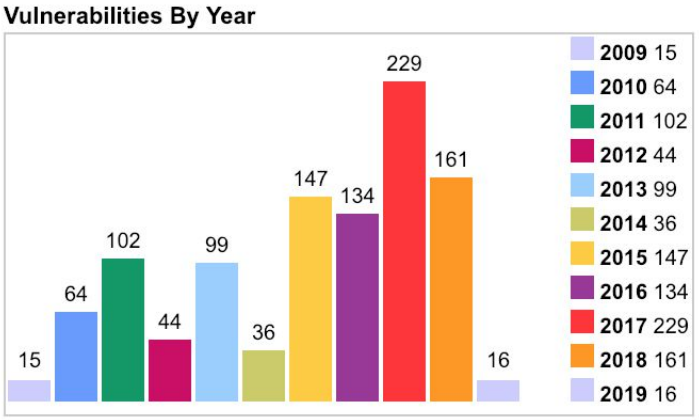

Figura 1.1: Numero di vulnerabilità individuate per anno per sistemi Windows

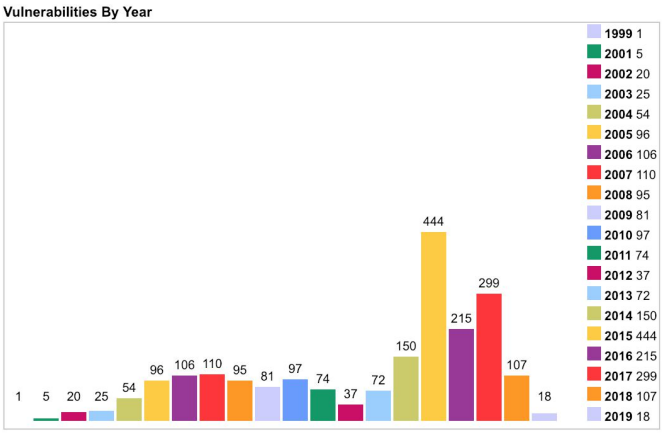

Figura 1.2: Numero di vulnerabilità individuate per anno per sistemi Mac OSx.

Negli anni sono stati sviluppati una serie di applicazioni anti malware, sviluppati da grandi aziende come Kaspersky[2] che aiutano a contrastare i più comuni malware conosciuti.
Tali applicazioni contengono un database di grandi dimensioni che contiene codice che può essere etichettato come malevolo, eseguono scan continui per garantire un certo livello di sicurezza.

E' stato più volte delineato che nessuno dei sistemi preso in considerazione in questa sezione è immune a vulnerabilità. Allo stato attuale non esiste un sistema più sicuro di un altro, esistono una serie di norme che applicate può garantire uno stato sicuro.

[2] Kaspersky Lab, https://www.kaspersky.it/

Di seguito, invece, saranno illustrati altri dispositivi, ritenuti più sicuri per una serie di meccanismi che saranno illustrati.

2.1.3.2 Dispositivi Android

In questa sezione vengono considerati tutti i dispositivi che hanno come sistema operativo Android, sviluppato da Google.

Android è sviluppato da Google ed è il sistema mobile più diffuso attualmente. Appartiene al mondo open source e questo spesso dispone dei grossi problemi per quanto riguarda la sicurezza dei dispositivi che lo impiegano come sistema operativo. Fino alla versione 7.0 di Android, chiamata Nougat, i livelli di sicurezza erano legati solamente a livello di applicazione. Per cui se l'utente non di dispone di certi strumenti per garantire la sicurezza o se non utilizza certi tipi di applicazioni Android non garantisce un sicuro uso del telefono.
Dalla versione Nougat i telefoni con questo sistema dispongono anche di un livello aggiuntivo tra il sistema operativo e il livello di applicativo chiamato Trusty TEE (Trusted Execution Environment [3]). Fondamentalmente Trusty TEE è un componente isolata sia da altri componenti hardware sia da altre applicazioni attive, proteggendo così le applicazioni che girano su TEE da eventuali codici malware o altre potenziali vulnerabilità di Android.

Dalle versioni 5.0, i dispositivi Android dispongono inoltre di un sistema di *full-disk encryption*, procedimento per il quale tutti i dati dell'utente vengono codificati utilizzato AES a 128bit.
Nelle versioni più recenti invece è stato introdotto anche *file-based encryption,* che da la possibilità di criptare alcuni file utilizzando chiavi diverse e dando la possibilità di decriptare ciascun file indipendentemente.
Questi sistemi di criptazione non sono però sufficienti per garantire un livello di sicurezza adatto per comunicazioni per un'organizzazione come una banca.

[3] https://source.android.com/security/trusty/

E’ da evidenziare il fatto che questi dispositivi danno la possibilità agli utenti di ottenere elevati privilegi sul sistema stesso e poterlo sfruttare attraverso un procedimento chiamato *rooting*.

Rooting è il processo per cui un utente che dispone di un device Android riesce a ottenere controlli privilegiati (come root di sistema) sui sottosistemi Android.
Eseguire un’app in un device *rooted* aumenta i rischi enormemente.
Il primo fattore da considerare è la capacità di un utente di poter utilizzare un malware inserito appositamente per danneggiare le altre applicazioni.
Il secondo fattore invece da non tralasciare è la possibilità di poter ricavare il codice di altre applicazioni tramite tecniche di Reverse Engineering.

Nei casi in cui è ritenuto necessario, è buona prassi effettuare il controllo all’avvio dell’app per intraprendere un’azione appropriata, ad esempio informare l’utente dei rischi o impedire determinate operazioni.
E’ possibile impostare delle regole di esclusione nella console di Google Play per impedire il download dell’app a device che non rispettano i requisiti base di integrità o che sono privi della certificazione da parte di Google.

Un ulteriore punto su cui è necessario soffermarsi è la possibilità di inserire codice senza venir approvato, è necessario solamente spuntare una casella di sistema che permette il Debug USB e da lì è possibile inserire applicazioni che passino dal controllo del Google Store.

2.1.3.3 Dispositivi iOS

In questa sottosezione viene illustrata l’architettura di un tipico dispositivo iOS, sviluppato da Apple. Per facilità di sviluppo della tesi, vengono considerati solamente i componenti valutati rilevanti.

I dispositivi mobili con sistema operativo iOS sono considerati i sistemi più sicuri al momento.

La prima analisi è sicuramente sull'hardware di questi device che rispetto ad altri dispositivi non dispongono di componenti che garantiscono un livello di sicurezza maggiore.

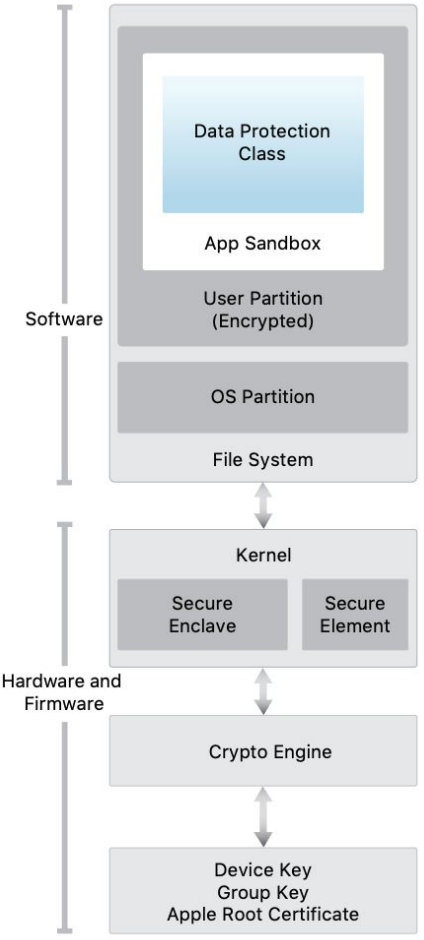

Figura 1.3: Architettura di un sistema iOS

I sistemi iOS dispongono di una serie molto complessa di componenti che collaborano per ottenere un grado di sicurezza sul sistema eccezionale.
Partendo dal *Boot ROM*, codice immutabile che lancia il sistema operativo è il punto di partenza, chiamato *root of trust,* costruito al momento della fabbricazione ed è implicitamente sicuro in quanto contiene la chiave per garantire uno startup sicuro del device.
In caso di fallimento del componente è possibile **unicamente** ripristinare i dati di fabbrica tramite connessione ad iTunes. In questo modo se il dispositivo viene

sottoposto ad un alterazione hardware il sistema dispone di questo meccanismo per garantire che i dati non possano essere sottratti ad agenti esterni.
Questi sistemi dispongono anche di un coprocessore chiamato *Secure Enclave* che ha il compito di provvedere tutte le operazioni di crittografia per protezioni di dati, gestione delle chiavi e mantiene l'integrità dei dati anche se il kernel viene compromesso.

Le operazioni crittografiche sono molto costose in termini di potenza computazionali e di vita della batteria, per questo motivo Apple ha introdotto un meccanismo dedicato su base AES a 256 bit che garantisce una crittografia ad alto livello di sicurezza con ottime prestazioni.
Le chiavi univoche del dispositivo, *unique ID* (UID) e *group ID* (GID) sono chiavi AES a 256 bit che sono integrate direttamente (o compilate nel caso delle GID) nel Secure Enclave nel momento della fabbricazione del prodotto, in modo tale che nessun software o hardware possano leggere tali chiavi direttamente.

In aggiunta ai sistemi hardware, iOS fornisce una serie di sistemi di più alto livello, tra cui il layer di *Data Protection*, presente in iOS 7 o superiore, per dare un'ulteriore protezione ai dati memorizzati in memoria flash. Implementando una gerarchia di chiavi, questo livello assegna ad ogni nuovo file una chiave, sempre basata su AES 256 bit, e passando la chiave e tale file al livello inferiore che si occupa della crittografia del file e procedendo poi con la memorizzazione.

Il sistema *Keychain* è il meccanismo sicuro con il quale Apple gestisce le password dell'utente e altri dati sensibili come token di login e chiavi. Implementato utilizzando un SQLite database nel file system del dispositivo, la tabella è costituita da due tipi di chiave *table key* e *secret key.* I metadati vengono crittografati con la table key per aumentare la velocità di ricerca mentre i dati da nascondere vengono criptati tramite secret key. La table key è sempre memorizzata nel Secure Enclave, anche se spesso viene messa in cache per velocizzare il procedimento.
I vari oggetti che appartengono alla Keychain sono elencati in un *Access Control List* (**ACL**).
In questo elenco viene specifico come ogni singolo oggetto possa essere prelevato, se tramite PIN o tramite impronta digitale.

Al livello applicativo, iOS fornisce un ulteriore catena di sicurezza che garantisce un perimetro ben delineato sui rischi legati alle applicazioni.
E' necessario evidenziare che tutte le applicazioni che vengono eseguite sul dispositivo non possono essere lanciate se chi le produce non possiede un certificato specifico rilasciato dall'azienda Apple. In questo modo solo le aziende o programmatori che possiedono questo certificato possono sviluppare applicativi che saranno poi messi in commercio.
Oltretutto, ogni applicazione che viene rilasciata deve essere firmata digitalmente, ad esempio le applicazioni preinstallate come Mail e Safari, sono firmate da Apple.
I sistemi iOS non permettono quindi di poter scaricare nessun tipo di codice o applicazioni da siti.
Dopo che l'applicazione è stata certificata e resa disponibile sullo store ufficiale, il sistema da un'altra serie di misure di sicurezza prevenendo le applicazioni a compromettere il resto del sistema a runtime.
Isolando l'applicativo viene tolta la possibilità di manomettere dati e altri applicazioni che sono attive. Dando ad ogni applicazione il grado minimo di privilegio limitando le azioni possibili. Tutto quanto il sistema operativo in realtà è montato in modalità *read only.*

L'ultimo punto è legato principalmente a come i device vengono sbloccati. La maggior parte dei dispositivi Apple utilizza il *Touch ID,* sistema biometrico basato sull'impronta digitale. Una volta fornita l'impronta dell'utente i vari sistemi vengono impiegati per decriptare i file. Le ultime versioni impiegano anche la tecnologia *Face ID,* che usa come strumento biometrico il sistema di riconoscimento facciale. Vi sono alcune controversie su questi sistemi, ma Apple garantisce, tramite tecnologia *True Depth*, di soddisfare gli standard internazionali di sicurezza con questo sistema. In caso di manomissione di sistema o di fotocamera, tale tecnologia viene interamente disabilitata.

Come anche per i dispositivi Android, per i device Apple è possibile ottenere una serie di permessi aggiuntivi tramite il procedimento di Jailbreak. A differenza di Android, questo meccanismo permette unicamente l'installazione e l'esecuzione di codice e applicazioni non firmate, destabilizzando l'architettura che iOS ha disegnato accuratamente.

Viene sottolineato che ci sono comunque dei meccanismi non ancora perfettamente delineati su come poter verificare che l'applicazione che viene installata possa capire se il dispositivo sia corrotto, in ogni caso i molteplici meccanismi di sicurezza non garantiscono una sicurezza totale in questo caso e togliere la possibilità di Jailbreak è la miglior soluzione in questo caso.

Come già anticipato, sopra elencati vi sono i componenti che sono stati valutati più importanti per l'analisi dell'architettura finale. iOS dispone di moltissimi altri elementi che arricchiscono questo prodotto rendendolo così uno dei dispositivi più sicuri ed efficaci del momento.

2.1.4 Considerazioni

E' d'obbligo evidenziare che le considerazioni di cui sopra non sono del tutto complete. Sono state fatte queste analisi architetturali di sistema che non comprendono altre componenti che possono essere fondamentali per un sistema di comunicazione da remoto per organizzazioni.

Questi componenti possono essere sistemi di rilevamento di intrusione, *Intrusion Detection System,* sistemi di prevenzione, *Intrusion Prevention System,* scanner di malware, antivirus.
Vi sono molte aziende, organizzazioni che, tra le policy di lavoro, forniscono ai dipendenti dispositivi con applicazioni di questo genere preinstallate.

Bisogna considerare che al momento iOS è il sistema più sicuro al momento. I vari meccanismi di isolamento, l'impossibilità di installare senza essere certificati direttamente da Apple ha permesso di costruire prodotti (non poco costosi) che impediscono il 98% delle minacce che oggi sono più frequenti in assoluto.
Secondo il *Threat Intelligence Report* gli iPhone sono soggetti solo allo 0,85% dei malware in circolazione. Una percentuale decisamente bassa considerando che per Android la percentuale è di 47,15%. [4]

[4] https://networks.nokia.com/solutions/threat-intelligence/infographic

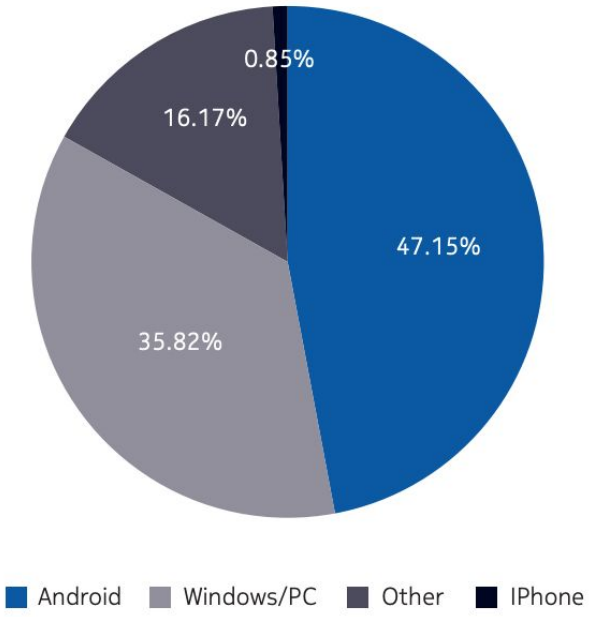

Figura 1.4: Percentuale di dispositivi soggetti a malware nel 2018 fornita ***Nokia Threat Intelligence Report***

2.2 Principali Minacce

Questa sezione elenca le principali minacce a cui un sistema viene sottoposto. Vengono considerate esclusivamente quelle ritenute fondamentali da affrontare per la stesura di questa dissertazione e l'analisi del sistema preso in considerazione.

Definiamo come minaccia il rischio a cui il sistema è sottoposto nel caso in cui un agente esterno, non avente diritto, sfrutta una falla del sistema per poter ottenere, danneggiare, alterare dati del sistema stesso.
Esiste una lista molto numerosa di questo tipo di rischi associati al mondo delle imprese.
Sono state qui riportate quelle che possono arrecare un danno maggiore a questo tipo di sistemi.

2.2.1 Accesso non autorizzato

L'accesso non autorizzata rappresenta l'atto diretto o indiretto per il quale una persona esterna al sistema ottiene permessi, informazioni confidenziali all'interno del sistema senza autorizzazione o averla ottenuto in modi fraudolenti.

Questa tipologia di attacco può essere messa in atto su diversi stati del sistema, dalla conoscenza di dati di un utente regolare, alla sottrazione di un dispositivo, furto di identità o attacchi fisici al sistema.

Viene fatta una distinzione di tipi di attacco per ottenere l'accesso senza autorizzazione in base alla tipologia di minaccia:

- **Social Engineering**: definiamo come Ingegneria sociale tutti quei attacchi che vengono portati in atto tramite inganno su una persona che fa parte del sistema. Un esempio concreto di questo tipo di vettore è *Phishing,* ossia truffa che viene spesso tramite mail. Un membro di una banca riceve una mail non autentica dove vi si richiede di inserire certe credenziali per l'accesso alla banca stessa. Il dipendente inserisce i dati inconsapevole che tali credenziali ora sono in possesso di una persona esterna che li utilizzerà per scopi malevoli.
- **Furto o perdita di dati:** se le credenziali vengono memorizzate in un dispositivo qualsiasi e tale dispositivo viene sottratto o perso, l'utente non ha più accesso a quei dati e vi è la possibilità che chi si è appropriato di tale dispositivo sia in possesso anche di tali dati e possa aver rubato l'identità dell'utente per poter accedere ai suoi dati sensibili, conti correnti, etc.

- **Attacchi brute force:** si indica generalmente un metodo utilizzato da un attaccante per individuare una password di accesso al sistema provando in maniera esaustiva tutte le possibili combinazioni di caratteri, lunghezza di stringhe ammesse dal sistema preso in esame. E' spesso poco efficiente in quanto richiede un numero elevato di tentativi per trovare la soluzione. E' però consigliato prendere in considerazione questa tipologia di attacco in quanto può danneggiare il sistema stesso.

2.2.2 Attacchi Man in The Middle

Definiamo come attacchi *Man-in-the-Middle* (**MITM**) una tipologia di rischio dove un attore si inserisca in una conversazione tra due parti.

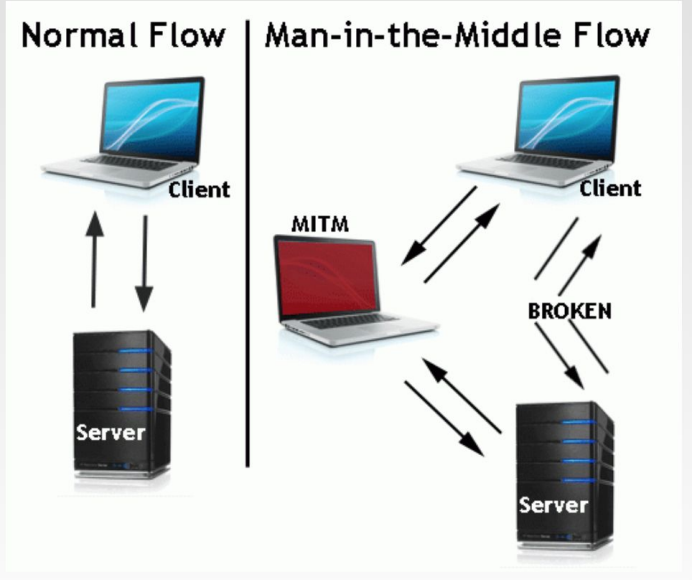

Figura 1.5: Flusso attacco Man in The Middle

In figura viene rappresentato un tipico flusso di comunicazione client - server.
A lato è raffigurato un esempio di flusso di attacco MITM, dove un agente esterno si posiziona nel canale di comunicazione tra il client e il server e interrompe il flusso normale di traffico.

Tra i principali attacchi MITM ricordiamo quelli più comuni, elencati in ordine crescente di danni che possono apportare al sistema:

- **Sniffing:** è un tipo di attacco passivo, in quanto l'attaccante è solamente in ascolto della comunicazione e non apportano direttamente nessuno tipo di danno al sistema.
- **Packet injection:** in questa tipologia di attacco, l'agente esterno interferisce con la comunicazione inserendo dei pacchetti in modo che appaiano come se facessero parte della comunicazione normale. Questo attacco porta a un abbassamento dell'integrità dei dati e dell'origine del traffico del client.
- **Session Hijacking:** l'attaccante in questo caso si appropria di un cookie HTTP che viene sottratto alla vittima per poi essere utilizzato nella ricreazione di una nuova sessione. In questo caso il server è ignaro che l'agente non sia il client normale e quindi stabilisce la sessione con la macchina dell'attaccante.
- **SSL stripping:** in questo attacco, l'agente intercetta la richiesta HTTP per la redirect su HTTPs. In questo modo l'attaccante continua a stabilire la connessione HTTPs con il server e una connessione HTTP con la vittima, agendo come ponte tra le due parti. Dopo aver stabilito questo flusso i messaggi del client arrivano in testo chiaro e quindi facilmente leggibili, l'attacco danneggia l'integrità e la confidenzialità di informazioni sensibili.
- **DNS spoofing:** questa tipologia di attacco comprende tutte le minacce che sfruttano vulnerabilità sul sistema di DNS che associa il nome di dominio a un indirizzo IP. L'attaccante inietta una entry in un server DNS che risulterà nella cache e tutti gli utenti utilizzeranno tale entry finché non scade. In questo modo si può redirezionare le vittime su un sito a piacere, utilizzando poi tecniche di phishing si può procedere al furto di informazioni e dati sensibili.

Questa tipologia di attacco è forse tra le più pericolose in quanto mette a repentaglio l'identità di utenti che vogliono raggiungere un server da remoto.
Nei capitoli successivi verranno discusse tutte le tecniche per mettere al sicuro l'identità dei clienti e dei server che si cerca di raggiungere.
Vi sono inoltre una serie di tool che alcune distribuzioni Linux mettono a disposizione per poter testare anche l'integrità dei sistemi che siano a prova di questa tipologia di attacchi.

2.2.3 Attacchi Denial of Service

Denials of Service è un tipo di attacco nel quale l'attaccante cercare un nodo della rete o una macchine non disponibile o irraggiungibile da tutti gli utenti temporaneamente o indeterminatamente interrompendo i servizi.

In questa tipologia di attacco non abbiamo furto o perdite di dati però possono avere dei costi alti per le vittime in termini di tempo e denaro per rimettere in piedi il sistema.

Vi sono due tipologie di attacchi DoS: *flooding* o *crashing*.
Nel primo tipo, i flooding DoS "inondano" il carico dei nodi o delle macchine con l'intento di rallentare o addirittura fermare i servizi.
In questa categoria ricordiamo gli attacchi più comuni:

- **Buffer overflow:** in generale il buffer overflow consiste in un buffer che viene riempito con dati che non può contenere, facendo scrivere dati nella zona successiva in memoria. In questo vettore di attacco il flooding DoS sovraccarica i buffer dei server per mandare il sistema in segmentation fault e farlo crashare.
- **ICMP flood:** l'attaccante in questo caso tenta di interrompere i servizi di un server mandando un numero molto alto di richieste sul protocollo **ICMP.** Questo attacco è anche conosciuto come *Ping flood,* in quanto il modo più tradizionale è mandare una serie di ping al router finale. Evidenziamo il fatto che spesso questi attacchi vengono fatti con protocollo *UDP*.
- **SYN flood:** in questo caso l'attacco DoS avviene tramite una serie richieste *SYN,* utilizzate solitamente per stabilire una connessione TCP, in questo modo il server in risposta non riesce mai a stabilire una connessione TCP poichè l'attaccante non completa mai l'handshake del protocollo, e ne apre continuamente di nuove interrompendo la possibilità al server di stabilire connessioni con clienti regolari.

Nel secondo tipo, gli attacchi sfruttano bachi o vulnerabilità del sistema con il fine di portare il sistema offline in modo che non sia più accessibile e utilizzato.

Questo tipo di vulnerabilità sono dipendenti dal sistema di utilizzato, le tipologie di tecnologie impiegate, software utilizzati, codice sviluppato e librerie esterne utilizzate.

In aggiunta, ricordiamo i *Distributed Denial of Service* (DDoS). Questo attacco consiste in una serie di macchine che vengono orchestrate per eseguire un singolo attacco DoS. La differenza sostanziale è che negli attacchi DoS tradizionali, questi vengono eseguiti da più località, spesso diverse.

2.3 Sistemi di protezione

In questa sezione verrà illustrata i principali metodi che vengono utilizzati contro le minacce verso il nostro sistema preso in considerazione.

2.3.1 Autenticazione

Questa sottosezione mostra la definizione di autenticazione e i tipici meccanismi di autenticazione messi in atto.

Definiamo con autenticazione il processo di riconoscimento dell'identità di una persona o di un utente. E il procedimento che associa una richiesta a una serie di credenziali relative ad una persona.
Ogni sistema sicuro dovrebbe iniziare ogni singola operazione con l'autenticazione prima di lasciare che l'utente possa procedere con altre operazioni non autorizzate, evitando così accessi non autorizzati e usi autorizzazioni non lecite.

Possiamo suddividere il processo di autenticazione in due fasi distinte: identificazione e autorizzazione.
La procedura di identificazione provvede a fornire un'identificazione dell'utente al sistema, munito di un identificativo univoco.

Definiamo fattore di autenticazione un attributo che può essere impiegato da un utente per accedere al sistema.
Attualmente esistono cinque tipi di fattori.

- **Fattore di conoscenza**: costituisce un informazione che l'utente conosce. Possono essere costituisce da un PIN, un username, password o una risposta ad una specifica domanda
- **Fattore di possesso:** è una credenziale basata su un oggetto fisico che l'utente possiede: invio di one-time-password tramite SMS, chiavetta USB che genera un codice.
- **Fattore di inerenza:** informazione basata su dati biometrici dell'utente: scan della retina, impronta digitale, riconoscimento facciale.
- **Fattore di locazione:** basata sulla posizione dell'utente al momento della richiesta. Tipicamente utilizza sistemi di localizzazione come GPS.
- **Fattore temporale:** meccanismo che si basa sull'orario in cui la richiesta avviene.

I primi tre fattori sono quelli più comuni e utilizzati attualmente, in quanto gli ultimi due non sono affidabili e si basano su meccanismi tradizionalmente poco sicuri.

Nelle seguente sezioni mostrerò i processi di autenticazioni più utilizzati attualmente, suddivisi in base ai tipi di fattori che utilizzano.

2.3.1.1 Single-Factor Authentication

Questo tipo di identificazione fa affidamento ad un fattore di autorizzazione, spesso basato su una coppia di credenziali dell'utente, tipicamente login e password.

Durante il processo di identificazione le credenziali fornite dall'utente vengono confrontate con una lista di credenziali presente in un database attraverso un server di autenticazione.
Se le credenziali inviate dalla richiesta coincidono con quelle presenti sul server l'entità viene autorizzata a utilizzare le risorse del sistema.
Si assume che la conoscenza di una coppia di login e password sia sufficiente per garantire che l'utente che sta eseguendo la richiesta sia autentico.

2.3.1.2 Multi-Factor Authentication

I sistemi che richiedono più livelli di sicurezza le autenticazioni single-factor non garantiscono un livello alto di fiducia, in quanto ottenere un solo fattore facilmente reperibile (come le credenziali dell'utente) è molto semplice.

Aggiungendo fattori al processo di autenticazione tipicamente aumentiamo la sicurezza dei sistemi.
In questo tipo di autenticazione il processo include due fattori che aumentano le operazione di identificazione e

Normalmente un'autenticazione forte impiega due fattori, dove i fattori sono di tipologie differenti. La distinzione è importante in quanto se vengono utilizzati fattori della stessa categoria l'autorizzazione viene considerata Single-Factor.
Aggiungendo un fattore di un genere diverso diminuiamo il rischio che l'utente possa essere esposto ad un attacco uguale che ottiene un doppio risultato.

Tipicamente il primo fattore coincide con i sistemi Single-Factor, utilizzando delle credenziali conosciute dall'utente.
Comunemente questo viene combinato con sistemi biometrici o tramite possesso di un hardware particolare, invio di SMS.

Esistono inoltre sistemi Multi-Factor authentication che combinano più di due fattori per l'identificazione dell'utente.

2.3.1.3 Single Sign-On

Questa sezione è dedicata alla descrizione del funzionamento dell'autenticazione *Single Sign-On* (SSO).

Con SSO definiamo quel processo per cui l'utente può autenticarsi un'unica volta per poter accedere a più applicazioni o permettere che l'autenticazione sia valida per un'intera sessione di lavoro.

Questo procedimento richiede l'utilizzo di un repository di autenticazione a cui possono accedere tutte le applicazioni che si vogliono utilizzare. Questo approccio centralizzato viene utilizzato per garantire le misure di sicurezza e relative policy identiche a tutte le applicazioni utilizzate e agli utenti a cui vogliono accedere.

In aggiunta a un'identificazione semplificata due aspetti chiave di cui beneficia SSO sono l'aumento della produttività dell'utente e il costo operazionale ridotto, dovuto alla diminuzione di passaggi da compiere per l'autenticazione e al calo della ridondanza dei dati per le credenziali.
Con SSO gli utenti non devono più memorizzare password per tutte le applicazioni che devono utilizzare e, soprattutto, anche il reset in caso di password facilitata incrementa la facilità di gestione dell'accesso al sistema.

Un aspetto chiave di questa tipologia di accesso è la relazione tra SSO e l'identità dell'utente.
Definiamo come *Federated Identity Management* (**FiDM)** un insieme di policy, pratiche e protocolli che agiscono per collegare l'identità di un utente a cui dare fiducia alle applicazioni dell'azienda.
Possiamo quindi dire che SSO costituisce un sottoinsieme di FIDM in quanto si occupa unicamente di autenticazione e interoperabilità.
FIDM e SSO hanno il preciso scopo di eliminare ridondanza, ridurre costi, diminuire rischi e aumentare la sicurezza lasciando che un utente si autentichi una volta sola e che la sua identità possa essere utilizzata attraverso più sistemi per facilitare l'utilizzo da parte dell'utente.

Alcuni standard più rinomati per ottenere questa tipologia di identità sono *Security Assertion Markup Language* (**SAML)**, Open Authentication (**OAuth)** e OpenID.

2.3.1.3 Considerazioni

Di seguito verranno fatte una serie di considerazioni sulle tecniche di autorizzazione e dei loro meccanismi di difesa.
Come già anticipato nelle precedenti sezioni le autenticazioni che avvengono tramite

Single-Factor non sono sufficientemente considerate sicure in quanto tutti i fattori elencati in precedenza sono vulnerabili.
E' necessario quindi integrare sistemi di SFA con altri sistemi di autenticazione, trasformandosi quindi in Multi Factor Authentication.
Negli ultimi anni vi sono state una serie di trasformazioni relative in questo ambito. Oltre all'aggiunta di sistemi biometrici nei più comuni sistemi mobili, vi sono una serie di sistemi moderni che hanno permesso a tutti gli utenti di potersi autenticare in modo sicuro.
Un sistema che utilizza il famoso 2 Step Authentication è Universal Second Factor[5], o U2F. Il sistema prevede l'inserimento di una chiavetta che inserisce un codice e che serve all'utente per verificare la propria identità oltre alle credenziali fornite al sistema.
Un sistema simile è stato introdotto da Google, chiamato Authenticator, un'applicazione installata su un dispositivo diverso in possesso dall'utente che genera un codice ogni 3 minuti.
Ogni sistema ha i pregi e i suoi difetti, l'analisi provvede a costruire un sistema che include il minor numero di falle possibili in modo da rendere l'esperienza dell'utente gradevole e sicura.

[5] U2F, https://www.yubico.com/solutions/fido-u2f/

2.3.2 Soluzioni anti-DoS

Come descritto precedentemente, gli attacchi DoS e DDoS costituiscono un vettore molto pericoloso quando si tratta di esporre servizi o permettere a utenti di poter usufruire di potersi connettere alla rete di un'organizzazione.
Per questo motivo è necessario munirsi di una serie di strumenti per contrastare i più conosciuti attacchi che appartengono alla stessa categoria.

Come prima strategia è consigliabile utilizzare una banda di rete molto larga. Il primo obiettivo di un vettore DoS è quello di non permettere più ad altri utenti di usufruire del servizio bersagliato. Per questo motivo è opportuno utilizzare una banda larga per avere la possibilità di contrastare sul momento l'attacco e lasciando comunque disponibile il servizio ad altri utenti.

Come seconda strategia da adottare è la ridondanza del sistema. Avere un unico punto su cui esporre i servizi è alquanto limitato e nel caso di fallimento dell'architettura e del sistema tale servizio non è più fruibile e bisogna attendere tempi molto lunghi in caso di attacco per poter determinare l'origine e un metodo per contrastarlo. Per questo motivo è altamente consigliato diffondere lo stesso servizio, data-center in più luoghi diversi così ottenendo una ridondanza del servizio da erogare.

Aggiungere sistemi di protezione come *firewall* che hanno il compito di bloccare certe porte e regolare i flussi.
Come precedentemente descritto molti attacchi avvengono utilizzando protocolli particolari come UDP. Bloccare la porta 53, quella di UDP, blocca la possibilità di sfruttare un attacco ICMP Flood.

Nei prossimi capitoli verrà anche illustrata la tecnologia di *Reverse Proxy* che affronta diverse tematiche tra cui il bilanciamento del carico e altre funzionalità contro attacchi DDos.

2.3.3 Confidenzialità del canale di comunicazione

In questa sezione verrà discussa come è possibile mettere in sicurezza il canale di comunicazione tra client e server, facendo distinzione tra i vari Layer del modello OSI.

Definiamo un canale sicuro il modo di trasferire i dati da un punto A ad un punto B che sia resistente a manomissione di dati e da possibili agenti in ascolto.

2.3.3.1 IP Security

IP Security [RFC 2041, 2042, 2046, 2048], abbreviato spesso in IPsec, è un suite di protocolli che ha lo scopo di di mettere in sicurezza i servizi di IP.
Lo stack di protocolli sono stati progettati per criptare e autenticare i pacchetti del livello IP, oltre che provvedere un sistema di scambio di chiavi sicuro.

Definiamo come *Security Association,* spesso abbreviata con SA, una connessione logica unidirezionale tra due host. Quindi se si vuole stabilire una comunicazione tra client e server e si vuole comunicare tra entrambe le parti è necessario stabilire due SA, una client - server e una server - client.

E' possibile utilizzare IPSec in due diverse modalità: *trasporto* o *tunnel*.

La modalità trasporto adopera l'header per instradare il pacchetto criptato e quindi deve contenere un indirizzo instradabile su rete pubblica. Quindi in questa modalità viene criptato solo il messaggio e non gli indirizzi di origine e destinazione.

La modalità tunnel, invece, il pacchetto che deve essere protetto viene incapsulato in un altro pacchetto IP nella sua interezza.
Il pacchetto esterno definisce due endpoint, diversi da quelli del pacchetto interno, che vengono chiamati gateway di sicurezza e che hanno il compito di cifrare e decifrare i messaggi e poi inoltrare i messaggi all'endpoint stabiliti nel pacchetto protetto.

La differenza sostanziale è che nel primo modo i pacchetti hanno indirizzi visibili e non cifrati e che la fase di cifrazione e decifrazione avviene *end-to-end*. Nel secondo caso invece la cifrazione avviene all'interno di due gateway. E' evidente che il primo metodo ha delle performance più elevate mentre il secondo ottiene un livello di sicurezza maggiore. Il secondo metodo è quello più utilizzato comunemente poiché è quello che fornisce le comuni VPN che verranno discusse nel corso di questa dissertazione.

Citiamo i meccanismi più utilizzati: *Encapsulating Security Payload* (**ESP**) e *Internet Key Exchange* (**IKE**).

ESP (RFC 4303) definisce un header aggiuntivo che provvede confidenzialità attraverso criptaggio del pacchetto, protezione dell'integrità, autenticazione dell'origine, controllo di accesso e protezione contro analisi di traffico e replay.

IKE (RFC 2409) si occupa di scambiare le chiavi, gestirle dopo lo scambio, autenticazione e negoziazione dei parametri di sicurezza.

2.3.3.2 Secure Socket Layer e Transport Layer Security

Questa sezione è dedicata a come è possibile arricchire Il Layer di Trasporto, in particolare TCP, per ottenere un livello di sicurezza più alto.

SSL, *Secure Socket Layer*, definito in [RFC 6101], è un meccanismo crittografico atto ad aggiungere a TCP riservatezza, integrità dei dati e autenticazione del client e del server.
Il protocollo aggiunge un sottolivello tra il livello di trasporto e quello applicativo.

Come già anticipato SSL aggiunge dei passaggi obbligatori ad una tipica connessione TCP, sfruttando una crittografia a chiave asimmetrica.

Questa versione gettò le basi per il successore di SSL, **TLS** o *Transport Layer Security* [RFC 4346].
TLS è considerato un'evoluzione più efficace e sicura di SSL, in quanto le ultime versioni più aggiornate risolvono bachi che compromettono la sicurezza della comunicazione e contrastano attacchi a cui le versioni precedenti erano vulnerabili.

Durante una normale connessione TCP, nella fase di handshake, il client stabilisce con il server una serie di specifiche quali:

- **Versione del protocollo SSL / TLS da utilizzare**
- **Il tipo di crittografia**
- **Metodi di compressione (opzionale)**

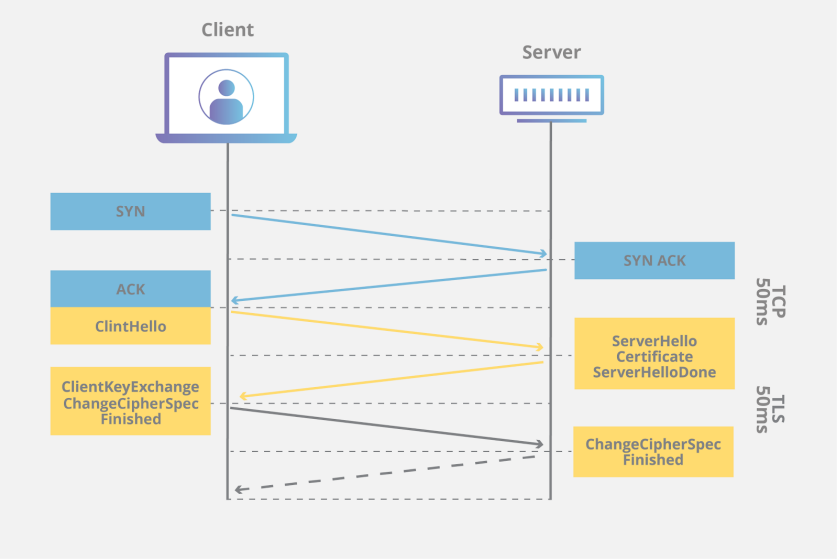

Figura 1.6: Handshake TLS

Il server controlla la versione più alta compatibile, il tipo di crittografia supportato e se specificato anche il tipo di compressione utilizzato.
Dopo la prima parte di setup, il server manda il certificato, rilasciato da un'autorità riconosciuta, al client.
Dopo aver verificato che il certificato ed essendo sicuri che il server sia autentico viene mandata una chiave ed entrambi, client e server, utilizzano la chiave per la cifrare i messaggi, e il server li decifra utilizzando la sua chiave privata.

Il client conferma che la comunicazione sarà crittografata con un messaggio autenticato, e inizia così una connessione.

Ora la connessione stabilita è sicura, crittografata, e abbiamo modo di capire che il server e il client siano autentici grazie all'utilizzo di certificati e della chiave scambiata.

Uno degli utilizzi più comuni di questi protocolli è *HyperText Transfer Protocol over Secure Socket Layer,* più conosciuto come **HTTPs**, nel quale il protocollo garantisce l'autenticità, l'integrazione dei dati di un sito web o del server web associato.

TLS costituisce in questo momento il protocollo in grado di mettere in sicurezza il livello di trasporto.

2.3.3.3 Certificato digitale

Un certificato digitale, spesso chiamato anche certificato a chiave pubblica, è un documento digitale per collegare la proprietà di una chiave pubblica ad un ente.

E' uno strumento utilizzato per crittografia e autenticazione dell'ente che fornisce la chiave pubblica, le informazioni che fornisce sull'entità della chiave pubblica e i relativi metadati annessi alla firma digitale.
I certificati digitali vengono utilizzati nella crittografia a chiave pubblica, dove la chiave pubblica che è condivisa viene autenticata dal certificato stesso o per autenticare una firma digitale.
Un utilizzo comune è per inizializzare in modo sicuro le connessioni SSL tra un web browser e un web server.

La grande maggioranza dei certificati vengono istituiti da un *Certificate Authority,* enti di terzi parti considerate di fiducia. Questo meccanismo fa sì che l'utente estenda fiducia dall'ente al certificato che emette.

2.3.4 Tabella riassuntiva

In questa sezione vengono mostrate le principali minacce verso un sistema a confronto con le tecniche di protezione illustrate precedentemente con lo scopo di dare una panoramica di ciò che è stato implementato durante lo svolgimento di questo sviluppo.

Accesso non Autorizzato	Livelli di Autenticazione
Attacchi MITM	IPSec VPN SSL / TLS (HTTPs)
Attacchi DoS	Bilanciamento del carico Ridondanza nei servizi Firewall Reverse Proxy

3 Tecnologie analizzate

In questo capitolo vengono spiegate nel dettaglio i principali componenti che sono state utilizzate nell'architettura finale per la gestione di una connessione da remoto.

Sono state riportate volutamente le tecnologie che ho ritenuto più interessanti per lo svolgimento di questa dissertazione.
Le tecnologie fondamentali che sono state individuate sono: reti MPLS, VPN e Reverse Proxy.

E' fondamentale evidenziare che le reti MPLS non costituiscono una tecnologia dell'architettura analizzata, ma costituiscono la rete interna tra le varie sedi dell'organizzazione in cui è stata svolta l'analisi.
E' stato quindi obbligatorio fornire un'analisi di tale tecnologia per determinare in quali casi l'accesso remoto non possa costituire una minaccia per tale struttura.

3.1 Reti Multi Protocol Label Switching

In questa sezione vengono illustrate le reti *Multi Protocol Label Switching* [RFC 3031], come funzionano e quali sono le loro implementazioni tipiche.

Prendendo in considerazione il tradizionale modello OSI, **MPLS** è una tecnologia che opera tra il livello 2, collegamento, e il livello 3, rete.

MPLS aggiunge nuove funzionalità al trasporto dei pacchetti sul protocollo IP.
Permette di gestire come il traffico viene ridirezionato all'interno di una rete, ripartire la capacità dei canali, amministrare una priorità diversa per i servizi e prevenire la congestione.

3.1.1 Concetti chiave

In questa sezione definirò alcuni concetti chiave essenziali per capire il funzionamento del protocollo **MPLS**.

LSP (*Label Switched Path*) costituisce il tunnel unidirezionale tra il router di ingresso e il router terminale.

LER (*Label Edge Router*) è il router d'ingresso. Ha il compito di calcolare il path di trasmissione e di etichettare i pacchetti all'interno di un LSP.

LSR (*Label Switching Router*) è un router all'interno di un LSP, che l'unico scopo di ridirezionare i pacchetti dentro il path prestabilito.

Egress Node è il router finale dell'LSP e rimuove l'etichetta dai pacchetti.

3.1.2 Label Switching

Il Label Switching è la tecnica utilizzata dalle reti MPLS per inoltrare e gestire il traffico all'interno di un LSP.
Il LER , il primo router del path, esegue un ***lookup***, stabilisce il router di destinazione e predetermina un percorso.
Il router quindi provvede a etichettare il pacchetto in modo che tutti i router situati tra il router di partenza e quello di destinazione utilizzino l'etichetta per inoltrare il pacchetto senza la necessità di eseguire ulteriori lookup.

3.1.3 Fast Reroute

Fast Reroute è un procedimento che fa parte del protocollo di instradamento delle reti MPLS che viene messo in atto in caso di fallimento di nodi all'interno del percorso.

Solitamente in una normale rete IP un router il calcolo per il miglior percorso viene richiesto in caso di fallimento. Questa operazione può dilungarsi per alcuni secondi e possono essere calcolati dei loop.

Le reti MPLS adottano invece Fast Reroute: vengono inseriti dei cammini di backup nel caso un nodo fallisca a recapitare il messaggio o non sia più accessibile. In questo modo il ricalcolo di un nuovo percorso non deve avvenire e l'inoltro dura poco più di qualche milionesimo di secondo. Dato che l'interno cammino è calcolato dentro un LSP è impossibile che vengano calcolati dei loop.

Tramite questa tecnica MPLS garantisce un meccanismo di difesa contro attacchi di DoS verso un router nella rete, in quanto se non fosse più disponibile sia già stato prestabilito un percorso di backup per arrivare alla destinazione.

3.1.4 Osservazioni

Per questioni di sicurezza la rete MPLS non dev'essere visibile da reti esterne come Internet o qualsiasi VPN connessa. In questo modo si ottiene una visibilità limitata dei nodi della rete, e gli indirizzi non sono conosciuti, prevenendo così una serie di attacchi DoS.

3.2 VPN

In questa sezione verrà illustrata cos'è la tecnologia VPN e il suo funzionamento e i suoi principali utilizzi.

VPN, acronimo di *Virtual Private Network,* è una tecnologia che crea una connessione sicura crittografata sopra una rete meno sicura, principalmente utilizzata per permettere agli utenti di accedere ad una rete privata da remoto.

E' una tecnologia che crea un tunnel su una rete pubblica non sicura, come Internet, tra due macchine o reti private. Questa operazione è detta *tunneling.*

Nella nostra analisi questa tecnologia è fondamentale in quanto un client che cerca di impiegare un servizio da remoto deve ottenere un tunnel sicuro di comunicazione con l'intranet aziendale. VPN costituisce uno degli elementi fondamentali per raggiungere l'obiettivo e uno degli strumenti più efficaci per prevenire un numero molto alto di attacchi sul canale comunicativo.

L'esigenza attuale è quella di poter accedere alla rete non sicura in modo che nessun agente esterno possa essere in ascolto o alterare i dati della comunicazione, quindi prevenire tutti gli attacchi MITM.

Allo stato attuale, esistono una serie di VPN differenziate in base al tipo di implementazione, l'impiego di crittografia e in base al livello del modello OSI in cui avviene il tunneling dei dati.

Tra le VPN più comuni ricordiamo **PPTP VPN**, *Point-to-Point Tunneling Protocol* VPN, dove non è richiesto nessun tipo di hardware particolare, è richiesto l'accesso autenticato tramite login e password.
In questa tipologia di VPN non vi è nessun utilizzo di crittografia.
Nelle sottosezioni successivi sono elencate e descritte le principali VPN che sfruttano, invece, elementi di crittografia e meccanismi di sicurezza.

Evidenziamo che le VPN IPSec e SSL o TLS VPN sono utilizzate su rete pubblica, mentre per quanto riguarda le VPN MPLS sono utilizzate per reti private.

3.2.1 IP Security

In questa tipologia di VPN, la costruzione del tunnel avviene a livello di rete.

Per l'implementazione di VPN IPSec, viene principalmente utilizzata la modalità tunnel di IPSec, in modo tale da garantire che nessun dato sensibile, quali l'origine e la destinazione dei pacchetti, possano essere alterati o ascoltati.

Un grandissimo vantaggio per questo strumento è la possibilità di aggiungere questo livello di sicurezza oltre ad altri strumenti che coprono gli altri livelli di comunicazione. E' invece negativo il fatto che l'installazione e l'implementazione di questa tipologia di VPN richiede molto tempo e spesso anche molto costosa in termini di hardware e software.

3.2.2 Secure Socket Layer e Transport Layer Security

Questa tipologia di Virtual Private Network utilizza protocolli di sicurezza precedentemente illustrati, quali SSL e TLS.
Principalmente queste VPN sono standard per web browser per provvedere sicuro accesso da remoto.
Tipicamente utilizzano un algoritmo di cifrazione *end-to-end* (**E2EE**).
Organizzazioni usano queste VPN per permettere a utenti da remoto per accedere in sicurezza a risorse interne aziendali, oltre a provvedere accesso a Internet a utenti che vi accedono dall'esterno.

Come precedentemente detto, durante gli anni il protocollo SSL è stato deprecato, dando spazio al suo successore TLS, e i moderni browser attualmente utilizzano TLS come protocollo di cifrazione e di autenticazione per i dati trasmessi attraverso la VPN.

Le VPN SSL agiscono quindi a i livello di trasporto, quindi il traffico di rete può essere facilmente essere inoltrato in circuiti tunnel per proteggere accesso alle risorse o applicazioni.

Esistono due tipologie principali di VPN SSL: VPN *portal* e VPN *tunnel*.
La prima tipologia si riferisce ad una connessione a tempo a un sito web in remoto.
Gli utenti in remoto accedo al gateway SSL attraverso il loro web browser dopo essersi autenticati attraverso metodi supportati dal gateway.
Il secondo tipo, invece, abilita la possibilità all'utente di connettersi a servizi multipli tramite web browser, ad altri servizi o protocolli non basati sul web.
Sostanzialmente il VPN tunnel agisce come un circuito stabilito tra l'utente in remoto e il server VPN, il server connette a uno o più siti web remoti a posto del client.

3.2.3 Per-App VPN

Nelle VPN tradizionali, la sicurezza veniva garantita al device utilizzato nella sua interezza, cioè una volta attiva la VPN, tutti i dati dell'utente, tutte le applicazioni, il traffico, passa attraverso il tunnel VPN.
Alcune organizzazioni preferiscono che solo alcuni dati o alcune applicazioni utilizzino la VPN fornita.
Per questo motivo sono nate le per-app VPN, che limitano l'utilizzo della VPN unicamente ad un set di applicazioni scelte dall'amministratore.

Le VPN Per-App garantiscono un livello superiore di sicurezza per una connessione da remoto in quanto solo una serie di endpoint sono stabiliti e solo quindi un insieme limitato di risorse interne sono raggiungibili dall'esterno.

Questa per - app VPN riguardano unicamente dispositivi mobili, in quanto la lista di applicazioni che partecipa a questa tipologia di VPN viene collezionata dal Mobile Device Management.

3.2.2 Considerazioni

In quest sotta sezione verranno discussi i vantaggi e gli svantaggi dell'utilizzo di una VPN.
Come prima osservazione è necessario specificare che le VPN hanno un buon risultato e sono funzionali se vengono utilizzati agli estremi di un firewall che diventa ponte e regolare del traffico tra i due estremi della VPN.
Un rischio che si può verificare è quello di lasciare aperta la sessione della VPN con il dispositivo fruibile fisicamente da altre persone che possono guadagnare accesso alla rete interna aziendale.

La tecnologia VPN parte con un difetto non indifferente. In caso di accesso tramite Browser, tutto il traffico che parte dall'utente viene inoltrato sulla rete VPN e quindi raggiunge l'estremo.
Per ovviare a questo problema esiste una tecnica chiamata Split tunneling, tecnica che permette di accedere sia a endpoint stabiliti tramite VPN e accedere a normali IP tramite rete tradizionale.

La tecnica di Split Tunneling può essere applicata sia a SSL VPN che IPSec VPN.
Al momento le VPN SSL sono largamente più utilizzate rispetto a IPSec VPN in quanto l'accesso remoto tramite IPSec richiede un'installazione di software IPSec sul client e richiede addizionalmente l'acquisto di software aggiuntivo.
D'altra parte SSL VPN utilizzano browser moderni e hanno una configurazione minimale richiesta.

3.3 Reverse Proxy

Questo capitolo è dedicato ai Reverse Proxy, da cosa sono più comunemente costituiti e quali sono i principali utilizzi.

Un reverse proxy è un server intermediario situato dentro una rete privata e ha il compito di inoltrare le richieste di un client verso il server appropriato. In questo modo il reverse proxy aggiunge un livello di astrazione e di controllo addizionale che assicura il flusso di traffico tra client e server.

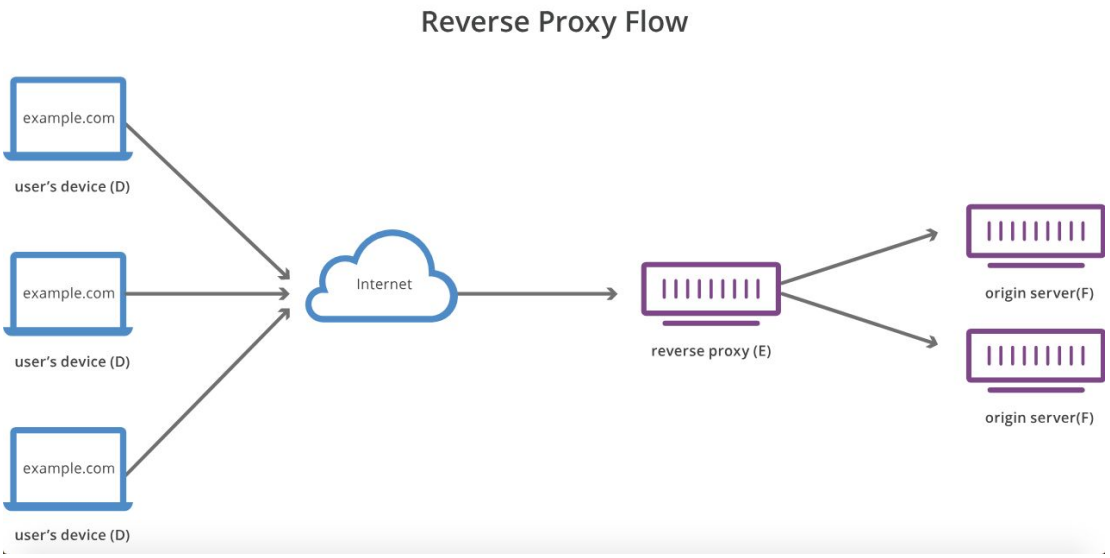

Figura 1.7: Flusso di traffico verso un Reverse Proxy

Consideriamo il flusso tipico di una richiesta client - server, prendendo come riferimento la figura di cui sopra.
Il client D comunemente fa una richiesta al server F che passa attraverso una rete, in questo caso Internet, e F risponde direttamente a D.
Il reverse proxy, in questo caso E, si frappone tra il client e il server, intercettando le richieste del client e inoltrando i dati al server per poi ottenere una risposta dal server e mandare la risposta al client.

Gestire in questo modo il traffico garantisce una serie di benefici lato server:

- **Gestione del carico:** il reverse proxy agisce come semaforo sulle richieste al server, distribuendo i dati ad un gruppo di server in modo da ottimizzare velocità e capacità. In questo modo si garantisce che nessun server sia sovraccarico di richieste. Nel caso in cui un server non sia più disponibile, il reverse proxy redirige il traffico verso un altro server.
- **Caching:** un reverse proxy può contenere in cache dati e restituire una risposta senza passare per il server. In questo modo si migliorano le prestazioni e si evita di inoltrare richieste al server.
- **Accelerazione Web:** oltre alla possibilità di utilizzare memoria cache, il reverse proxy permette anche di utilizzare protocolli SSL o TLS per decifrare il contenuto dei messaggi senza sovraccaricare il web server.
- **Anonimità del server:** intercettando le richieste per il server, l'identità stessa del server viene nascosta dal reverse proxy e aggiunge un livello di difesa aggiuntivo.
- **Protezione dagli attacchi:** la maggioranza dei reverse proxy attualmente in commercio comprende già una serie di sistemi di sicurezza che garantisce a tali di contrastare gli attacchi DoS, aggiungendo così un grado di sicurezza maggiore per la gestione dei server.

Come sottolineato più volte, la principale applicazione dei Reverse Proxy, oltre ad essere altamente performanti, garantiscono una linea di difesa contro attacchi DoS, nascondendo l'identità quindi l'indirizzo fisico del server.

3.3.1 Web Application Firewall

In questa sezione mostrerò cosa sono i *Web Application Firewall*, il loro funzionamento e le loro applicazioni.

I WAF sono una tipologia di Reverse Proxy che si interpone tra la rete e un web server, controllando il flusso del traffico.

In particolari, i WAF filtrano e monitorano il traffico HTTP tra l'applicazione Web e Internet. Tipicamente quindi protegge il server dai tipici attacchi che riguardano il mondo Web come *Cross-Site-Scripting*, *SQL Injection, cross-site-forgery, DNS spoof.*

Esistono vari tipi di WAF attualmente, un metodo utile per classificarli è in base al modello che di sicurezza che implementano.

- I *Blacklist* WAF*,* basati su un modelli negativo di sicurezza, proteggono contro attacchi conosciuti.
- I *Whitelist* WAF*,* invece basati su modelli positivi di sicurezza, ammettono solamente traffico che è stato pre approvato.

Entrambi i modelli offrono pro e contro differenti, spesso però i prodotti sul mercato sono un ibrido tra i due tipi.

Un altro metodo di classificazione dei WAF è basato invece sul tipo di implementazione e sulla costruzione di tali strumenti :

- **WAF di rete:** spesso basati su hardware, installati localmente e sono tra i più costosi in quanto esigono prodotti fisici e manutenzione elevata
- **WAF di applicazione:** integrati interamente lato applicazione. Offrono una maggiore personalizzazione e possibilità di adattare il prodotto al sistema. Alta complessità di sviluppo, costo di risorse elevato e costi alti di manutenzione
- **WAF basati su cloud:** costo per l'implementazione e deploy è il più basso tra i vari, costi di manutenzione molto bassi, continuamente aggiornati per proteggere da nuove vulnerabilità e attacchi. Il server deve però lasciare che la responsabilità di questi Proxy sia delegata a terze parti e spesso molte funzionalità sono nascoste.

4 Analisi architetturale

In questo capitolo viene discussa l'architettura di F5 BIG-IP APM, i componenti di cui è costituita e le valutazioni relative alle scelte implementative.
Nella seconda parte di questo capitolo viene fatto un focus su una Proof of Concept che è stata sviluppata durante l'analisi di questa dissertazione.

F5 BIG-IP APM (Access Policy Manager) è una soluzione di F5 che provvede a garantire un accesso globale unificato alla rete, al cloud e alle applicazioni dell'organizzazione, convergendo e consolidando, accessi remoti, mobili e web.
I principali aspetti chiave che garantisce questa soluzione sono un accesso tramite SSO (Single Sign-on) tramite Multi Factor Authentication (MFA) e identità federativa.

4.1 Componenti architetturali

Questa sezione è dedicata ai componenti che possono costituire un'architettura con soluzione F5.
In particolare sono curati i componenti di cui sono stati delineati gli aspetti più importanti relativi allo svolgimento di questa tesi.
Viene quindi spiegato il funzionamento di una VPN F5, da che cosa può essere costituita.
In questo capitolo è stato curato in modo particolare le possibilità di autenticazione che una soluzione F5 offre e messa a confronto con quella che attualmente BPER Banca utilizza.

4.1.1. Virtual Private Network

Al momento dell'analisi durante lo svolgimento di questa dissertazione, BPER Banca ha messo a disposizione la possibilità dell'installazione di un certificato solo tramite un file .exe eseguibile unicamente su sistemi con Windows installato, versione 7 o superiore.
Per cui l'installazione del certificato e l'utilizzo della VPN è stata svolta in un computer Windows.
Ad ogni avvio viene attivata una VPN F5, che sfrutta le tecnologie SSL per accedere ai servizi interni di BPER Banca.

4.1.2 Security Assertion Markup Language

Security Assertion Markup Language, detto **SAML**, è uno protocollo standard di sicurezza per scambio di informazioni legate all'autorizzazione e autenticazione, dette *asserzioni,* tra due domini differenti, un *Identity Provider,* ente esterno che fornisce il livello di autenticazione degli utenti e un *Service Provider,* che fornisce i servizi a tali clienti.
Nei sistemi precedenti questi due livelli erano uniti, un'organizzazione forniva entrambe le cose, ma oggi, spesso, queste due tipologie di servizio vengono erogate da due fornitori differenti.
SAML costituisce un ponte di comunicazione tra le due entità, cercando di fornire Web-SSO tra entità che appartengono a domini distinti.

4.1.2.1 Funzionamento

Figura 1.8: Flusso di funzionamento di SAML

Lo standard SAML trasferisce l'identità dell'utente dall' Identity Provider al Service Provider, che avviene tramite lo scambio di asserzioni in formato XML.

Il processo avviene seguendo i seguenti passaggi:

l'utente accede da remoto ad un servizio o applicazione, che redirige l'utente verso l'IdP chiedendogli di autenticarsi.
A questo punto può avvenire che l'utente abbia attiva una session sul proprio web browser attiva, altrimenti gli sarà chiesto di effettuare il login stabilendone una.
Identity Provider ora costruisce la risposta di autenticazione creando un certificato X.509 e inoltrando tale pacchetto al service provider.
A questo punto il servizio applicativo, che ha già fiducia dell'identity provider, accetta l'autenticazione dell'utente e effettua il login.

Questo processo consiste nel fornire la pratica SSO a tutte le applicazioni che hanno fiducia nell'Identity Provider.

4.1.3 Lightweight Directory Access Protocol

Lightweight Directory Access Protocol, acronimo **LDAP**, è un protocollo per accedere ai servizi di directory basati sul protocollo *X.500*, utilizzato per l'autorizzazione e autenticazione di utenze in un'architettura client - server centralizzata.

Con il termine di entry definiamo una collezione di informazioni relative ad un'entità. Ogni entry è costituita da tre differenti tipi di informazioni: Distinguished Name, una collezione di attributi e una collezione di oggetti di classe.

Ogni entry fa riferimento ad una *Distinguished Name* o **DN,** che identifica unicamente la entry e la sua posizione nell'albero.
Una DN è costituita da zero o più Relative DN (o **RDN)** che a loro volta sono costituite da coppie attributi-valori.
Esiste un DN particolare costituito da zero RDN, spesso chiamato DN nullo e si riferisce alla root dell'albero e contiene le informazioni sul contenuto e le capacità della directory.

Gli attributi contengono le informazioni relative alla entry. Ogni attributo contiene un tipo, opzioni e un set di valori con gli effettivi dati.

I tipi di attributo sono schemi di elementi che specificano come gli attributi sono trattati dai client e server LDAP. Tutti gli attributi devono avere un identificativo dell'oggetto, o OID, e zero o più nomi che possono essere usati come riferimento agli attributi di quel tipo. I tipi di attributo possono anche indicare quando un attributo ha il permesso di avere più valori nella stessa entry, se l'attributo è utilizzato per contenere informazioni sull'utente o se utilizzato per le operazioni sul server.

Le classi di oggetto sono elementi di schema che specificano una collezione di tipi di attributi che possono essere relativi a un particolare oggetto, processo o entità.
Ogni entry ha una classe strutturale di riferimento che indicano che tipo di oggetto rappresenta.

L’informazione è organizzata in una struttura gerarchica ad albero, cominciando a livello di organizzazione, gruppi e singole entry di persone aziendali.

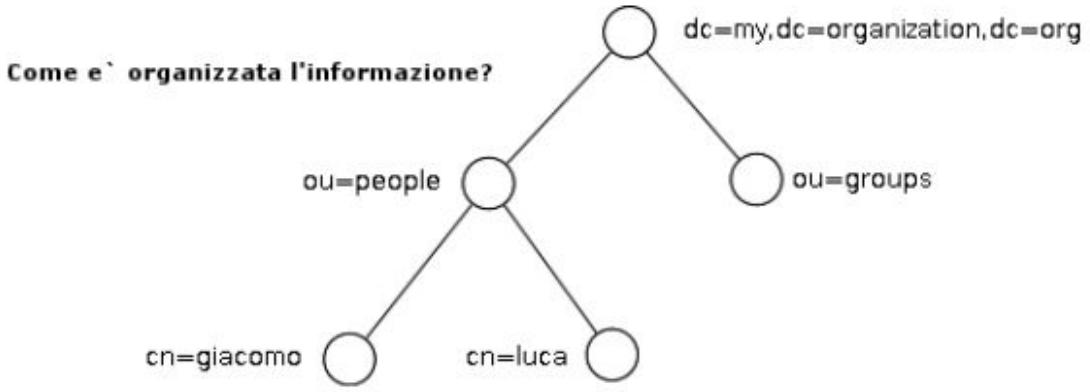

Figura 1.9: Organizzazione gerarchica dell’informazione in un server LDAP

Nella figura prendiamo in considerazione il DN finale in basso a sinistra.
Tale DN è costituito dai RDN (cn=giacomo,
ou=people,dc=my,dc=organization,dc=org).

L’autenticazione avviene tramite risoluzione di DN. All’input dell’username o email il server LDAP esegue una ricerca sugli input inseriti su tutte le entry degli utenti finché non ha un riscontro. Tipicamente le Directory adottano un sistema di indicizzazione e di caching efficienti per cui i tempi di ricerca spesso sono molto brevi.
Per validare la password inserita LDAP fornisce un’operazione di BIND.

Un aspetto molto importante dei codici di risposta al login su LDAP corrisponde al fatto che non rivela la causa di del fallimento dei login, che invece è riportato nei log del sistema.

Come anticipato LDAP fornisce un controllo di utenze centralizzato, il che significa che per ogni servizio vi è l’obbligo di autenticare la propria identità al server.

4.1.4 Secure Web Gateway Services

In questa sezione viene spiegato l'utilizzo del componente di F5 chiamato *Secure Web Gateway Services*, un moderno forward proxy in grado di donare all'architettura un livello alto di sicurezza web.

Rispetto ai forward proxy tradizionali, **SWGS** ha una serie di vantaggi non indifferenti che sono critici per un'architettura sicura.

Attraverso la tecnica *URL filtering*, tecnica che controlla accesso a siti esterni, applicazione web, protocolli e file multimediali, questo proxy provvede ad una serie di meccanismi per impedire che vengano eseguite azioni non lecite.

All'interno di questo prodotto viene integrato un sistema di rilevamento malware, che contiene un motore di analytics per web malware, firme, e sistemi euristici di rilevamento che identificano e eradicano minacce generiche e specifiche.

Quando un utente remoto accede al web tramite VPN per-app tunnel SWGS protegge la sessione come se l'utente accede tramite la rete interna dell'organizzazione.

Una funzionalità aggiuntiva di questo meccanismo è l'ispezione in partenza di pacchetti SSL per bloccare contenuti di siti che utilizzano SSL per questioni di privacy o di disciplina.

4.1.5 Mobile Device Management

Mobile Device Managent, o **MDM**, è uno strumento che semplifica la gestione dei dispositivi mobili dei dipendenti di un'organizzazione.
Sono piattaforme che si occupano della gestione centralizzata dei dispositivi mobili.
Non sono direttamente gestite da BPER ma da terze parti, in questo sono gestite interamente da VMWare con AirWatch.

Le funzioni base del MDM sono la geolocalizzazione del dispositivo smarrito, e in alcuni casi anche la cancellazione di dati su dispositivi smartphone iOS e Android nel caso venissero distribuiti direttamente dal fornitore di MDM.
Oltretutto, hanno lo scopo di funzionare da app store aziendali, dove i dipendenti ottengono applicazioni specifiche aziendali che non sono reperibili negli Store comuni.

Lo scopo principale di queste piattaforme è limitare l'utilizzo delle applicazioni e dei dati, applicazioni sui dispositivi che rispettano delle policy aziendali.

Le funzionalità di un MDM sono:

1) Gestione e limitazione delle applicazioni installate sul dispositivo
2) Gestione VPN per-app
3) Rilevamento minacce sui dispositivi

Come precedentemente illustrato, vi sono una serie di pratiche che permettono agli utenti di alterare il normale funzionamento del dispositivo, quali Jailbreak o Rooting, e spesso questi strumenti hanno la capacità di rilevare quando il dispositivo che tenta di accedervi abbia questo tipo di alterazioni.

4.1.6 Advanced Web Application Firewall

I tradizionali WAF garantivano sicurezza ai server web di fronte ai tradizionali attacchi basati sul web, come SQL injection, Cross-Site Scripting.
Un aspetto negativo è che questi strumenti tradizionali non interagivano con falsi positivi e determinate complessità operazionali.
Tendenzialmente gli WAF si basano su passivi filtri basati su metodi utilizzati per rilevare payload malevoli e per il controllo di protocolli utilizzati che siano disciplinati dalle policy dell'organizzazione.
La difficoltà che gli WAF affrontano oggigiorno è l'impossibilità di riconoscere e rilevare quando un'attività viene svolta da un bot o da un utente autentico, nonostante l'utilizzo di CAPTCHA e altri meccanismi.

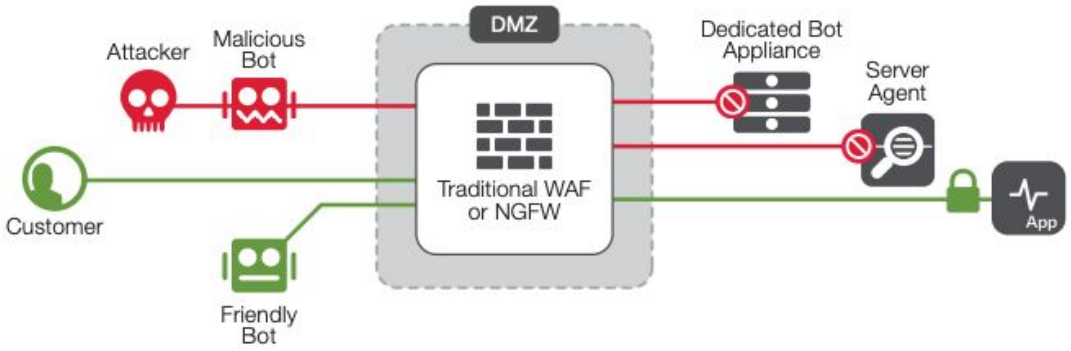

Figura 1.10: Web Application Firewall tradizionali

In questo tipo di applicazioni spesso il browser risulta essere l'anello debole del flusso. Tramite attacchi di phishing contenuti in mail o post sui social media, l'attaccante tenta di iniettare codice malevolo che può rendere il pc infettato parte di una botnet. Comunemente questi tipi di attacchi possono prendere forma di *Remote Access Trojan*, *keylogger* o altri metodi per ottenere dati sensibili dalle organizzazioni.
Il client spesso è ignaro che il proprio dispositivo sia infetto e possa effettivamente danneggiare l'architettura dell'organizzazione.

Questo componente, oltre ai sistemi tradizionali di un WAF, integra anche un sistema per l'analisi comportamentale e un sistema di rilevamento per iniezione del codice a runtime, tramite algoritmi di machine learning.
Advanced Web Application Firewall ha fatto un passo in avanti in questo senso, evolvendo verso un controllo attivo, capace anche di interrogare il richiedente.

Inizializzando il sistema con una profilazione di un normale flusso di traffico, gli algoritmi di Machine Learning riescono a rilevare traffico anomalo molto facilmente.

Riepilogando gli Advanced WAF hanno la capacità di:

1) Mitigare in maniera completa delle minacce bot web e mobile
2) Protezione contro eventuali tentativi di furti e abusi di credenziali
3) Rilevamento di DDoS a livello applicativo attraverso tecniche di analisi comportamentale

4) Aumentare le performance grazie a un controllo attivo che permette al componente di non interpellare più componenti esterni per rilevare minacce

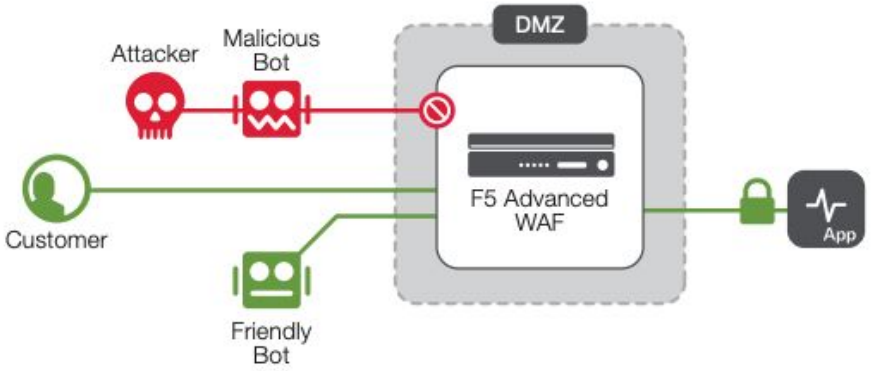

Figura 1.11: F5 Advanced Web Application Firewall

4.1.7 Architettura finale

Al momento dei test e della Proof of Concept che è stata deployata il sistema BIG IP APM includeva unicamente i moduli di Web Application Firewall e di accesso tramite VPN.
Il sistema di autenticazione utilizzato è tramite LDAP.
Il flusso comincia con il client che tramite macchina Windows e Internet Explorer fa una richiesta ad un server interno ai servizi BPER Banca tramite accesso LDAP.
Il server esegue una scansione tramite Internet Explorer con plugin di ActiveX per controllare che sul sistema vi sia installato un software di rilevamento malware.

Dopodichè viene attivata e scaricata la VPN F5 tramite la quale è possibile connettersi alla intranet di BPER tramite BIG IP APM.
A questo punto LDAP fornisce i livelli di autenticazione e autorizzazione necessari per eseguire l'accesso ai servizi a cui l'utente è registrato.
Il WAF viene utilizzato sulle macchine di frontiera della Demilitarized Zone pubblica, che espone servizi di reverse proxy e di load balancing sulla rete internet.

4.2 Dettagli Implementativi

In questo capitolo viene illustrato il codice e i dettagli della Proof of Concept sviluppata durante l'analisi dell'architettura.

Al momento dell'analisi durante lo svolgimento di questa dissertazione, BPER Banca ha messo a disposizione la possibilità dell'installazione di un certificato solo tramite un file .exe eseguibile unicamente su sistemi con Windows installato, versione 7 o superiore. Per cui l'installazione del certificato e l'utilizzo della VPN è stata svolta in un computer Windows e relativi sviluppi.

4.2.1 Client

Per motivi legati a limitazioni dovuti alla banca, e certificati disponibili solo per macchine Windows, il client di questa PoC era una macchina Windows che accedeva ai server di Numera di BPER Banca tramite Browser.

Il client ha la necessità di predisporre una VPN F5 per usufruire di questi servizi altrimenti gli endpoint non saranno raggiungibili.

4.2.1 Server

BPER Banca mi ha messo a disposizione una utenza presso le macchine della loro intranet che mi hanno permesso di fruire alcuni servizi accedendovi tramite VPN F5.

Il deploy del Mockup è avvenuto tramite un repository GitLab situato nelle macchine nella Intranet aziendale e tramite Jenkins[6], un sistema open source che permette di implementare flussi di Continuous Integration o Continuous Delivery.

[6] Jenkins, https://jenkins.io/

Nel momento in cui avviene il push nel repository del progetto, Jenkins avvia un processo di Continuous Delivery nel quale esegue una build di progetto e esegue un deploy internamente su OpenShift[7] una Platform as a Service (PaaS) prodotta da RedHat, per applicazioni cloud.

In questo modo l'applicazione web che è stata deployata su Red Hat è accessibile tramite VPN F5 e visualizzabile tramite Browser

La PoC sviluppata si tratta molto semplicemente di un sistema di prenotazione per le sale riunioni.

In quanto Proof of Concept non ha un utilizzo reale ma è stata sviluppata solamente con lo scopo di dimostrare che VPN F5 è facilmente utilizzabile e possibile fruibile da tutti gli utenti anche non tecnici.

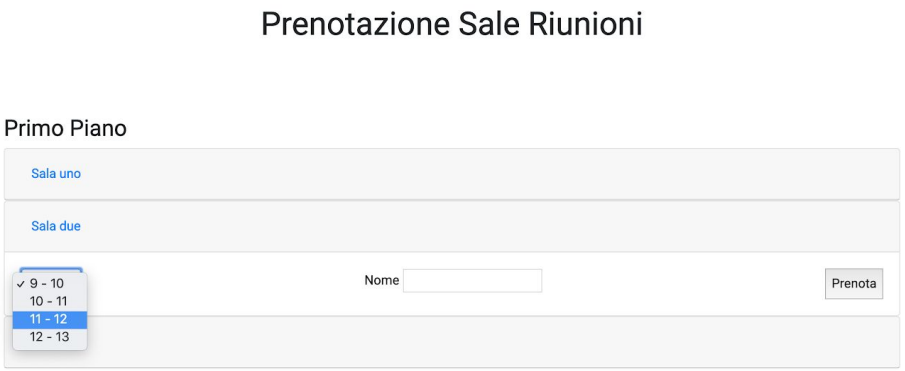

Figura 1.11: View della PoC sviluppata

[7] OpenShift, https://www.openshift.com/

5 Conclusione

In questo capitolo viene eseguita un'analisi finale dei vari componenti utilizzati, di ciò che non è stato preso in considerazione durante lo svolgimento della dissertazione e durante gli sviluppi della Proof of Concept.

5.1 Considerazioni finali

Con questa dissertazione ho voluto analizzare dettagliatamente i singoli componenti che riguardano un'architettura client server sicura.
L'obiettivo prefissato era quello di mostrare i rischi e le possibili soluzioni a tali minacce, rispondendo con una soluzione concreta.
Ogni componente di questo flusso ha un'importanza innegabile, partendo dall'autenticazione all'utilizzo di componenti che permettono di mitigare le varie minacce.
Vi sono una serie di aspetti che non sono stati coperti durante lo svolgimento della dissertazione e riguardano soprattutto molte tipologie di attacco che possono avvenire, quali iniezione di malware, backdoor, sviluppo di bot.
Il prototipo da me sviluppato voleva dimostrare la facilità e la possibilità di ottenere un risultato sicuro e veloce che tutti gli utenti finali possono usufruire.

5.2 Sviluppi futuri

Dalle analisi svolte sono emerse una serie di criticità che i test non hanno preso in considerazione. Una futura versione ha la necessità di tenere conto di questi fattori.

Il primo fattore da analizzare è legato all'autenticazione. Nel nostro test l'accesso e autorizzazione avviene tramite LDAP che richiede unicamente Single-Factor Authentication, ossia tramite credenziali.

Come precedentemente illustrato, SFA non risulta sicuro sufficientemente, per cui in futuro è necessario integrare questo sistema con altri fattori di autenticazione, biometrici o di possesso, passando quindi a Multi-Factor Authentication.
Oltre a questo BIP IP APM rende possibile l'inserimento a moduli di componenti aggiuntivi e diversi per l'autenticazione.
Tra i componenti per l'autenticazione citiamo Kerberos[8], standard prodotto da MIT, che permette anche l'autenticazione tramite SSO, che rende più semplice e sicura l'autenticazione.

Un altro fattore analizzato ma non testato riguarda la presenza di codice malevolo presente nei dispositivi del client.
Una policy sulla presenza di un software per la scansione e rilevamento di Malware è buona pratica nelle maggiori aziende che provvedono alla distribuzione di dispositivi ai dipendenti

Una pratica che è stata messa in atto riguarda gli sviluppi di applicazioni sicure.
Abbiamo visto che è stato impiegato un flusso di CD per quanto riguarda le applicazioni Web. E' da tenere conto il fatto che è necessario stilare una serie di policy che limitano la possibilità di ottenere codice non sicuro.
Come già anticipato questa soluzione di F5 permette l'inserimento di una serie di moduli che incrementano il livello di sicurezza in modo significativo.
Durante lo svolgimento di questa tesi sono stati coperti pochi di questi per via di limitazioni della Banca.

Queste considerazioni, non del tutto mature, fanno parte di un ambito in continua ricerca. La sicurezza che riguarda questa tipologia di architettura è in continuo movimento e trasformazione e questa dissertazione dimostra come è possibile contrastarle.

[8] Kerberos, http://web.mit.edu/kerberos/

6 Bibliografia

[1] **Andrew S. Tanebaum, David J. Wetherall**, *Reti di Calcolatori*

[2] **James F. Kurose, Keith W. Ross.** *Reti di Calcolatori e Internet, un approccio top-down*

[3] https://www.cisco.com/en/US/tech/tk436/tk428/technologies_white_paper09186

[4] https://tools.ietf.org

[5] https://www.cisco.com/c/en/us/td/docs/net_mgmt/vpn_solutions_center/2-0/mpls

[6] https://www.sans.org/reading-room/whitepapers/vpns/openvpn-ssl-vpn-revolution

[7] https://oauth.net/

[8] https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/

[9] https://www.microsoft.com/en-us/research/wp-content/uploads/2017/03/

[10] https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf

[11] https://www.android.com/security-center/

[12] https://networks.nokia.com/solutions/threat-intelligence/infographic

[13] https://www.microsoft.com/en-us/research/wp-content/uploads/2017/03