Cybersecurity

Corso tenuto dalla Professoressa Federica Paci

Università degli Studi di Verona

Alessio Gjergji

Indice

Capitolo 1

Cyber Kill Chain

1.1 Introduzione

La funzione principale della **cybersecurity** è proteggere i dispositivi che utilizziamo e i servizi a cui accediamo da accessi non autorizzati, danni o abusi. Essa mira anche a prevenire l'accesso non autorizzato a grandi quantità di dati salvati sia sui dispositivi e online.

1.1.1 Principi fondamentali della cybersecurity

Gli elementi fondamentali della cybersecurity sono:

- **Confidenzialità**: garantisce che i dati siano accessibili solo a chi è autorizzato.
- **Integrità**: assicura che i dati non siano alterati da persone non autorizzate.
- **Disponibilità**: rende i dati accessibili quando necessario.
- **Autenticazione**: verifica l'identità di un utente per accertarne la legittimità.
- **Autorizzazione**: assicura che l'utente abbia i permessi necessari per accedere ai dati.
- **Safety**: sistemi progettati e funzionanti in modo sicuro.
- **Accountability**: garantisce che le azioni degli utenti siano tracciabili.

1.1.2 Asset

Definizione di Asset

Un **asset** è qualsiasi elemento che ha valore per un'organizzazione. Tra gli asset rientrano persone, dispositivi, sistemi `IT`, network, software e ogni altro elemento a cui si può attribuire un valore.

1.1.3 Concetti chiave della cybersecurity

- **Vulnerabilità**: un bug, difetto o debolezza di un'applicazione, sistema o servizio che potrebbe compromettere le sue proprietà di sicurezza.
- **Cyber Threat**: una potenziale minaccia che potrebbe sfruttare una vulnerabilità per compromettere un asset.
- **Attacco**: la concretizzazione di una minaccia (*cyber threat*) che impatta negativamente su un asset.
- **Threat Actor**: un'entità (*es. individuo, gruppo o organizzazione*) che sfrutta una vulnerabilità per attaccare un asset.
- **Rischio**: il livello di impatto sulle operazioni, gli asset dell'organizzazione, sugli individui, su altre organizzazioni o sulla reputazione, derivante dalla combinazione di una minaccia e della probabilità che questa si verifichi.
- **Security Controls**: misure di gestione e controlli tecnici prescritti per proteggere la confidenzialità, integrità e disponibilità di un sistema, dei suoi componenti, processi e dati.

1.2 Cyber Kill Chain

Definizione di Cyber Kill Chain

La **Cyber Kill Chain** è un modello che descrive le fasi di un attacco informatico, dalla fase di ricognizione fino a quella di azione.

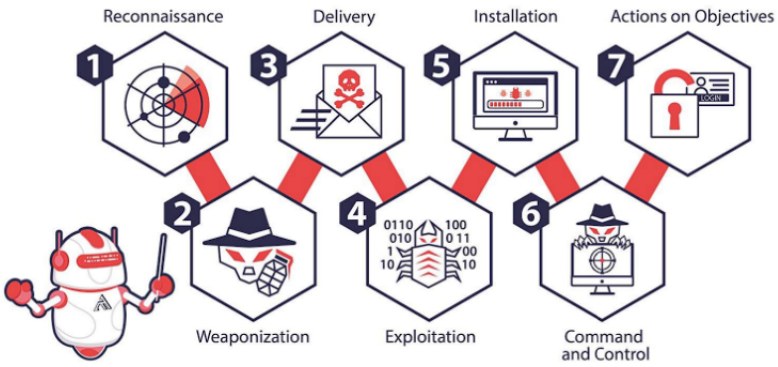

Figura 1.2.1: Cyber Kill Chain

1.2.1 Fasi della Cyber Kill Chain

1. **Reconnaissance**: fase in cui l'attaccante raccoglie informazioni sull'organizzazione, sui suoi asset e sulle vulnerabilità presenti. Questa fase può essere:

- **Passiva**: l'attaccante raccoglie informazioni da fonti pubbliche.
- **Attiva**: l'attaccante raccoglie informazioni tramite attività di scansione e sondaggio (`nmap`, *port scanning*).

2. **Weaponization**: fase in cui l'attaccante crea un payload malevolo e lo trasforma in un file eseguibile (*metasploit*, *air crack*).
3. **Delivery**: si seleziona in che modo trasportare l'exploit (*es. email, USB, social engineering*).
4. **Exploitation**: fase in cui il payload malevolo sfrutta una vulnerabilità per eseguire il codice malevolo (es. *buffer overflow*, *SQL injection*).
5. **Installation**: fase in cui si mantiene la persistenza nell'ambiente (*remote access trojan, powershell commands, DLL hijacking*). Si cerca anche di fare movimento laterale e spostarsi su altre macchine.
6. **Command and Control**: fase in cui si stabilisce un **canale di comando e controllo**, abbreviato in `C2`, in modo da manipolare la vittima, si apre quindi un canale di comunicazione a due vie tra l'attaccante e la vittima.
7. **Actions on Objectives**: fase in cui l'attaccante raggiunge i suoi obiettivi, come rubare dati o interrompere i servizi.

1.2.2 Trickbot

Trickbot

Un **Trickbot** è un trojan avanzato che si diffonde principalmente tramite email di phishing. Una volta scaricato inconsapevolmente dall'utente, Trickbot stabilisce un canale di comunicazione con un server di comando e controllo (`C2`), attraverso il quale l'attaccante può inviare comandi, distribuire Trickbot stesso o altri malware all'interno della rete compromessa.

1.3 `MITRE PREATT&CK` e `ATT&CK`

Definizione di MITRE ATT&CK

Il `MITRE ATT&CK` è un framework di tattiche e tecniche utilizzato per descrivere le fasi di un attacco informatico. Il framework è suddiviso in due parti: il `PREATT&CK` e l'`ATT&CK`.

È possibile mappare il `MITRE PREATT&CK` nelle fasi di reconnaissance e weaponization della *Cyber Kill Chain*, mentre l'`MITRE ATT&CK` corrisponde alle fasi di delivery, exploitation, installation, command and control, e actions on objectives.

1.3.1 MITRE PREATT&CK

Definizione di MITRE PREATT&CK

Il MITRE PREATT&CK raccoglie tutte le tattiche e le tecniche utilizzate nelle prime fasi di un attacco informatico. La struttura del framework è organizzata in colonne di tattiche, mentre le righe rappresentano le relative tecniche associate.

Ad esempio, la tattica *technical information gathering* rappresenta il processo mediante il quale un attaccante identifica informazioni critiche sul target, necessarie per pianificare efficacemente l'attacco.

Tra le tecniche di questa tattica troviamo **discover target logon/email address format**, che consiste nel determinare come sono strutturati i formati degli indirizzi email di una specifica organizzazione, ad esempio il dominio o il modello utilizzato.

1.3.2 MITRE ATT&CK

Definizione di MITRE ATT&CK

Il MITRE ATT&CK si concentra sulle tattiche e tecniche adottate durante le fasi operative di un attacco informatico.

MITRE ATT&CK matrix

La MITRE ATT&CK Matrix organizza le sue informazioni in colonne rappresentanti le tattiche e righe che descrivono le tecniche associate a tali tattiche. Le tattiche rappresentano ciò che un attaccante spera di ottenere, mentre le tecniche rappresentano come l'attaccante può raggiungere tali obiettivi.

Ad esempio, nell'ambito della tattica *Initial Access*, troviamo la tecnica di *phishing*, che consiste nel tentativo di ottenere informazioni sensibili o prendere il controllo di un sistema attraverso messaggi ingannevoli rivolti alla vittima. Se l'attacco è mirato, viene definito *spearphishing*, il quale a sua volta presenta delle sottotecniche. Una di queste è lo *spearphishing attachment*, in cui si invia un'email contenente un allegato malevolo progettato per ottenere informazioni sensibili o accesso al sistema. Gli allegati possono essere file eseguibili, PDF o documenti Office.

Per mitigare tali minacce, è possibile utilizzare strumenti come *network intrusion detection systems*, *email gateways* e antivirus, che generalmente sono in grado di rilevare allegati malevoli.

1.3.3 Tattiche e tecniche utilizzate da TrickBot

Analizziamo le tecniche dell'ATT&CK Matrix e come TrickBot le sfrutta nelle varie fasi della *cyber kill chain*:

- **Reconnaissance**: TrickBot utilizza tecniche come lo *spearphishing attachment* o lo *spearphishing link* per raccogliere informazioni e compromettere il bersaglio.

- **Execution**: Viene eseguito codice malevolo tramite tecniche come *scheduled task* o file *JavaScript* maligni.
- **Persistence**: Per mantenere l'accesso al sistema compromesso, TrickBot crea un servizio che si avvia automaticamente all'accensione della macchina.
- **Privilege Escalation**: L'attacco mira a ottenere privilegi maggiori sul sistema compromesso.
- **Defense Evasion**: TrickBot impiega diverse tecniche per eludere i sistemi di difesa, tra cui:
  - Offuscamento del codice;
  - Cifratura del malware;
  - Disattivazione di strumenti di sicurezza come Windows Defender.
- **Credential Access**: Include tecniche per la scoperta e il furto di credenziali.
- **Lateral Movement**: Movimento laterale all'interno della rete per compromettere ulteriori sistemi.
- **Collection**: Raccolta di dati sensibili dal sistema compromesso.
- **Command and Control**: Comunicazione tra il malware e il server di comando e controllo per ricevere istruzioni o inviare dati raccolti.
- **Exfiltration**: Estrazione dei dati sensibili verso server esterni controllati dall'attaccante.
- **Impact**: Tecniche per influenzare, sabotare o interrompere i sistemi compromessi.

1.3.4 Chi c'è dietro gli ultimi attacchi?

Dietro gli ultimi attacchi informatici troviamo tre principali tipologie di attori: cybercriminali, attori finanziati dallo Stato (*nation-state hackers*) e hacktivisti. Le origini principali di questi gruppi sono Russia e Cina, mentre i loro obiettivi principali includono gli Stati Uniti, seguiti dal Regno Unito. Le industrie più colpite sono i governi, i servizi finanziari e il settore tecnologico.

Cybercriminali

I cybercriminali sono mossi dall'interesse di ottenere profitti illegali. Tra gli attacchi tipici che eseguono troviamo:

- **Ransomware**: Blocco dei dati delle vittime in cambio di un riscatto.
- **Infostealers** (es. *Raccoon Stealer*): Software progettati per rubare informazioni sensibili.

- **Proxyjacking** (es. *Avrecon*): Una tipologia di attacco che sfrutta piattaforme di *proxyware*, le quali consentono agli utenti di guadagnare condividendo la propria connessione Internet con altri. Gli attaccanti monetizzano la larghezza di banda delle vittime sfruttando queste piattaforme.

Nation-State Hackers

Gli attori finanziati dallo Stato (*nation-state hackers*) sono interessati principalmente a:

- **Intelligence**: Raccolta di informazioni riservate.
- **Sabotaggio e Spionaggio**: Danni a infrastrutture critiche o spionaggio tecnologico e industriale.

Tra gli attacchi tipici eseguiti troviamo:

- Attacchi a infrastrutture critiche.
- Wipers: Malware progettati per distruggere i dati.
- Attacchi *DDoS*: Interruzione dei servizi tramite sovraccarico di traffico.

Hacktivisti

Gli hacktivisti sono motivati da visioni politiche, credi religiosi/sociali o ideologie terroristiche. Tra i principali attacchi da loro eseguiti troviamo:

- *DDoS*.
- Furti di dati (*data breaches*) o pubblicazioni di dati (*data leaks*).
- *Data wipers*.

Tra i gruppi più noti di hacktivisti ci sono *Anonymous*, *GhostSec* e *KillNet*.

1.3.5 Come operano gli attori?

Gli attori informatici utilizzano diverse tecniche e strumenti avanzati per raggiungere i loro obiettivi:

- **Attack-as-a-Service**: Gli attaccanti offrono servizi di attacco in cambio di un compenso (*fee*), permettendo a chiunque di "affittare" un attacco come servizio.
- **Compromissione dei dispositivi di rete**: L'accesso iniziale viene spesso ottenuto compromettendo dispositivi di rete. Strumenti come *Shodan*, *Censys* e *Kamerka* vengono utilizzati per individuare dispositivi esposti su Internet, come router o videocamere IP, spesso compromessi tramite credenziali di default o deboli.
- **Strumenti di offensive security**: Strumenti come *Metasploit* vengono utilizzati per condurre attacchi mirati.

- **Living Off The Land Binaries (LOLBins)**: Gli attaccanti utilizzano elementi di sistema legittimi, come processi nativi di Windows, per mascherare malware ed evitare il rilevamento.

Capitolo 2

Tipologie di Malware

Malware

Il **malware** (*malicuous software*) è un software o firmware che esegue un processo non autorizzato che porta ad avere un impatto su **confidenzialità**, **integrità** o **disponibilità** di un sistema.

I sistemi vengono infettati dai malware attraverso diverse modalità:

- **Accesso diretto al sistema**: disco infetto, chiavetta USB, ecc.
- **Ingegneria sociale**
- **Phishing**: spear phishing, whale phishing, ecc.
- **Siti web infetti**.

2.1 Virus

Virus

Il **virus** è in grado di replicare se stesso e ha bisogno di un'azione umana per essere eseguito. Può infastidire gli utenti infettati con modifiche alle loro macchine e può essere trovato dagli antivirus.

I virus possono essere classificati in diverse categorie:

- **Macro virus**: si diffondono attraverso i documenti.
- **Polymorphic virus**: cambiano la loro firma per evitare di essere rilevati dagli antivirus.
- **Companion virus**: si mascherano da file legittimi presenti nel sistema.

2.2 Worm

Worm

I **worm**, simili ai virus, ma non infettano e non hanno bisogno di un'azione umana per essere eseguiti. Si diffondono attraverso la rete con movimento laterale, sono solitamente più pericolosi di un virus e spesso attaccano i server sfruttando difetti di configurazione.

2.3 Key Loggers

Key Loggers

Dal nome si intuisce che sono un malware che registra le battiture della tastiera.

È quasi sempre presente un'operazione di data exfiltration (*upload `FTP`, emailing `LOGS`*), anche se comunque spesso i dati vengono salvati sempre localmente. Vengono tipicamente utilizzati per rubare informazioni sensibili come password, codici di accesso, ecc.

2.4 Trojans

Trojans

Spesso rappresentano se stessi come un software utile, creano una backdoor da dove gli hacker controllano la macchina. Spesso scaricati da siti non ufficiali, vengono usati per rubare informazioni personali, file e trasformare la macchina in uno zombie.

2.4.1 `RATs` (Remote Access Trojan)

`RATs`

Sottocategoria dei trojan, progettati per permettere a un attaccante di controllare da remoto la macchina infetta. Essenzialmente imposta un `C2` (*command and control channel*) con il server dell'attaccante, da dove vengono mandati i comandi al `RAT` e dove i dati generati vengono spediti. Spesso hanno comandi predefiniti e metodi per nascondere il traffico del `C2`.

2.5 Rootkits

Rootkits

Si installano tra il sistema operativo e l'hardware del computer, usati per assicurare agli hacker il controllo di una macchina infetta e per mascherare la presenza di altri malware nel sistema. Alcuni sono impossibili da rimuovere a tale livello che il drive deve essere distrutto.

2.6 Droppers/Downloaders

Droppers/Downloaders

Dal nome, i droppers "droppano" un file embedded spesso contenuto in documenti Word o Excel. Da soli non sono pericolosi, ma lo è ciò che scaricano.

2.7 Bots

Bots

Una volta infettato il sistema, quest'ultimo diventa parte di una botnet controllata dal botmaster. Spesso usata per `DDoS` attack o per distribuire malicious spam. Botnet note sono Mirai e Satori.

2.8 Cripto Miners

Cripto Miners

Minano criptovalute con la macchina della vittima che vengono poi spedite al wallet dell'attaccante.

2.9 Ransomware

Ransomware

Cifra tutti i file sul sistema quando viene eseguito, mostrando un messaggio alla vittima per pagare e ottenere i file decifrati. Tipicamente accetta pagamenti in Bitcoin. La chiave usata per cifrare i file è simmetrica ed è a sua volta criptata con una chiave asimmetrica in modo da rendere l'attaccante l'unico a possedere la chiave privata.

Un elemento importante quando si parla di ransomware sono i **kill switches**, implementati dagli autori dell'attacco per evitare di infettare la propria infrastruttura.

A volte vengono lasciati nel codice per errore e riconosciuti dai ricercatori che li usano per fermare il malware.

Kill Switch

Un kill switch è un meccanismo per disabilitare un malware o un virus. Può essere usato per fermare la propagazione di un malware o per disabilitare le funzionalità di un virus.

2.9.1 Cyber Kill Chain di un Ransomware

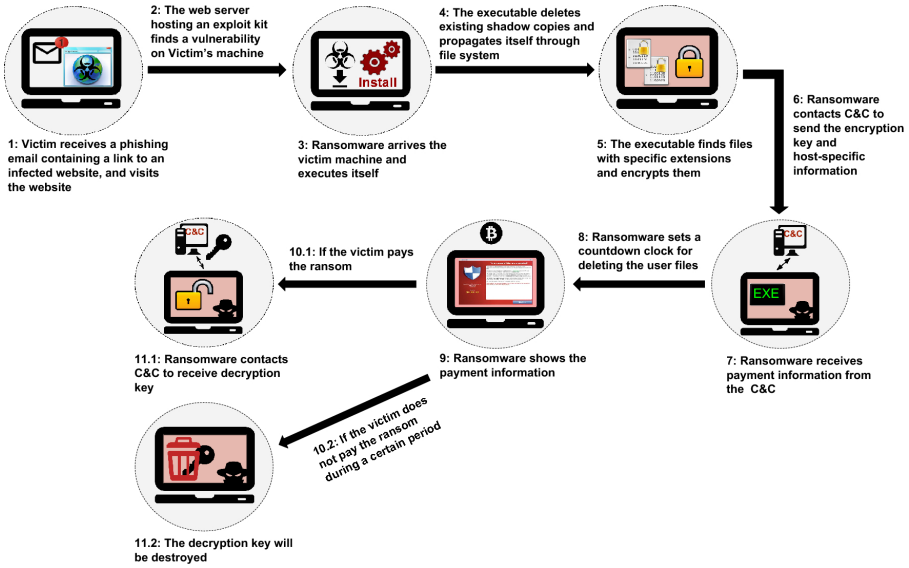

Figura 2.9.1: Cyber Kill Chain di un Ransomware

I cybercriminali utilizzano diverse strategie per massimizzare il profitto e aumentare la pressione sulle vittime di ransomware. Nel corso del tempo, gli attacchi sono evoluti da una semplice cifratura dei file a metodi sempre più sofisticati, creando i concetti di **doppia, tripla e quadrupla estorsione**.

1. Estorsione Semplice (Ransomware Tradizionale)

Il ransomware classico si limita a **crittografare i file della vittima** e a richiedere un riscatto per ottenere la chiave di decrittazione.

- **Obiettivo**: Impedire l'accesso ai dati e forzare il pagamento del riscatto.
- **Pressione sulla vittima**: Impossibilità di accedere ai propri file fino al pagamento.
- **Esempio**: WannaCry.

2. Doppia Estorsione

In questa variante, gli attaccanti **esfiltrano i dati** prima di cifrarli e minacciano di **pubblicarli o venderli** se il riscatto non viene pagato.

- **Obiettivo**: Aumentare la pressione sulla vittima con il rischio di esposizione di dati sensibili.
- **Pressione sulla vittima**: Il danno reputazionale e legale diventa un'arma in più per i criminali.
- **Esempi**: Maze, REvil.

3. Tripla Estorsione

Oltre alla doppia estorsione, gli attaccanti **colpiscono anche terze parti collegate alla vittima**, come clienti, fornitori e partner, minacciando di divulgare informazioni su di loro.

- **Obiettivo**: Moltiplicare le potenziali vittime che possono spingere l'organizzazione principale a pagare.
- **Pressione sulla vittima**: Coinvolgere altre aziende o persone, aumentando il rischio di sanzioni e perdita di fiducia.
- **Esempio**: Ragnar Locker, Clop.

4. Quadrupla Estorsione

Aggiunge un ulteriore livello di coercizione: **gli attaccanti avviano attacchi DDoS** contro i server della vittima, interrompendo i servizi online.

- **Obiettivo**: Creare un'interruzione operativa completa per forzare il pagamento.
- **Pressione sulla vittima**: L'azienda potrebbe perdere clienti e subire gravi perdite economiche a causa del downtime.
- **Esempio**: Avaddon.

2.9.2 Nuove Tendenze e Target dei Ransomware

I nuovi target dei ransomware includono **infrastrutture cloud**, le quali vengono compromesse attraverso tecniche di **phishing** per ottenere l'accesso iniziale. Inoltre, gli sviluppatori di ransomware stanno adottando nuove tecnologie per eludere i sistemi di sicurezza:

- **Linguaggi di programmazione emergenti**: mentre molti ransomware sono stati storicamente scritti in `C#`, `C++` o `Python`, oggi si sta assistendo a una migrazione verso linguaggi come `Rust` e `Go`, noti per la loro efficienza e difficoltà di analisi da parte degli antivirus.
- **Intermittent Encryption**: tecnica che cripta solo parte dei file per accelerare il processo e ridurre il rischio di rilevamento da parte delle difese automatiche.

- **Ransomware-as-a-Service (RaaS)**: modello di business in cui gruppi di attaccanti forniscono ransomware a affiliati meno esperti, ampliando il numero di attacchi globali.

2.9.3 Come Proteggersi dai Ransomware

Per ridurre il rischio di infezione da ransomware, è fondamentale adottare una strategia di sicurezza informatica basata su prevenzione e buone pratiche:

- **Non cliccare su link non verificati** o scaricare software da siti non attendibili (*untrusted*).
- **Mantenere il sistema aggiornato**: installare le patch di sicurezza per ridurre le vulnerabilità.
- **Eseguire backup regolari**: conservare copie dei dati in posizioni sicure (offline o in cloud con versioning) per garantire il ripristino in caso di attacco.
- **Utilizzare autenticazione multi-fattore (MFA)** per proteggere gli account critici.
- **Limitare i privilegi utente**: ridurre i permessi amministrativi sui sistemi per limitare l'impatto di un'infezione.

2.9.4 Come Rispondere a un Attacco Ransomware

Se un sistema è stato compromesso da un ransomware, è importante agire rapidamente per limitare i danni:

- **Disconnettersi immediatamente dalla rete** per prevenire il movimento laterale del malware e l'infezione di altri dispositivi.
- **Utilizzare strumenti di decrittazione** (*ransomware decryption tools*) se disponibili per tentare di recuperare i file senza pagare il riscatto.
- **Ripristinare i file da backup sicuri** per riportare il sistema a uno stato precedente all'infezione.
- **Contattare le autorità competenti** (es. CERT, Polizia Postale) per segnalare l'attacco e ricevere assistenza.

2.10 Principali Tipologie di Malware per i Diversi Sistemi Operativi

2.10.1 Windows

I malware più diffusi su sistemi Windows includono:

- **Botnet**: reti di dispositivi compromessi utilizzate per attacchi DDoS o spam.
- **Infostealer**: malware che ruba informazioni sensibili come credenziali e dati bancari.

- `RATs` (*Remote Access Trojans*): trojan che permettono l'accesso remoto al sistema compromesso.
- **Ransomware**: malware che cifra i file e chiede un riscatto per la decrittazione.

2.10.2 Linux

I malware più comuni su Linux includono:

- **Miner**: software malevolo che sfrutta le risorse della macchina per minare criptovalute.
- `DDoS`: strumenti utilizzati per generare attacchi di tipo Distributed Denial of Service.
- **Tsunami**: backdoor che permette l'accesso remoto ai sistemi Linux compromessi.

2.10.3 MacOS

Anche MacOS non è esente da malware, tra i più diffusi troviamo:

- **Nukesped** (*trojan*): un malware che consente agli attaccanti di controllare da remoto il dispositivo.
- **VSearch** (*browser hijacking*): software malevolo che modifica le impostazioni del browser per reindirizzare il traffico a siti indesiderati.

2.11 Malware: Prevenzione, Rilevamento ed Eradicazione

2.11.1 Prevenzione della Consegna del Malware

Per ridurre il rischio che il malware venga consegnato ai sistemi, è possibile adottare le seguenti misure di sicurezza:

- **Mail filtering**: filtra le email in entrata per bloccare allegati malevoli e link dannosi.
- **Proxy di intercettazione**: previene il download di contenuti pericolosi e blocca connessioni sospette.
- **Internet security gateways**: ispeziona i protocolli di rete per rilevare eventuali minacce.
- **Safe browsing lists**: blocca l'accesso a siti web noti per distribuire malware.

2.11.2 Prevenzione dell'Esecuzione del Malware

Per impedire che il malware venga eseguito una volta che è stato consegnato al sistema, è possibile adottare le seguenti strategie:

- **Gestione centralizzata dei dispositivi**: consente di eseguire solo applicazioni fidate e approvate dall'azienda.
- **Antivirus e antimalware**: installare e mantenere aggiornati i software di protezione per rilevare e bloccare minacce.

- **Disattivazione dell'autorun**: evitare che dispositivi esterni montati automaticamente possano eseguire codice malevolo.
- **Limitazione o disabilitazione delle macro**: ridurre il rischio di attacchi che sfruttano macro in documenti Office.
- **Limitazione degli ambienti di scripting**: controllare o disabilitare l'uso di strumenti come PowerShell.
- **Utilizzo dell'ultima versione del sistema operativo**: mantenere aggiornati `OS` e applicazioni per beneficiare delle ultime patch di sicurezza.
- **Configurazione di firewall di rete**: disabilitare connessioni in entrata non necessarie per limitare i punti di accesso.

2.11.3 Prevenzione dell'Espansione del Malware

Nel caso in cui un malware riesca a infiltrarsi in un sistema, è fondamentale prevenire la sua diffusione adottando le seguenti contromisure:

- **Utilizzo dell'autenticazione multi-fattore (`MFA`)**: impedisce che credenziali rubate vengano riutilizzate dal malware per propagarsi.
- **Isolamento di piattaforme obsolete**: segregare vecchi sistemi operativi e applicazioni non più aggiornati per limitare il rischio di infezioni.
- **Rimozione dei permessi non necessari**: ridurre i privilegi utente per limitare l'impatto dell'attacco.

2.11.4 Strategie di Backup

I backup sono una difesa fondamentale contro gli attacchi informatici, incluso il ransomware. Per garantire la sicurezza dei dati, si consiglia di:

- **Mantenere i backup in posizioni diverse**: conservare copie dei dati sia offline che in cloud, evitando che siano accessibili dal sistema compromesso.
- **Evitare la connessione permanente alla rete**: se un backup è memorizzato su un dispositivo fisico, assicurarsi che non sia sempre connesso alla rete.
- **Scansionare i backup per malware**: prima di effettuare il ripristino, verificare che i backup non siano stati compromessi da minacce informatiche.

2.11.5 Risposta a un'Infezione da Malware

Se un malware ha già infettato un'organizzazione, è necessario agire tempestivamente per limitare i danni e ripristinare la sicurezza:

- **Disconnettere immediatamente i dispositivi infetti** per prevenire il movimento laterale della minaccia.
- **Disattivare il `Wi-Fi` e qualsiasi connessione di rete** per isolare il malware.

- **Effettuare un wipe dei dispositivi compromessi**: formattare e reinstallare il sistema operativo per eliminare completamente la minaccia.
- **Collegare i dispositivi a una rete pulita** prima di installare software di sicurezza e aggiornamenti.
- **Monitorare il traffico di rete** per rilevare eventuali segni di attività sospetta.
- **Utilizzare antivirus e strumenti di sicurezza** per individuare e rimuovere eventuali residui di malware.

Capitolo 3

Ingegneria Sociale

3.1 Introduzione

Ingegneria sociale

L'**ingegneria sociale** (dall'inglese *social engineering*) è la tecnica di attacco che sfrutta la *manipolazione psicologica* delle persone per indurle a rivelare informazioni o compiere azioni dannose.

Si tratta di uno dei rischi più insidiosi in ambito *cybersecurity* poiché, nonostante gli investimenti in strumenti di difesa (firewall, sistemi di `access control`, antivirus), il cosiddetto *fattore umano* spesso rappresenta l'anello debole della catena di sicurezza. Difatti, un'azienda può spendere milioni in tali sistemi, ma basta *corrompere o ingannare* una sola persona all'interno per vanificare investimenti così ingenti.

3.2 Il fattore umano come anello debole

Un singolo dipendente, se *ingannato* o *corrotto*, può permettere a un criminale informatico di penetrare le difese di un'intera infrastruttura. Diventa quindi fondamentale *formare il personale* a riconoscere e prevenire gli attacchi di ingegneria sociale, oltre a implementare misure di sicurezza tecniche e fisiche.

3.3 Ciclo di vita dell'attacco

Un classico attacco di *social engineering* si articola in quattro fasi principali:

1. **Raccolta di informazioni** (*Information Gathering*): l'attaccante raccoglie dati (ad esempio *email, organizzazione interna, nomi di dipendenti, ruoli, recapiti*) da fonti pubbliche, social network e ricerche `OSINT` (*Open Source Intelligence*).

2. **Stabilire la relazione** (*Establish Relationships*): tramite email, telefonate o altri mezzi, l'attaccante finge di essere un ente affidabile (banca, ufficio `HR`, ecc.) per guadagnare la fiducia della vittima.
3. **Exploitation**: sfruttando la fiducia ottenuta, l'attaccante induce la vittima a rivelare credenziali o a compiere azioni (*cliccare su link malevoli, scaricare allegati dannosi, ecc.*).
4. **Esecuzione** (*Execution*): si passa al piano operativo, come il furto di dati, l'installazione di malware o l'accesso non autorizzato a sistemi interni.

3.4 Principali tipologie di attacco

Le strategie di ingegneria sociale spaziano dalle tecniche di **phishing** a quelle di raccolta fisica di informazioni (dumpster diving, shoulder surfing), fino a campagne di *tailgating* o *vishing*.

3.4.1 Phishing, Spear Phishing e Whaling

`Phishing`

Phishing: attacco che mira a ottenere informazioni sensibili (*credenziali, dettagli bancari, numeri di carta*) fingendosi un'entità affidabile. Generalmente avviene via email o `SMS`, invitando la vittima a cliccare un link che porta a un sito falso, simile a quello legittimo.

- **Spear Phishing**: versione *mirata* del phishing, diretta a un bersaglio specifico (ad esempio un singolo dipendente di un'azienda), in cui i messaggi sono personalizzati e dunque più credibili.
- **Whaling**: forma di *spear phishing* che prende di mira dirigenti o funzionari di alto livello (*"whales"*), simulando spesso comunicazioni tra top manager o enti prestigiosi.

3.4.2 Vishing e SMiShing

`Vishing & SMiShing`

Vishing (*Voice Phishing*): l'attaccante si spaccia per un ente rispettabile mediante chiamate telefoniche o messaggistica vocale, inducendo la vittima a rivelare informazioni confidenziali.
SMiShing (*SMS Phishing*): utilizza gli `SMS` come canale d'attacco per ingannare la vittima e ottenere dati personali (numero di carta di credito, codice fiscale, ecc.).

3.4.3 Dumpster Diving e Shoulder Surfing

Vishing & SMiShing

Dumpster Diving L'attaccante fruga tra i rifiuti (*documenti, bollette, ricevute*) in cerca di informazioni sensibili, senza ricorrere a tecnologie particolari.
Shoulder Surfing L'*"sbirciata da dietro le spalle"* per carpire credenziali mentre la vittima digita, o per esaminare documenti su scrivanie e monitor. L'attaccante può camuffarsi da addetto alle pulizie per aggirare i controlli.

3.4.4 Tailgating

Tailgating

Attacco in cui l'aggressore segue da vicino un dipendente autorizzato, approfittando della sua buona fede (porta tenuta aperta, badge non controllato) per accedere a un'area protetta dove altrimenti non potrebbe entrare.

3.5 Phishing: diffusione e fattori di persuasione

Il phishing è tra gli attacchi di ingegneria sociale più diffusi a livello mondiale.

Molti attacchi sfruttano tecniche di *persuasione* per convincere la vittima:

- **Autorità**: fingendosi un ente autorevole (uffici `HR`, `CEO`, banca, ecc.).
- **Urgenza o scarsità**: "*affrettati, restano pochi posti disponibili*".
- **Liking & social proof**: mostrare contatti o situazioni familiari per sembrare affidabili.

Uno studio ha dimostrato che:

- Il 33% degli utenti era più suscettibile a *messaggi urgenti*.
- Il 21% rispondeva a *autorità*.
- Il 15% cadeva nell'attacco anche senza alcuna tecnica di persuasione.

3.6 Phishing come servizio

Phishing-as-a-Service

Oggi esistono servizi "*chiavi in mano*" che offrono campagne di phishing a pagamento, fornendo siti cloni e template di email. Il *cybercriminale* acquista il pacchetto, imposta la campagna, e ottiene in automatico i dati rubati, senza particolari competenze tecniche.

3.7 Riconoscimento e *Exploitation* Technique

Nel ciclo di vita di un attacco rientra la fase di **raccolta di informazioni** (*information gathering*). Ecco alcune tecniche di analisi preliminare:

3.7.1 `DNS` Analysis

Si analizza il traffico `DNS` alla ricerca di nomi di dominio, server mail o servizi esterni di terze parti. Tra i record `DNS` più utilizzati troviamo:

- `SOA` State of Authority
- `NS` Name Server
- `A` Indirizzo `IPv4`
- `MX` Mail Exchange
- `CNAME` Canonical Name
- `TXT` Testo generico (spesso usato per `SPF`, `DKIM`, ecc.)

Strumenti come `dnsrecon` permettono di automatizzare la raccolta di questi dati e identificare potenziali punti deboli.

3.7.2 `OSINT` Analysis

Si cercano informazioni su dipendenti (*username, indirizzi email, numeri di telefono, profili social*) e sull'organizzazione (*posizioni, sedi, sottodomini, credenziali eventualmente trapelate*) esclusivamente da fonti pubbliche, in modo del tutto legale.

3.7.3 Active Scanning

L'attaccante investiga l'infrastruttura di rete inviando traffico (sondando porte aperte, servizi `TCP/UDP`, ecc.) per:

- **Scanning `IP` blocks**: le aziende spesso hanno range di `IP` pubblici. Analizzandoli, si scoprono host attivi e potenziali servizi vulnerabili.
- **Vulnerability scanning**: si confronta la configurazione di host/applicazioni con exploit noti, individuando punti deboli da sfruttare.

3.8 Riconoscere e prevenire gli attacchi di phishing

3.8.1 Indicatori di un Phishing

Tra i segnali comuni per identificare email o siti di phishing:

- **Domini irregolari**: *typosquatting* (nomi con lettere invertite, sottodomini sospetti).
- **Mancanza di `HTTPS`**: siti non protetti da certificato o con certificato scaduto.

- **Messaggi inaspettati**: richieste di dati confidenziali con urgenza esagerata.
- **Link sospetti**: puntare il cursore sul link per verificare la destinazione effettiva prima di cliccare.

3.8.2 Prevenzione

- **Formazione continua**: addestrare i dipendenti a riconoscere email sospette e verificare `URL`.
- **Filtri anti-phishing**: soluzioni software e hardware che analizzano allegati e link in tempo reale.
- **Autenticazione forte**: abilitare l'autenticazione a più fattori (`MFA`) per mitigare il rischio di compromissione di credenziali.
- **Procedure di segnalazione**: predisporre canali interni (es. `help desk` o `CSIRT`) per segnalare mail o comportamenti sospetti.

Capitolo 4

Attacchi a Infrastrutture Critiche, Sistemi di Controllo Industriale e Cyberwar

4.1 Definizione di infrastrutture critiche

Le **infrastrutture critiche** sono strutture, sistemi informativi e organizzazioni di importanza strategica per il funzionamento di una nazione, da cui dipende la vita quotidiana dei cittadini. Esempi tipici includono:

- **Dighe**
- **Settore energetico** (*produzione, distribuzione*)
- **Healthcare** (*ospedali, sistemi sanitari*)
- **Servizi finanziari** (*banche, borse valori*)

Oltre ai *servizi essenziali*, ricadono nelle infrastrutture critiche anche quelle organizzazioni che, se compromesse, potrebbero provocare danni significativi ai cittadini. La compromissione di tali strutture può comportare:

- Impatto negativo su *disponibilità* e *integrità* dei servizi essenziali.
- Conseguenze rilevanti per la sicurezza e la difesa nazionale, nonché il funzionamento dello Stato.

4.2 Sistemi di Controllo Industriale (`ICS`)

Molte infrastrutture critiche sono gestite e monitorate da `ICS` (*Industrial Control Systems*), come i `PLC`, che verificano ad esempio l'andamento dei processi industriali, rilevando possibili anomalie. Questi sistemi, spesso datati o scarsamente aggiornati, risultano particolarmente vulnerabili ad attacchi informatici.

Dispositivi vulnerabili in ambito ICS

Esempi di componenti critici e vulnerabili:

- `NAS` (Network-Attached Storage)
- `IP Camera`
- `PLC` (Programmable Logic Controller)
- `UPS` (Uninterruptible Power Supply)
- **Monitor di pazienti** in ambito sanitario

4.2.1 `ICS` Cyber Kill Chain

Gli attacchi a un `ICS` possono seguire una **kill chain** in più fasi:

1. **Raccolta di informazioni**: l'attaccante ricerca dettagli sull'infrastruttura target (topologia di rete, dispositivi `ICS` utilizzati, fornitori).
2. **Sviluppo dell'attacco**: solitamente multi-stage, sfrutta più vulnerabilità in sequenza per arrivare ai sistemi `ICS`.
3. **Delivery & Exploitation**: consegna del malware (o exploit) e compromissione del `PLC` o del software di controllo.
4. **Fase conclusiva**: sabotaggio, furto di dati o manipolazione del processo industriale.

4.3 Cyberwar

Si parla di **cyberwar** quando una nazione prende di mira le infrastrutture critiche di un'altra nazione con finalità di sabotaggio o spionaggio. Tipologie comuni di attacchi in uno scenario di guerra informatica includono:

- **Spionaggio** (*esfiltrazione di dati riservati*)
- **Sabotaggio** (*manipolazione o danneggiamento di impianti strategici*)
- **Attacchi a fornitori di energia** (*blackout programmati*)

4.4 Stuxnet: la prima cyber-weapon

Stuxnet

Stuxnet è considerata la prima vera arma cibernetica mirata a sabotare specifici `PLC` nelle centrifughe di una centrale iraniana per l'arricchimento dell'uranio. Secondo alcune fonti, fu sviluppata da `NSA, CIA` e dall'intelligence israeliana per rallentare il programma nucleare dell'Iran.

4.4.1 Funzionamento

- **Reconnaissance**: per ottenere informazioni, gli attaccanti osservarono *video propagandistici* iraniani notando, in sottofondo, monitor di controllo delle centrifughe.
- **Delivery**: la propagazione iniziale avvenne tramite `USB` infette. Successivamente, Stuxnet si diffuse in rete sfruttando credenziali hardcoded e vulnerabilità di `Windows`.
- **Installazione e `C2`**: una volta su macchine target (`PLC` Siemens), stabiliva un canale `C2` (*command & control*) per ricevere istruzioni o aggiornamenti da server contraffatti.
- **Sabotaggio**: manipolava i controlli dei `PLC`, danneggiando le centrifughe mentre forniva dati falsificati ai sistemi di monitoraggio.

4.5 NotPetya

`NotPetya`

NotPetya è un *ransomware* (in realtà un *wiper* mascherato) che ha colpito numerose organizzazioni, tra cui la Banca Centrale ucraina e la compagnia di shipping *Maersk*.

- **Delivery**: diffuso tramite l'aggiornamento manomesso di un software fiscale ucraino.
- **Propagazione**: usava exploit `EternalBlue` ed `EternalRomance`.
- **Effetti**: cifrava file e dischi con `AES128`, richiedendo un riscatto di 300 dollari in Bitcoin.

Capitolo 5

User Authentication

5.1 Introduzione

Processo di Autenticazione

Il processo di autenticazione determina l'identità di un utente sulla base di qualcosa che:

- **Conosce** (*password, PIN, risposte segrete*)
- **Possiede** (*smart card, token fisico*)
- **È** (*caratteristiche biometriche*)

5.2 Autenticazione con password

Il metodo più diffuso è l'autenticazione via *username* e *password*, dove il sistema confronta le credenziali inserite con quelle archiviate. Tuttavia, questo approccio presenta diversi problemi di sicurezza e usabilità:

- **Password overload**: le persone hanno molti account e scelgono password semplici o comuni per ricordarle facilmente.
- **Password reuse**: riutilizzo della stessa password su più servizi; se una password viene compromessa, l'attaccante può riusarla altrove.

5.2.1 Tipi di attacchi alle password

Tipologie di attacchi

- **Offline attacks** L'attaccante non ha accesso al sistema di autenticazione online, ma possiede (ad esempio) gli hash delle password.
- **Non-technical attacks** *Social engineering*, come phishing, dumpster-diving o shoulder surfing.
- **Active online attacks** L'attaccante ha accesso al sistema di autenticazione e prova iterativamente password.
- **Passive online attacks** L'attaccante intercetta comunicazioni di rete (es. *sniffing*) per catturare password in transito.

Tra gli attacchi comuni che troviamo alle password ci sono *brute force*, *social engineering*, *keylogger*, *shoulder surfing*.

5.3 Collocazione delle password nei sistemi operativi

Nei sistemi operativi, le password (o i loro hash) sono conservate in file specifici:

- **Windows**: Nel database `SAM` (*Security Account Manager*), tipicamente in `System32\config`.
- **Linux**: `/etc/passwd` e `/etc/shadow`.

5.3.1 Hash delle password in Windows

- `LM` hash:
  - Massimo 14 caratteri per la password.
  - Trasforma le lettere in maiuscolo, fa padding se < 14 caratteri, poi divide in due blocchi da 7, hashati separatamente.
- `NTLM` hash:
  - Supporta password fino a 256 caratteri.
  - L'hash viene calcolato sull'intera stringa, senza limiti di maiuscolo/minuscolo come in `LM`.

5.4 Attacchi comuni alle password

5.4.1 Brute force

Brute Force

Si provano tutte le combinazioni possibili di caratteri fino a trovare quella corretta. Lo spazio delle password è $|A|^n$ (dove n è la lunghezza e A l'alfabeto). Ad esempio, con 8 caratteri e un alfabeto di 96 simboli, si ottengono 96^8 (7.2 quadrilioni) combinazioni.

5.4.2 Dictionary attack

Dictionary Attack

Simile al *brute force*, ma si utilizzano liste predefinite di password comuni (*"dizionari"*). Trova la password solo se essa è inclusa in tali elenchi.

5.4.3 Hybrid attack

Hybrid Attack

Combina l'approccio a dizionario con variazioni su lettere (maiuscole, numeri, simboli) per aumentare le probabilità di successo senza provare l'intero spazio del brute force.

5.4.4 Rainbow tables

Rainbow Tables

Si tratta di tabelle precompute di *coppie password-hash*. Consentono di invertire l'hash velocemente senza *bruteforce* completo. L'uso di *salt* (un valore casuale unito alla password prima dell'hash) mitiga questo attacco.

5.4.5 Pass-the-hash

Pass-the-Hash

L'attaccante fornisce direttamente l'hash della password (ottenuto altrove) a un server, autenticandosi come la vittima senza dover decifrare la password.

5.4.6 Online dictionary attack

Online Dictionary Attack

Si tenta di indovinare la password dall'esterno, provando credenziali comuni su un'applicazione web o un servizio. Rischio di lockout o meccanismi di rate-limiting.

5.4.7 Credential stuffing

Credential Stuffing

Si dispone di credenziali rubate da un servizio, e si prova la medesima combinazione *username-password* su altri siti (sfruttando il *password reuse*).

5.4.8 Password spraying

Password Spraying

L'opposto del *brute force*: si prova una singola password molto comune (*es. "password123"*) su molti account diversi, sperando che qualcuno la utilizzi.

5.4.9 Keylogger

Keylogger

Un piccolo software (o dispositivo hardware) che registra tutti i tasti premuti dall'utente sulla tastiera.

5.4.10 Social engineering

Social Engineering

Include *phishing*, *shoulder surfing*, *dumpster-diving*. Queste tecniche di ingegneria sociale mirano a ingannare l'utente per ottenere direttamente le sue credenziali.

5.5 Valutare la robustezza di una password

5.5.1 Entropia e password strength

La *password strength* si valuta spesso con l'entropia:

$$\text{Entropia} = \log_2(|A|^n) = n \times \log_2(|A|)$$

Dove $|A|$ è la dimensione dell'alfabeto (numero di simboli diversi) e n la lunghezza. Un valore superiore a 60 bit è considerato "forte".

5.5.2 `Zxcvbn`

Una libreria che stima la robustezza di una password considerando la lunghezza, la presenza di maiuscole/minuscole, caratteri speciali, sequenze comuni e corrispondenze con dizionari.

5.6 Contromisure

- **Salting**: aggiungere un valore casuale (*salt*) prima di hasharla, per rendere inefficaci le rainbow tables.
- **Access control al file delle password**: solo utenti autorizzati possono accedere ai file in cui risiedono gli hash.
- **Lockout mechanism**: bloccare l'account dopo un certo numero di tentativi falliti.
- **Throttling**: introdurre ritardi tra un tentativo di login e l'altro.
- **Security monitoring**: rilevare pattern di login anomali, tentativi in orari sospetti, e avvisare l'utente.
- **Password blacklisting**: non consentire password banali (tipo *"123456"* o *"password"*).
- **Criptare le comunicazioni**: usare `SSL/TLS` per evitare che password in chiaro vengano intercettate (*passive online attack*).

Capitolo 6

Protocolli Single Sign-On

6.1 Introduzione

Identità Digitale

La **gestione dell'identità digitale** riguarda la rappresentazione digitale delle informazioni personali di un utente, inclusi dati identificativi (*nome, cognome, ID utente, numeri identificativi*) e credenziali di accesso. Un sistema di `Identity Management` centralizza la gestione delle identità digitali e garantisce l'accesso sicuro alle risorse e ai servizi.

6.1.1 Attori principali

- **Soggetto o Utente**: l'utente che si autentica al servizio.
- **Asserting Party o Identity Provider**: entità del sistema che crea asserzioni su un soggetto.
- **Relying Party o Service Provider**: entità del sistema che consuma le asserzioni create dall'`IdP`.

L'utente di autentica una sola volta e poi accede a tutte le risorse che è autorizzato ad utilizzare. L'autenticazione delle risorse è gestito dall'`IdP` in maniera trasparente all'utente, che non deve inserire nuovamente le credenziali.

6.1.2 Federated Identity

Se si parla di **Federated Identity**, ci si riferisce alla pratica in cui diverse organizzazioni stabiliscono un identificatore comune per riferirsi a un utente. Questo facilita la condivisione dell'identità tra varie organizzazioni e migliora l'efficacia dell'`SSO`.

Con un sistema di Federated Identity:

- Un utente che si autentica presso un membro della federazione ottiene accesso a tutti i membri senza dover inserire nuovamente le credenziali.
- Si riducono i costi di manutenzione e gestione delle identità.

6.2 SAML - Security Assertion Markup Language

SAML - Security Assertion Markup Language

Il protocollo SAML abilita l'SSO e la federazione dell'identità attraverso la generazione di asserzioni firmate.

6.2.1 Processo di Autenticazione SAML

Il meccanismo di autenticazione tramite SAML si articola nei seguenti passaggi:

1. L'utente invia una richiesta di accesso a un **Service Provider**.
2. Il **Service Provider** inoltra una richiesta di autenticazione all'**Identity Provider**.
3. L'utente interagisce con l'**Identity Provider** per effettuare il login, fornendo le proprie credenziali.
4. L'**Identity Provider**, dopo aver verificato le credenziali, genera e firma digitalmente un **assertion**, che viene poi inviato al **Service Provider**.
5. Il **Service Provider** valida l'**assertion** e, a seguito di una corretta autenticazione, invia un cookie all'utente per gestire la sessione.

Una volta autenticato, l'utente potrà accedere ad altri servizi senza dover ripetere l'intero processo. Infatti, quando l'utente richiede l'accesso a un nuovo **Service Provider**, questo inoltra un messaggio `<AuthnRequest />` all'**Identity Provider**. Poiché l'utente è già autenticato, l'**Identity Provider** consente l'accesso senza richiedere nuovamente le credenziali.

Authentication Request AuthnRequest

Una richiesta di autenticazione SAML contiene:

- **ID**: identificativo univoco della richiesta.
- **Versione**: la versione del protocollo SAML che è SAML 2.0
- **IssueInstant**: il tempo in cui la richiesta è stata emessa in UTC.
- AssertionConsumerServiceURL: L'interfaccia URL SAML del fornitore di servizi, dove l'Identity provider invia il token di autenticazione.
- **Subject**: l'identità dell'utente che richiede l'autenticazione.
- **Issuer**: l'UID dell'Identity Provider.
- **NameIDPolicy**: politica di identificazione dell'utente.

SAML <Response>

- **ID**: identificativo univoco della richiesta.
- **Versione**: la versione del protocollo SAML che è SAML 2.0
- **IssueInstant**: il tempo in cui la richiesta è stata emessa in UTC.
- InResponseTo: l'ID della richiesta a cui si sta rispondendo.
- **Status**: lo stato dell'operazione di autenticazione.
- **Issuer**: l'UID dell'Identity Provider.
- **Assertion**: contiene informazioni dell'utente autenticato.
- **Signature**: firma digitale dell'Identity Provider.

SAML Authentication Assertion

- **ID**: identificativo univoco della richiesta.
- **Versione**: la versione del protocollo SAML che è SAML 2.0
- **IssueInstant**: il tempo in cui la richiesta è stata emessa in UTC.
- **Subject**: l'identità dell'utente autenticato.
- **Issuer**: l'UID dell'Identity Provider.
- **Conditions**: le condizioni di validità dell'assertion.
- **AudianceRestriction**: specifica il service provider a cui è destinato l'assertion.
- **AuthnStatement**: informazioni sull'autenticazione.
- **AttributeStatement**: lista gli attributi di identità certificati dall'Identity Provider.
- **Signature**: firma digitale dell'Identity Provider.

6.3 SPID - Sistema Pubblico di Identità Digitale

SPID

Lo SPID è il sistema di identità digitale italiano che permette l'accesso ai servizi della Pubblica Amministrazione e delle aziende private accreditate.

6.3.1 Attori principali

- AgID: ente che monitora e certifica i provider SPID.
- **Identity Provider**: entità pubblica o privata che eroga servizi di autenticazione ed è certificato da AgID, ha la responsabilità di verificare l'identità dell'utente.
- **Service Provider**: eroga i servizi richiedendo l'autenticazione via SPID.
- **Attribute Provider**: ente che rilascia agli utenti gli attributi qualificanti.
- **User**: proprietario delle credenziali SPID.

6.3.2 Shibboleth

Shibboleth è un progetto del consorzio Internet2 che consente alle università di condividere risorse e attività di ricerca oltre i confini istituzionali. Grazie a questo sistema, studenti, docenti e personale possono accedere alle risorse delle istituzioni partner senza dover creare credenziali separate.

6.4 `OAuth2.0` e `OpenID Connect`

6.4.1 `OpenID Connect`

`OpenID Connect`

`OpenID Connect` è un protocollo di autenticazione basato su `OAuth2.0` che consente l'autenticazione sicura, decentralizzata e semplificata per gli utenti, permettendo alle applicazioni di verificare l'identità dell'utente e di ottenere informazioni di base sul profilo.

I vantaggi principali di `OpenID Connect` sono:

- Il **Single Sign-On**: permette all'utente di accedere a più servizi con un'unica identità.
- La **Centralità dell'utente**: gli utenti controllano quali dati vengono condivisi con le applicazioni, migliorando la privacy.
- **Interoperabilità**: supportato dai principali provider (*Google, Microsoft, ecc.*) e compatibile con un'ampia gamma di servizi.
- **Sicurezza**: utilizza il protocollo `OAuth2.0` per garantire la sicurezza delle transazioni.

6.4.2 `OAuth2.0`

`OAuth2.0`

È un protocollo di autorizzazione standard che consente a un'applicazione di terze parti di accedere a risorse protette ospitate su un server `HTTP`

- Richiede un token di accesso al server di autorizzazione.
- L'applicazione di terze parti utilizza il token di accesso per richiedere l'accesso alla risorsa protetta.

Attori

- **Resource Owner**: entità in grado di concedere l'accesso a una risorsa protetta.
- **Client**: l'applicazione di terze parti che richiede l'accesso a risorse protette per conto del proprietario della risorsa e con la sua approvazione.
- **Resource Server**: il server che rilascia il token di accesso al client dopo aver autenticato il proprietario della risorsa e aver ottenuto la sua autorizzazione.

- **Authorization Server**: il server che memorizza le risorse del proprietario della risorsa.

Flusso di autorizzazione

1. Il client richiede l'autorizzazione al proprietario della risorsa.
2. Il proprietario della risorsa autorizza il client a accedere alla risorsa.
3. Il client richiede un token di accesso all'authorization server.
4. L'authorization server autentica il client e rilascia un token di accesso.
5. Il client richiede la risorsa al resource server e presenta il token di accesso.
6. Il resource server verifica il token e, se valido, fornisce la risorsa al client.

6.5 Autenticazione basata su Token e Biometrica

Le tipologie di token sono:

- **Hardware**: Smart card, chiavette `USB`.
- **Software**: Applicazioni mobile (`Google Authenticator, Authy`).

I protocolli di autenticazione basati su token sono:

- `OPT` (*One-Time Password*): password monouso generata da un token.
- **Challenge-Response**: il server invia una sfida al client, che la risolve e invia la risposta.

6.5.1 One Time Password

- `SMS`**-based** `OTP`: il server invia un codice via `SMS` all'utente.
- `TOTP`**-based** `OTP`: l'utente genera un codice temporaneo tramite un'applicazione.

`HOTP` (***HMAC-based OTP***)

1. L'utente abilita la `MFA`.
2. Il *server* genera una chiave segreta `K` per lo user.
3. La chiave `K` viene condivisa con l'app di autenticazione sul dispositivo dell'utente.
4. L'app usa un `counter C` che incrementa a ogni `OTP` generata.
5. La `OTP` si calcola come: `HOTP(K,C)` = `Truncate(HMAC-SHA1(K, C))`.
6. L'utente invia la `HOTP` al server, che calcola la stessa funzione e verifica la corrispondenza, poi incrementa `C`.

`TOTP` (***Time-based** `OTP`*)

La differenza principale rispetto a `HOTP` è che si usa il tempo corrente invece di un `counter`, sincronizzando *server* e *app* sull'orario (*Unix time*).

6.6 Biometric Authentication

L'autenticazione biometrica sfrutta **tratti biologici/fisiologici** dell'utente:

- **Universalità**: tutti dovrebbero possedere la caratteristica (es. impronta digitale).
- **Distintività**: il tratto dev'essere univoco.
- **Permanenza**: il tratto non deve cambiare sensibilmente nel tempo.
- **Collezionabilità**: il dato dev'essere catturabile e processabile (es. scanner di impronte).

Tra i metodi più comuni troviamo impronte digitali, riconoscimento vocale, firma e `DNA`. Possibili limiti:

- **Falsi positivi / negativi** dovuti all'accuratezza degli algoritmi di matching.
- **Privacy e accettazione sociale**: non tutti vogliono fornire dati biometrici.
- **Facilità di ricostruzione** (es. impronte da superfici toccate).

Capitolo 7

Access Control

7.1 Introduzione

Controllo degli Accessi

Il **controllo degli accessi** è un elemento centrale della `cybersecurity`, prevenendo accessi non autorizzati a risorse o l'uso improprio di risorse da parte di entità non autorizzate coinvolgendo utenti e gruppi.

7.1.1 Principi fondamentali

- **Authentication**: verifica l'identità dell'entità che richiede accesso.
- **Authorization**: assegna i permessi per le risorse.
- **Accountability**: monitora e traccia gli accessi e l'uso delle risorse.

7.1.2 Elementi delle policy di accesso

- **Soggetto**: entità che richiede accesso a una risorsa.
- **Oggetto**: risorsa il cui accesso è controllato.
- **Permessi**: le operazioni consentite al soggetto sugli oggetti.

7.2 Modelli di `Access Control`

7.2.1 `DAC` - Discretionary Access Control

L'accesso agli oggetti di dati (*file, directory, ecc.*) è consentito in base all'identità degli utenti. Ci sono regoli di accesso esplicite che stabiliscono chi può e non può eseguire determinate azioni e su quali risorse.

È discrezionale, il che significa che gli utenti possono avere la possibilità di trasmettere i propri privilegi ad altri utenti, dove la concessione e la revoca dei privilegi è regolata da una politica amministrativa, spesso fornita tramite una **matrice di accesso**.

Strutture di Controllo in DAC

- **Access Matrix**: matrice con soggetti, oggetti e permessi (*non scalabile, ovvero difficile da gestire su sistemi complessi*).
- **Access Control List (`ACL`)**: elenca i permessi concessi a ogni oggetto.
- **Capability List**: specifica cosa un soggetto può fare su diversi oggetti.

Limitazioni

- La gestione delle policy diventa complessa su sistemi di grandi dimensioni.
- `ACL` non consente una visione chiara dei permessi assegnati agli utenti.
- `Capability List` non offre una panoramica dei permessi su un determinato oggetto.

7.2.2 `MAC` - Mandatory Access Control

L'accesso è determinato da **etichette di sicurezza** (*security labels*) assegnate a soggetti e oggetti. Gli utenti non possono modificare direttamente i permessi.

7.2.3 `RBAC` - Role Based Access Control

Il modello `RBAC` regola l'accesso alle risorse in base ai ruoli assegnati agli utenti all'interno di un'organizzazione.

Tipologie di `RBAC`

- $\texttt{RBAC}_0$: Nel modello base, ogni utente viene assegnato a un ruolo che possiede un insieme definito di permessi. L'accesso alle risorse è direttamente determinato da tali permessi.
- $\texttt{RBAC}_1$: Questo livello introduce la gerarchia dei ruoli. I ruoli possono essere organizzati in una struttura gerarchica in cui quelli di livello superiore ereditano automaticamente i permessi dei ruoli subordinati, facilitando la gestione e la scalabilità delle autorizzazioni.
- $\texttt{RBAC}_2$: Aggiunge ulteriori controlli attraverso i vincoli di separazione dei compiti (`SoD - Separation of Duties`). Questi vincoli impediscono che un singolo utente possa accumulare ruoli o permessi che potrebbero portare a conflitti di interesse, migliorando così il controllo e la sicurezza all'interno dell'organizzazione.

  La separazione dei ruoli può essere:

  - **Statica**: ad un utente non possono essere assegnati a più di n ruoli nello stesso insieme.
  - **Dinamica**: un utente non può attivare più di n ruoli nel set di ruoli all'interno della stessa sessione.

Vantaggi e limitazioni

- **Pro**: gestione centralizzata e riduzione degli errori nell'assegnazione dei permessi.
- **Contro**: difficile implementazione su sistemi molto complessi con migliaia di ruoli.

7.2.4 ABAC - Attribute Based Access Control

L'engine di autorizzazione prende decisioni basate su attributi dell'utente, dell'oggetto e del contesto.

XACML - Extensible Access Control Markup Language

XACML è uno standard basato su XML per implementare ABAC. Include:
- **Linguaggio per la definizione delle policy di accesso.**
- **Protocollo per richieste e risposte di autorizzazione.**
- **Architettura per la gestione e l'applicazione delle policy.**

7.2.5 Architettura di XACML

- **Policy Enforcement Point** (PEP): riceve richieste di accesso, inoltra le richieste al PDP e applica le decisioni di autorizzazione.
- **Policy Decision Point** (PDP): esamina la richiesta di accesso, recupera le policy applicabili e prende una decisione di autorizzazione.
- **Policy Administration Point** (PAP): definisce e gestisce le policy di accesso.
- **Policy Information Point** (PIP): fornisce informazioni supplementari sulle richieste di accesso.
- **Context Handler**: converte le richieste nel formato XACML e restituisce la risposta convertita in formato nativo.

7.2.6 Flusso di Autorizzazione in XACML

1. **Definizione delle policy**: Il PAP scrive e memorizza le policy di accesso.
2. **Richiesta di accesso**: Il PEP riceve una richiesta di accesso da un soggetto.
3. **Valutazione della richiesta**: Il Context Handler inoltra la richiesta al PDP, includendo gli attributi rilevanti.
4. **Raccolta di informazioni**: Se necessario, il PDP interroga il PIP per ottenere ulteriori dati.
5. **Decisione di accesso**: Il PDP valuta la policy e invia la decisione finale al PEP.
6. **Applicazione della decisione**: Il PEP concede o nega l'accesso alla risorsa.

7.2.7 Componenti Chiave di `XACML`

- **Policy Set**: Aggrega più policy o altri policy set.
- **Policy**: Contiene regole (`Rules`) e condizioni per prendere decisioni di accesso.
- **Target**: Definisce a quali richieste si applica una policy.
- **Rule**: Esprime le condizioni per l'accesso, specificando quando una policy si applica.
- **Algoritmo Combinatorio**: Specifica come combinare le decisioni di più policy per ottenere una decisione finale.

7.2.8 Algoritmi di Decisione

Per combinare le policy e ottenere una decisione finale, `XACML` definisce i seguenti algoritmi:

- **Deny Overrides**: Se almeno una regola restituisce `Deny`, l'accesso viene negato.
- **Permit Overrides**: Se almeno una regola restituisce `Permit`, l'accesso è concesso.
- **First-Applicable**: Viene applicata solo la prima policy valida.
- **Only-One-Applicable**: Se più di una policy è applicabile, il risultato è indeterminato.