

SICUREZZA

- La sicurezza è un processo, niente è al 100% sicuro
- La sicurezza di un sistema dipende dalle sue componenti meno sicure (spesso è l'utente finale)
- Non spiegare come funzionano gli algoritmi non li rende più sicuri.
- La critografia è il miglior elemento ma non è sufficiente

Cosa significa sicurezza?

- > È la protezione effettuata ad un sistema di informazione automatizzato per ottenere obiettivi applicabili per preservare integrità, disponibilità e confidenzialità dell'informazione delle risorse del sistema.
- > La sicurezza non riguarda solo i dati ma hardware, software, firmware e telecomunicazioni.
- * TRIADE CIA (confidenzialità, integrità, disponibilità)
 - perdere confidenzialità vuol dire che qualcuno ha avuto un accesso non autorizzato a delle informazioni riservate. GARANTIRE CONFIDENTIALITÀ = garantire che nessun altro abbia accesso a info. confidentiali
 - perdere l'integrità comporta la modifica o la distruzione di operazioni
 - perdere disponibilità vuol dire che si perde la possibilità di accedere a risorse a cui dovremmo poter accedere

CONFIDENTIALITÀ → dati confidentiali: info private che non devono essere accessibili
 → PRIVACY: scegliere l'utente cosa è privato e cosa no

Concetti addizionali:

- autenticità: non solo i dati devono essere integri ma posso dimostrare di averli generati. È anche la possibilità di dimostrare che gli utenti sono chi dicono di essere.
- accountabilità: rintracciare chi è responsabile di cosa. Bisogna tener traccia delle attività fatte perché non me ne accorgo sul momento.

TIPI DI ATTACCO

- > attacco PASSIVO: l'attaccante legge le informazioni, non interagisce
 - > attacco ATTIVO: l'attaccante legge, modifica, distrugge e genera info
- Qui attacchi passivi sono più difficili da identificare, possiamo fare prevenzione.
 Qui attacchi attivi si dividono in 4 categorie:
1. Replay: catturano i dati e li riutilizzano più avanti per produrre un effetto non autorizzato
 2. Masquerade: cercano di prendere il posto di un altro persone
 3. Modifiche del messaggio: sono sulla linea, catturano messaggi, li modificano e li inoltrano
 4. Negare il servizio

ATTACCANTI:

1. HACKER: in generale lo fa per il gusto di farlo e non per soldi e per prestigio
 - BLACK HAT: malintenzionati
 - WHITE HAT: divertimento e segnalazione
2. INTERNO: persone che lavorano o lavoravano pr. es. lavoratore che viene assunto da azienda concorrente.
3. GRUPPI CRIMINALI: gruppi organizzati, si scambiano info e si coordinano. Hanno in genere dei target e cercano di essere ricoperti.
4. ADVANCED PERSISTENT THREAT: gruppi sponsorizzati dalle nazioni (es. gruppo militare cyberspazio) fanno attacchi continui per raggiungere l'obiettivo

QU'È IL MALWARE?

Ha l'obiettivo di causare danni o ottenere risorse. Spesso viene nascosto.
Ci sono tanti tipi di malware.

(2)

1. BACKDOOR: non sono necessariamente malware. Sono punti di accesso segreti a programmi, può essere lo stesso programmatore a lasciarli. Sono usati anche per attacchi. È difficile identificare la presenza.
2. CAVALI DI TROIA: programmi o comandi apparentemente utili che contengono codice che eseguito provoca qualcosa di dannoso. I cavalli di troia possono:
 - continuare a fare quello che faceva il programma originale + altre attività
 - continuare a fare quello che faceva il programma ma in modo diverso per nascondere attività malevole
 - fare qualcosa di diverso dal programma originale
3. PLATFORM INDEPENDENT CODE: malware che funzionano su qualsiasi sistema e quindi sono altamente dannosi
4. VIRUS: software che infettano altri programmi modificandoli. Contengono codice che vengono viene trasferito all'interno del programma. Può diffondersi. Un virus è composto da 3 parti: un meccanismo di infezione (modem file o di diffusione), un trigger (quello che l'attiva) e un payload (parte di codice che deve essere esattamente copiata). I virus vanno ad infettare nel bootsector (quindi in fase di avvio), i file o le macro.
5. MALWARE MULTIPLE-THREAT: usano diversi veicoli di infezione.
6. ROOTKIT: è un insieme di programmi che vengono installati per mantenere l'accesso da amministratore (può essere anche benevolo). I rootkit possono essere persistenti quindi vengono riavviati ogni volta che vengono riavviati i sistemi, ~~oppure sono~~ memory based, user mode o kernel mode.

BUFFER OVERFLOW

È un comune meccanismo di attacco. È dovuto ad una gestione sbagliata di un buffer che permette di scrivere più byte del possibile sovrascrivendo celle già usate oppure usando lo stack. L'attaccante deve analizzare il software per trovare la vulnerabilità.

Sovrascrivono il buffer con il codice dell'attacco e alterano il return address affinché parta dal punto del codice (spesso apre una shellcode). Più il linguaggio è di basso livello più è facile manipolare la memoria. Abbiamo 2 meccanismi di difesa:

- Compile-time defenses: sistemi introdotti nei compilatori.
- Stack protection mechanisms: utilizzare funzioni che considerino la lunghezza del buffer come strcpy
- Canarini: cercare di identificare un buffer overflow sul return address. La funzione identifica, tramite una variabile, un bufferoverflow perché quando controllerò la mia variabile la troverò modificata dal momento che un attaccante non può evitare di modificare anche il canarino.
- DEP/Nx bit: l'idea è quella di tenere separate l'area dati dall'area dell'esecuzione. Dunque anche se inserisco codice in memoria non riesco ad eseguirlo direttamente dall'area dati.

7. BOTS:

È un programma che comunica con altri in internet e risiede nel nostro computer. Viene chiamato anche zombie perché sta fermo, poi agisce ed è privo di intelligenza oppure viene chiamato drone perché è telecomandato. Un bot viene posizionato nel computer ma è utile quando si crea un botnet cioè una rete di bot che agisce in modo coordinato. Ogni botnet è composto da: bot, controllo remoto e meccanismo di spreading (per la diffusione). ~~Oltre~~ L'obiettivo iniziale è estendere la rete infettando più dispositivi possibili. Gli elementi chiave sono: un software che possa effettuare l'attacco, sfruttare una vulnerabilità presente in molti sistemi e ~~che~~ delle strategie per controllare altri dispositivi con vari criteri. I bots possono fare tante cose:

- attacco distributed denial-of-service: impedire ad altri utenti di accedere ai servizi
- spamming: invio di mail
- sniffing traffic: ottenere info sul traffico

- Keylogging: catturare quello che viene digitato (username, password)
- diffusione di un nuovo malware
- installare add-ons pubblicitari nei browser
- attaccare chat
- manipolare poll e giochi

Uno degli attacchi principali è il denial-of-service cioè impedire l'uso di reti, sistemi o applicazioni da utenti autorizzati consumando risorse. Attaccare i computer in questo modo non è molto efficace, lo è più attaccare direttamente i server. Classico attacco è saturare la rete tramite richieste di connessione: ~~TCP, UDP~~ ICMP, UDP e TCP SYN. Come ci si può difendere?

Cercando di evitare picchi di traffico che però potrebbero essere leggibili: dunque ci sono 3 linee di difesa: preventione, identificazione di denial-of-service e filtro del traffico, creare un traceback allo scopo di partire da azioni legali.

Come si fa PREVENTIONE?

- Bloccando indirizzi spoofed e da cui quelli sono già partiti attacchi
- limitando il traffico in upstream per specifici pacchetti (ICMP, TCP, UDP)
- usare i cookies
- bloccando IP broadcast
- bloccando servizi e combinazioni sospetti
- utilizzando puzzle per distinguere umani e non umani
- utilizzando server replicanti

Come si risponde agli attacchi?

servire le coinvolgimento degli ISP che devono impostare filtri in uscita. Si deve catturare e analizzare i pacchetti, identificare i bug usati.

TECNICHE DI PREVENZIONE

La tecnica più usata è la criptografia:

la base di tutto sono i numeri casuali che si ottengono con algoritmi oppure tramite eventi catturabili ad esempio con sensori. Alcuni algoritmi sono: meccanismi sono:

- Symmetric Encryption: abbiamo un input e una chiave, chi riceve ha la chiave e quindi decifra. Il dato cifrato ha la stessa dimensione del dato originale. È veloce. Il problema è che mittente e ricevente devono avere la chiave e deve essere diversa per ogni ricevente. Si può attaccare facendo criptoanalisi per ricostruire la chiave oppure con brute-force cioè provando tutte le chiavi (più grande la chiave più è difficile)
- Public Key Encryption: cifratura efficiente, decifratura onerosa. L'algoritmo ~~cifra~~ cifra con la chiave pubblica ~~per~~ del ricevente che decifra con una chiave privata (modalità asimmetrica). Diversi algoritmi usano questo meccanismo

Le chiavi asimmetriche sono usate anche per l'AUTENTICAZIONE: in questo caso quando voglio trasmettere qualcosa non lo voglio proteggere in modo che solo il mittente possa aprirlo ma chiunque nel mezzo potrebbe vederlo. Il mittente cifra con la sua chiave privata e il ricevente capisce che il mittente è chi dice di essere se riesce a decifrarlo con la sua chiave pubblica. L'autenticazione serve per proteggersi dai attaccanti attivi e per verificare che il messaggio è autentico. Ci sono vari metodi:

1. Algoritmo MAC:

Dato un messaggio e una chiave calcola una specie di cifrato con la caratteristica che l'output ha una certa dimensione indipendente dalla ~~lunghezza~~ quelle del mess. orig. Calcolato il MAC lo aggiungo al messaggio e trasmetto, chi riceve separa il MAC dal resto del messaggio e se conosce la chiave verifica che il messaggio l'ho mandato io. Serve a 2 scopi: autenticare e verificare che non è stato modificato. Il messaggio qui è in chiaro. Un MAC famoso è le Secure Hash Functions che dato un messaggio di dim. M genera dei valori hash h che, qualora il mess. venisse modificato, cambia molti dei bit. Modi per usare l'hash:

- Mess. + hash crittografato con chiave simmetrica
- Mess. + hash crittografato con chiave asimmetrica
- Valore segreto + ~~Mess.~~ mess., calcolo l'hash del mess+segreto e trasmetto mess+hash, se il ricevente conosce il segreto decifra

Un altro meccanismo MAC sono i certificati. Si prende il mess., si fa l'hash che viene cifrata con una chiave privata che ha il certificato di autenticità (non generata da me) e trasmetto. Il ricevente può decifrare con una chiave pubblica sempre richiesta all'autorità certificante. Lo usano ad esempio le granole attivate per aggiornare i dispositivi.
Si arriva alle digital envelopes che mette insieme le tecniche.

Per autenticare noi stessi (e non il messaggio) ci sono vari metodi:

- **username e password**: la password non deve essere memorizzata in chiaro da un server. Non si memorizza la password ma un suo hash con il SALT cioè un numero casuale memorizzato in chiaro. Per verificare la password confronto l'hash code con l'hash di salt e password.
- **token**: non mi identifico con qualcosa che so ma con qualcosa che posiedo, cioè una "carta". All'inizio si usava una memory card che non poteva ~~calcolare~~ elaborare dati. Ora si usa una smart card (con processori embedded). Servono per entrambi lettori appositi.
- **autenticazione biometrica**: mi identifico con qualcosa che sono. Es. impronta digitale. Sono caratteristiche univoche e devono avere poca variabilità nel tempo.

ACCESS CONTROL (controllo dei permessi)

È la parte di sicurezza che gestisce cosa possiamo fare dopo, cioè una volta entrati nel sistema. Abbiamo 3 tipi di controllo dell'accesso:

1. **Discretionary Access Control (DAC)**: si stabilisce, per ogni utente e per ogni oggetto, cosa l'utente può fare con quell'oggetto.

2. **Mandatory Access Control (MAC)**: è un sistema di controllo basato sui label, cioè ogni utente ha i permessi riguardo alle label. (la label è sull'oggetto)

3. **Role-based access control (RBAC)**: non si mette la label all'oggetto ma alla persona. Ogni sistema deve avere almeno uno dei tre ma può usarne più di uno. L'access control è una matrice in cui abbiamo una riga per l'utente e una per l'oggetto in cui negli oggetti ci sono disci, processi, file ecc...

Nel RBAC l'idea è catalogare gli utenti e poi dare loro i permessi ~~rispettive~~ tramite il gruppo. Nella matrice non abbiamo più gli utenti ma i ruoli e abbiamo un'altra matrice che dice qual è il ruolo di ogni utente.

ANTIVIRUS

L'antivirus fa detection, identification e removal. Controlla se vi è una compromissione e se riconosce qualcosa che non va cerca di identificare il tipo di virus per poi rimuoverlo insieme a tutte le sue attività ma per fare questo deve conoscere bene il virus.

Fortunatamente i virus appartengono a delle famiglie quindi una volta riconosciuta la famiglia ~~il~~ l'antivirus (che ormai è online) ottiene le info sul virus e sui procedure di prendere. Il virus però potrebbe essere voluto dall'utente (un crack) quindi, una volta che l'antivirus lo ha messo in quarantena, deve abilitarlo. L'antivirus, per non eseguire i file, contiene al suo interno un emulatore semplice della macchina stessa dove viene decifrato ed eseguito il file. Quindi un antivirus contiene un emulation control model, un emulatore della CPU e uno scanner della signature (per riconoscere le caratteristiche delle famiglie di virus).

L'antivirus non è ottimizzato per i sistemi distribuiti per i quali invece si usano i Digital Immune System. Nel momento in cui si infetta una macchina del sistema avrà una componente del DIS che comunica ad una macchina di amministrazione la presenza del virus e questa lo comunica ad una macchina esterna che fa l'analisi completa del virus. E seguita l'analisi, la macchina esterna comunica a quelle di amministrazione le procedure da seguire e questa lo comunica alla macchina infettata e anche alle altre macchine che potrebbero già essere infettate. La macchina esterna comunica di farsi anche alle altre reti private (che potrebbero essere di altre aziende) in modo da ricevere a sua volta segnalazioni su altri virus da queste in futuro. Dunque c'è ~~una~~ cooperazione fra aziende.

FIREWALLS

(5)

L'antivirus è la cura ma il firewall è lo ~~cuffia~~ il vaccino. Il firewall protegge la rete bloccando dei dati, è un singolo punto di controllo (un muro) infatti se abbiamo più accessi dobbiamo mettere più firewalls. fa monitoring ma ci sono comunque attacchi che riescono ad aggirarlo ad esempio quelli all'utente finale (es. virus per email) perché il firewall non fa analisi del dato ma della connessione, controlla chi manda a chi, blocca connessioni non autorizzate ecc...

Dunque l'idea è quella di piazzare un firewall fra la rete esterna e quella interna. Ci sono vari tipi di Firewall, quello più semplice è il packet filtering firewall che lavora al livello di trasporto e si basa sui coppi TCP-IP. Ci sono firewall più avanzati come l'Application proxy firewall che lavora al livello applicativo.

Come opera il packet filtering firewall? Controlla ~~che~~ l'IP di sorgente e destinazione della porta, ~~che~~ il protocollo e l'interfaccia e in base a delle regole decide se il pacchetto deve essere scartato. Ci sono 2 politiche di default per decidere come opera un firewall:

- scarso tutto: devo specificare cosa per cosa se può passare uno
- trasmetti tutto: a meno che non sia esplicitamente proibito (più facile)

Quelli svantaggi dei firewall sono:

- non possono prevenire attacchi basati su vulnerabilità
- fanno poco logging (quindi non è facile poi fare l'analisi)
- sono vulnerabili agli attacchi sui bug dei protocolli TCP/IP

Quando siamo in un sistema distribuito conviene avere un firewall che separa le reti interne da tutto ciò che è esterno, ~~e~~ una zona DMZ network dove ci sono cose accessibili sia dall'interno che dall'esterno (quindi protetta da un firewall esterno) e una zona accessibile solo dall'interno (protetta da un altro firewall interno).

INTRUSION DETECTION SYSTEM.

L'idea è quella di riuscire a riconoscere il traffico malevolo, bloccare login non autorizzati, bloccare l'abuso di privilegi. Questi sistemi hanno tecniche avanzate e si basano sul fatto che un intruso si comporta diversamente rispetto ad un utente normale.

Un sistema funziona bene quando è distribuito dove abbiamo ogni rete con il suo agente dell'IDS, un nodo centrale che fa da monitor e colleziona i dati per l'intera rete e qualcuno che lo gestisce a livello decentralizzato.

HONEY POTS

È un sistema finto che simula la mia rete. Se arriva un attacco prima di riuscire a passare il firewall finisce nell'honey pot e lì si diffondono però compromessi sistemi che sono finti. L'esperto di sicurezza monitora l'attività sulle honeypots e riesce ad anticipare quello che potrebbe succedere.