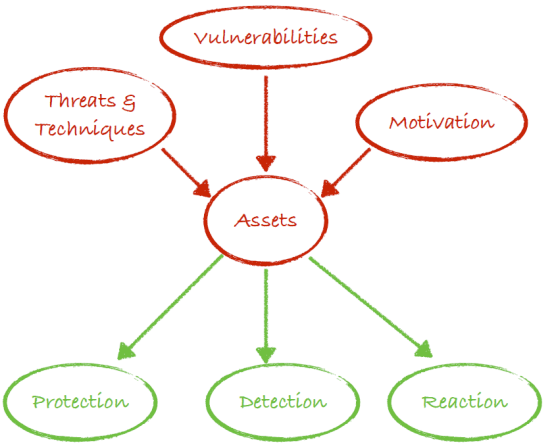
Introduzione

Quando si parla di sicurezza o di violazione alla sicurezza ci si riferisce all'accadere di **fallimenti intenzionali** del sistema, che vanno ad arrecare un qualche tipo di danno → il concetto fondamentale, quindi, è che la violazione alla sicurezza è qualcosa di intenzionale per andare ad arrecare un danno.

Per parlare di sicurezza vi sono tre livelli da considerare:

1. il livello degli **asset** → ovverosia con il termine asset intendiamo l'oggetto che noi vogliamo proteggere;
2. il livello delle metodologie che possono essere utilizzate per mitigare gli attacchi alla sicurezza. Tali metodologie possono diramarsi in:
    a. Protezione → ovverosia tutti quei provvedimenti, che si possono prendere per limitare la possibilità agli attaccanti di attaccare il sistema;
    b. Tecnologie per la Detection → ovverosia se l'attacco accade, vogliamo essere in grado di individuare l'attacco stesso;
    c. Reazione → ovverosia come reagiamo all'attacco in sé.
3. il livello che riguarda cosa ha permesso all'attacco di manifestarsi, quindi:
    a. la presenza di potenziali minacce;
    b. la presenza di vulnerabilità, che gli attaccanti hanno sfruttato;
    c. le motivazioni degli attaccanti, in quanto quest'ultimi per attaccare il sistema devono guadagnare qualcosa.

Questi livelli possono essere visti graficamente con la seguente immagine:



La sicurezza come abbiamo detto, riguarda certamente gli assets e le vulnerabilità, ma è certamente anche un **people problem**, ovverosia se le persone utilizzano il sistema in maniera non appropriata, questo può certamente aprire superfici di attacco. Vi è, quindi, la necessità di istruire e rendere consapevoli le persone. La sicurezza delle informazioni, o in generale la sicurezza del software, ha delle caratteristiche peculiari rispetto alla sicurezza di un oggetto materiale, in quanto:

- le informazioni/il software possono essere rubati, ma voi ne siete ancora in possesso;
- le informazioni riservate o il software proprietario possono essere copiati e venduti, ma il furto potrebbe non essere rilevato ed il criminale potrebbe trovarsi all'altro capo del mondo.

La sicurezza informatica si occupa, quindi, della prevenzione e dell'identificazione di azioni non autorizzate da parte degli utenti di un sistema informatico su quelli che sono gli assets, ovverosia:

- l'**hardware** → quale: laptop, cellulari, smart card;
- il **software** → quale: applicazioni, codice sorgente, SO, DBMS;
- i **dati** e le **informazioni** → quali: dati sui clienti, dati essenziali per la gestione, documenti di progettazione. Da notare, che il valore dei dati è difficile da misurare, perché ogni dato deve essere considerato e interpretato nel contesto giusto, anche perché questo valore può cambiare nel corso del tempo → vi è, quindi, un concetto molto importate, ovverosia quello della **temporalità**. Tale concetto, consiste nel fatto che **gli elementi di un sistema informatico devono essere protetti solo fino a quando hanno un valore**. (Questo principio vale naturalmente anche per gli altri assets).
- i **servizi;**
- la **reputazione**.

Iniziamo ad analizzare la parte alta dell'immagine riportata sopra, ovvero innanzitutto andiamo ad analizzare e classificare le minacce. Partiamo con il dire che le minacce sono tantissime, come per esempio: Cyberterrorism, Social Engineering, Stolen Credentials e Software Hacking e di conseguenza **per decidere se un sistema informatico è sicuro, è necessario innanzitutto decidere da che cosa è sicuro il sistema, ovverosia dobbiamo identificare le minacce che ci interessano**.

In questo senso, vi sono le **Security policy** → ovverosia una dichiarazione (= una serie di indicazioni) che definisce gli obiettivi di sicurezza di un'organizzazione.

Deve, quindi, indicare ciò che deve essere protetto e può anche indicare le modalità di protezione. Da notare, che proteggere gli assets di un'organizzazione è compito del management e le misure di sicurezza, spesso, limitano i membri dell'organizzazione nel loro lavoro e di conseguenza si può avere la tentazione di non rispettare le regole di sicurezza. Collegato a ciò, è importante sottolineare, che non tutti i membri dell'organizzazione devono diventare esperti di sicurezza, ma tutti i membri devono essere consapevoli sull'importanza della sicurezza e sulle buone pratiche da seguire.

Quando si inizia ad analizzare la sicurezza di un sistema, devono essere individuate le potenziali minacce da cui difendersi. **Una minaccia è un impatto negativo indesiderato su un'attività** e vi sono diverse classificazione delle minacce → per identificare le minacce più comuni è nato Stride, ovverosia un modello di minaccia, in cui ogni lettera della parola Stride corrisponde ad una tipologia di minaccia e quindi ci dà una sorta di checklist nel momento in cui andiamo ad individuare le minacce.

STRIDE Threat Model

Gli attacchi di spoofing sono tutti quegli attacchi in cui l'attaccante si presenta come chi non è, al fine di farci rivelare le credenziali di accesso ai sistemi. Alcuni esempi sono: Caller ID, Website spoofing, email spoofing, text message spoofing.

💡 Gli attacchi di ripudio, sono quegli attacchi in cui l'attaccante può negare di aver commesso una qualche azione, oppure può dichiarare il falso (es: l'attaccante ha prelevato 1000 euro, ma dice di averne prelevati solamente 100) → per difendersi da questi attacchi si fanno i log, ovverosia il sistema tiene traccia delle azioni compiute da ogni utente del sistema → per fare questo, è necessario che vi sia un sistema di autenticazione, in modo tale da associare l'utente all'azione.

💡 il rilascio di informazioni è causato da un attacco di Data Breach, mentre l'elevazione dei privilegi può essere verticale o orizzontale, ovvero l'attaccante riesce ad impersonificare l'utente con i privilegi di cui necessita, oppure l'attaccante riesce ad ottenere le credenziali di un utente che ha accesso al sistema, ma che non ha i privilegi di cui necessita e quindi deve riuscire ad ottenere i privilegi di cui necessita

💡 l'attacco di tampering with code, ovverosia la manomissione del codice. Questa manomissione può riguardare il bypass delle licenze, oppure modificare il comportamento del codice.

Una volta che abbiamo individuato le minacce, dobbiamo capire il motivo per cui il sistema ne è soggetto e quindi passiamo ad analizzare le vulnerabilità del sistema. **Le vulnerabilità sono punti deboli di un sistema che potrebbero essere sfruttate accidentalmente o intenzionalmente per danneggiare le risorse**. Per esempio, un sistema governato da password di default, che non vengono mai modificate, è soggetto a vulnerabilità. Stesso discorso vale anche, se i programmi godono di privilegi non necessari oppure hanno dei flussi noti. L'identificazione delle vulnerabilità comprende due macro-azioni:

1. Gli **scanner delle vulnerabilità** → i quali forniscono un modo sistematico e automatizzato di identificare le vulnerabilità;
2. L'**analisi del rischio** → l'analisi del rischio deve misurare la criticità delle vulnerabilità. La criticità di una vulnerabilità dipende dagli attacchi che potrebbero sfruttarla. Una vulnerabilità che consente ad un aggressore di impersonificare completamente un utente è più critica di una vulnerabilità che consente di impersonificare un utente solamente in un servizio specifico.

Una volta individuate le vulnerabilità, passiamo agli attacchi. Una minaccia si concretizza quando un attacco ha successo. La gravità di un attacco dipende da:

- probabilità che venga sferrato;
- probabilità che abbia successo;

  La probabilità dipende dalla difficoltà, dalle motivazione e dalle contromisure esistenti.
- dai danni che potrebbe causare.

La metodologia DREAD (di Microsoft) dimostra che sia possibile misurare, in modo sistematico, la gravità di un attacco attraverso cinque fattori:

1. **Potenzialità del danno** → si riferisce al valore degli assets colpiti;
2. **Riproducibilità** → gli attacchi facili da riprodurre hanno maggiori probabilità di essere dall'ambiente rispetto ad attacchi che funzionano solo in specifiche circostanze;
3. **Sfruttabilità** → indica lo sforzo, le competenze e le risorse necessarie per sferrare un attacco.;
4. **Utenti colpiti** → il numero di risorse colpite contribuisce al danno potenziale;
5. **Scopribilità** → l'attacco verrà rilevato? Nel caso più dannoso, l'utente non saprà mai che il sistema è stato compromesso.

Una volta che si ha fatto un’analisi degli assets che vogliamo proteggere, delle minacce, delle vulnerabilità e degli attacchi, dobbiamo combinare tutte le informazioni ottenute per fornire un’analisi del rischio. Vi sono metodi quantitativi e qualitativi per fornire tale analisi:

- **metodi quantitativi** → si basano su misure statistiche, ma il problema è che la distribuzione iniziale non è nota di tali misure statistiche. Di conseguenza non sono sempre utili e spesso sono molto complessi e per questi motivi si preferisce utilizzare i metodi qualitativi;
- **metodi qualitativi** → i quali forniscono delle misure con delle scale sul:
  - valore degli assets;
  - la criticità delle vulnerabilità;
  - la probabilità delle minacce.

Il risultato dell'analisi dei rischi è un elenco prioritario di minacce, insieme alle contromisure consigliate per mitigare il rischio. La conduzione di un'analisi dei rischi per un'organizzazione di grandi dimensioni richiede tempo, è molto costosa e il mondo esterno continua a cambiare. Per questi motivi, le organizzazioni possono optare per una protezione di base come alternativa, anche se l'analisi dei rischi, delle minacce, delle vulnerabilità e degli asset è fondamentale per identificare le giuste contromisure di sicurezza da utilizzare in un'organizzazione.

Una volta che abbiamo fatto un’alisi dei rischi, delle minacce, delle vulnerabilità e degli asset, è necessario mettere in campo delle misure di protezione, che andranno a:

- **prevenire** gli attacchi;
- **identificare** gli attacchi;
- **reagire** agli attacchi.

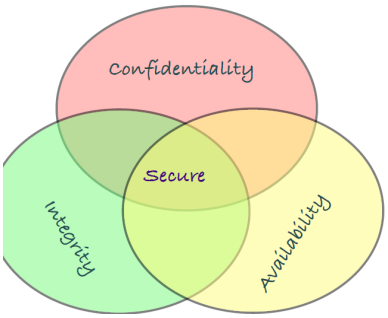
Security targets

La triade della sicurezza è:

- **Confidenzialità** → Storicamente la sicurezza è strettamente legata alla segretezza. La sicurezza coinvolgeva poche organizzazioni, che si occupavano principalmente di dati classificati e di conseguenza gli utenti non autorizzati non devono venire a conoscenza di informazioni sensibili. La riservatezza comporta:
  - privacy → ovverosia la protezione dei dati personali (gli individui controllano o influenzano le informazioni che li riguardano e che possono essere raccolte e conservate);
  - segretezza → ovverosia la protezione dei dati organizzativi (le informazioni private o confidenziali non vengono rese disponibili o divulgate a persone non autorizzate).
- **Integrità** → ovverosia “assicurarsi che tutto sia come dovrebbe essere”. L'integrità si occupa di prevenire la scrittura non autorizzata (può essere vista come il doppio della riservatezza - meccanismi simili). Con integrità intendiamo:
  - integrità dei dati → assicurarsi che le informazioni e i programmi vengano modificati solo in modo specifico ed aurizzato;
  - integrità del sistema → assicurarsi che un sistema svolga le funzioni per cui è stato progettato in modo ineccepibile e senza manipolazioni non

autorizzate del sistema.

- **Disponibilità** → assicura che i sistemi funzionino tempestivamente e che il servizio non sia negato agli utenti autorizzati. Nel contesto della sicurezza, vogliamo garantire che un malintenzionato non possa impedire agli utenti legittimi di avere un accesso ragionevole ai loro sistemi. In altre parole, vogliamo prevenire il denial of service, ovverosia l'impedimento dell'accesso autorizzato alle risorse o il ritardo di operazioni critiche in termini di tempo.



L’equilibrio di questi tre aspetti porta alla sicurezza.

L’obiettivo della sicurezza, quindi, è garantire queste tre proprietà e per garantire tali proprietà è fondamentale un meccanismo di controllo degli accessi e quindi un meccanismo di autenticazione. Siccome l’autenticazione non è sufficiente e gli attacchi si manifestano, vi sono ulteriori proprietà di sicurezza che sono state individuate:

- **Accountability**→ il sistema è in grado di fornire audit trails, ovvero un insieme di record che forniscono prove documentali della sequenza di attività che hanno influenzato in qualsiasi momento un’operazione, una procedura o un dispositivo specifico. Poiché i sistemi veramente sicuri non sono ancora un obiettivo raggiungibile, dobbiamo essere in grado di risalire da una violazione della sicurezza ad un responsabile. A tal fine, il sistema deve:
  - identificare e autenticare gli utenti;
  - mantenere una traccia audit degli eventi rilevanti per la sicurezza;
  - vi è il concetto di **Rensability** → ovvero le informazioni di log devono essere registrate in modo sicuro, al fine di poter associare ogni azione potenzialmente insicura al diretto responsabile.
- **Authenticity** → ovverosia essere in grado di poter verificare che gli utenti siano quelli che dicono di essere e che ogni input che arriva al sistema provenga da una fonte attendibile;

- **Non-repudation** → i servizi di non ripudio forniscono una prova che un’azione specifica si è verificata, come ad esempio la consegna di un messaggio ad un destinatario specifico. Solitamente il non ripudio si ottiene attraverso le firme digitali.

Chiaramente non vi è un meccanismo certo e sicuro per progettare sistemi privi di falle di sicurezza, però vi sono dei principi comunemente riconosciuti e condivisi, per rendere più complicato l’attacco degli attaccanti. Tali principi sono i seguenti:

- Principio di economia del meccanismo → i meccanismi di sicurezza devono essere il più semplice e piccolo possibile. Se il design e l’implementazione sono semplici, ci sono meno possibilità di commettere errori e conseguentemente anche i processi di test e verifica sono meno complessi. Con un progetto complesso, ci sono molte più opportunità per un
un avversario di scoprire sottili debolezze da sfruttare che potrebbero essere difficili da individuare in anticipo;
- Principio di sicurezza in caso di guasto → a meno che a un soggetto non sia stato concesso l'accesso esplicito a un oggetto, gli deve essere negato l'accesso a quell'oggetto;
- Principio della mediazione completa → richiede di controllare tutti gli accessi agli oggetti, al dine di garantire che siano consentiti;
- Principio dell’open design → afferma che la sicurezza di un meccanismo non dovrebbero dipendere esclusivamente dalla segretezza della sua progettazione o implementazione. I progettisti e gli implementatori non devono basare i loro meccanismi di sicurezza sulla protezione del design e dell’implementazione dei dettagli. Se la forza di un sistema di sicurezza si basa sull'ignoranza dell'utente, un utente esperto può sconfiggere il sistema;
- Principio della separazione dei compiti → se per eseguire un'azione critica sono necessari due o più passaggi, allora ci devono essere due utenti diversi che eseguono queste azioni;
- Principio della separazione dei privilegi → un sistema non dovrebbe concedere permessi basati su una singola condizione. Per ottenere l'accesso a una risorsa riservata sono necessari più attributi di privilegio;
- Principio del minimo privilegio → ad un soggetto devono essere concessi solo i privilegi di cui ha bisogno per portare a termine i suoi compiti. Ogni processo e ogni utente del sistema dovrebbe operare utilizzando il minimo di privilegi necessari per svolgere il compito. Se un soggetto non ha bisogno di un diritto di

accesso, non dovrebbe averlo. Vi è anche l’aspetto temporale, ovvero i programmi di sistema o gli amministratori, che godono di privilegi particolari, devono avere tali privilegi solo quando sono necessari; quando, invece, svolgono attività ordinarie, i privilegi dovrebbero essere ritirati. Lasciarli in funzione non fa altro che aprire la porta agli incidenti;

- Isolamento → la forma più semplice di protezione è l’isolamento. Limitare il numero di sistemi in cui sono archiviate le informazioni critiche e isolarli fisicamente o logicamente.