Crimini Informatici e Sicurezza dei Computer e delle Reti

Università degli Studi di Enna "Kore"

Giovanni Giuseppe Iacuzzo - Giovanni Micciche 28 febbraio 2026

Indice

1 Fondamenti di Sicurezza Informatica, Gestione del Rischio e Vulnerabilità di Rete **2**

1 Fondamenti di Sicurezza Informatica, Gestione del Rischio e Vulnerabilità di Rete

La sicurezza informatica nasce dall'esigenza imprescindibile di proteggere i dati e le comunicazioni da accessi non autorizzati, alterazioni o interruzioni. I pilastri su cui si fonda questa disciplina sono essenzialmente cinque: la riservatezza, che garantisce che l'informazione non venga intercettata da soggetti terzi; l'autenticazione, che assicura l'identità dell'interlocutore (ad esempio tramite sistemi biometrici o credenziali); l'integrità, che certifica che i dati ricevuti siano esattamente quelli trasmessi senza alcuna manipolazione; il non ripudio, che impedisce all'autore di un'operazione di negare di averla compiuta; e, infine, la disponibilità, che assicura che i servizi e i dati siano sempre accessibili e non vengano resi inservibili da attacchi, come ad esempio i Denial of Service (DoS). Per garantire integrità e non ripudio, uno degli strumenti più efficaci è la firma digitale, un sistema crittografico basato sull'uso di chiavi asimmetriche (una privata per firmare e una pubblica per verificare) che attesta in modo inequivocabile l'autenticità del mittente e la validità del documento.

Nonostante l'evoluzione dei protocolli e l'implementazione di sistemi di protezione sempre più sofisticati, il fattore umano rimane costantemente l'anello debole della catena di sicurezza. Molte delle vulnerabilità odierne non derivano da falle tecniche complesse, ma dalla pigrizia o dall'inesperienza degli utenti e degli amministratori, i quali spesso prediligono l'usabilità alla sicurezza. Quando un sistema di autenticazione risulta troppo invasivo o fastidioso, l'utente tende ad aggirarlo. A questo si aggiunge la natura intrinsecamente insicura dei protocolli di rete storici. Il protocollo Ethernet, ad esempio, essendo basato sulla trasmissione in broadcast su un canale condiviso, fa viaggiare il traffico completamente in chiaro. Chiunque si trovi nello stesso ramo di rete può utilizzare un software di sniffing come Wireshark per leggere password e dati sensibili. Similmente, protocolli obsoleti come Telnet e FTP trasmettevano informazioni non cifrate, e sono stati fortunatamente sostituiti nel tempo da standard sicuri come SSH (Secure Shell) e SFTP, che instaurano tunnel crittografati. Per proteggere ulteriormente le comunicazioni su reti pubbliche e non sicure, l'utilizzo di una VPN (Virtual Private Network) diventa fondamentale, in quanto garantisce una cifratura end-to-end che nasconde il traffico a occhi indiscreti. In ambito aziendale e privato, l'implementazione di un firewall è una contromisura di base essenziale: si tratta di un sistema (hardware o software) che filtra il traffico di rete in entrata e in uscita in base a regole di sicurezza predefinite, bloccando le connessioni sospette.

Per difendersi adeguatamente, è necessario implementare un rigoroso processo di gestione del rischio, guidato da standard internazionali come ISO, NIST e dalle direttive AgID per la Pubblica Amministrazione.

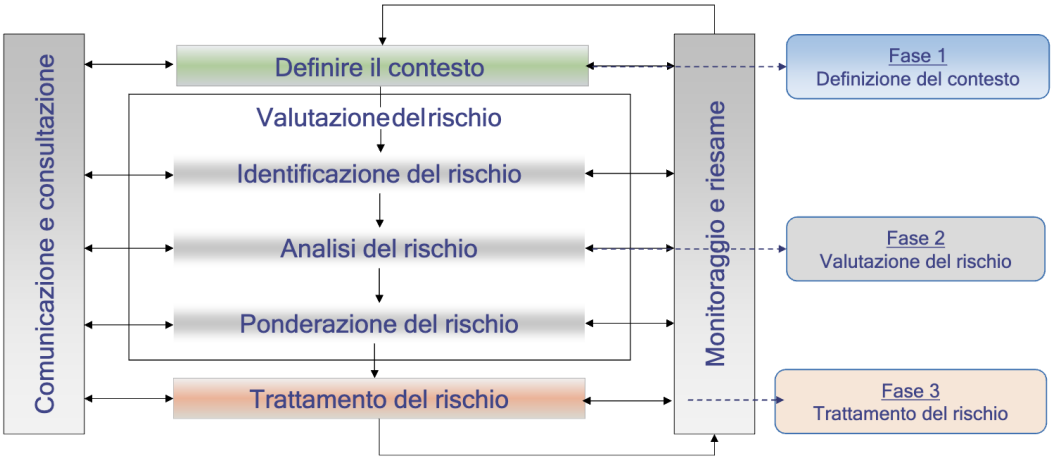

Figura 1: Framework di Cybersecurity Risk Management. Ciclo continuo e iterativo che comprende la definizione del contesto, la valutazione, il trattamento e il monitoraggio costante del rischio informatico.

Il rischio informatico, che non può mai essere azzerato, si calcola come il prodotto tra la probabilità che un attacco avvenga e l'impatto che esso avrebbe sul sistema. La probabilità dipende a sua volta dalla vulnerabilità del sistema e dal suo livello di esposizione verso l'esterno, mentre l'impatto si valuta misurando i danni arrecati alla riservatezza, all'integrità o alla disponibilità dei dati. Spesso le aziende sottovalutano questo rischio per questioni di budget, finendo per subire danni catastrofici. Gli attacchi informatici si dividono in due grandi categorie: quelli opportunistici e quelli mirati. Gli attacchi opportunistici sono casuali e automatizzati, caratterizzati da grandi volumi di traffico che scansionano la rete alla ricerca di porte aperte o dispositivi con password di default. Al contrario, gli attacchi mirati (o APT, Advanced Persistent Threat) sono estremamente silenziosi, mirano a bersagli specifici ad alto valore e iniziano sempre con una fase di OSINT (Open Source Intelligence), ovvero la raccolta metodica di informazioni pubbliche sull'infrastruttura e sui dipendenti. Sfruttando queste informazioni, gli attaccanti utilizzano tecniche di ingegneria sociale come lo spear phishing per ingannare le vittime e farle cadere in trappola. Una volta compromesso il sistema, gli obiettivi tipici includono l'esfiltrazione e la vendita di dati sensibili nel Dark Web, la cifratura dei file per richiedere un riscatto (Ransomware), o l'installazione di malware per trasformare la macchina in un bot all'interno di una rete distribuita per sferrare attacchi DDoS.

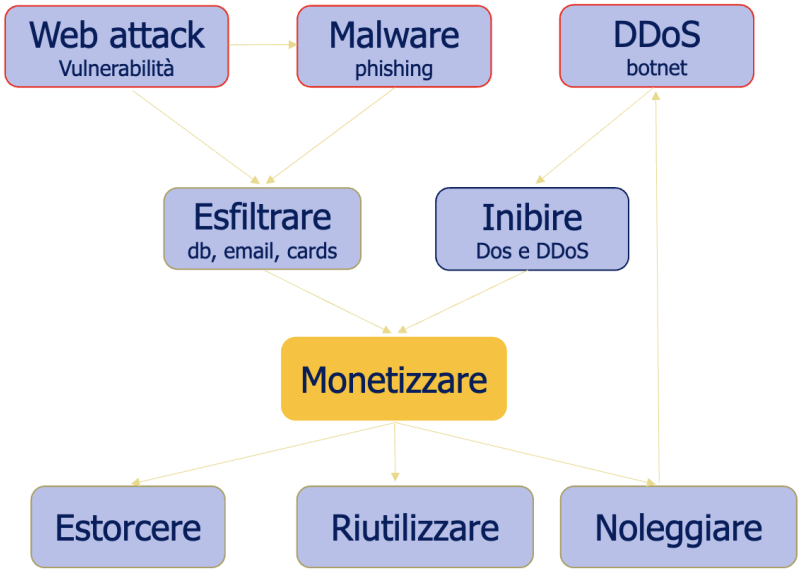

Figura 2: Flusso operativo della minaccia. Evoluzione tipica di un attacco: dai vettori di ingresso iniziali (es. Web attack o Malware tramite phishing) agli obiettivi intermedi (esfiltrazione o inibizione tramite DDoS), fino allo scopo finale di monetizzazione.

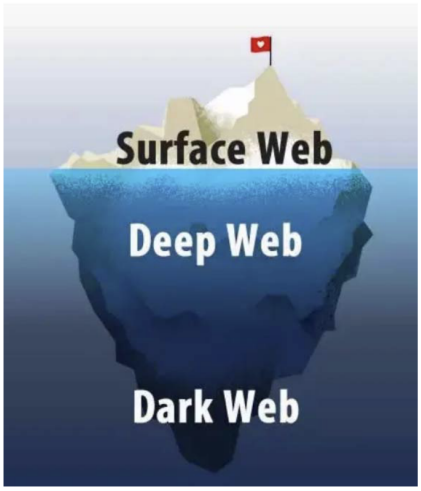

Figura 3: La struttura a iceberg del Web. Rappresentazione dei livelli Surface, Deep e Dark Web. Oggi i blackmarket per la compravendita di data leak (dati anagrafici, credenziali, carte di credito) si stanno spostando sempre più frequentemente dal Dark Web verso il Deep Web.

Analizzando il panorama attuale delle minacce, l'organizzazione OWASP evidenzia come le vulnerabilità delle applicazioni web siano il principale vettore di attacco. Sorprendentemente, le prime posizioni in classifica sono occupate da problematiche strettamente legate agli errori umani, come il Broken Access Control, che si verifica quando un utente riesce a forzare l'accesso a risorse riservate semplicemente manipolando il percorso di una URL, e le errate configurazioni di sicurezza da parte degli amministratori.

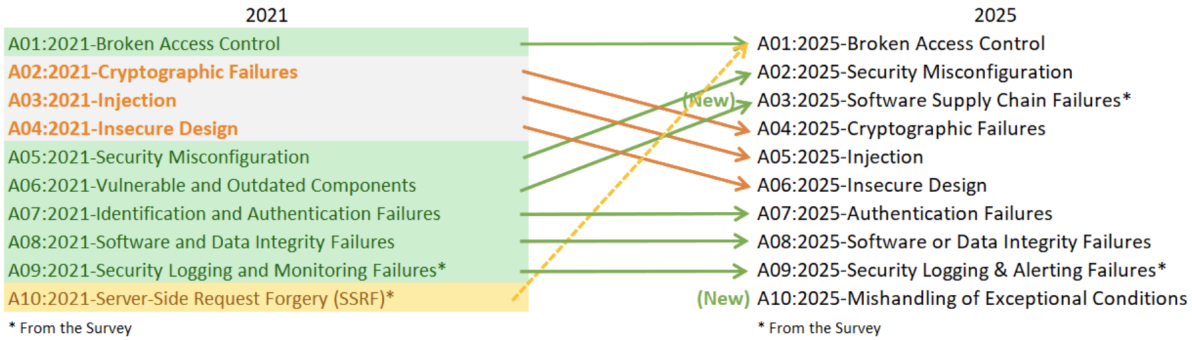

Figura 4: Classifica OWASP Top 10 (2021 vs 2025). Evoluzione delle principali vulnerabilità delle applicazioni web. Si nota in particolare l'impatto critico degli errori legati al fattore umano, come il *Broken Access Control* e la *Security Misconfiguration*.

Un classico esempio di quest'ultimo è lasciare aperta la visualizzazione dell'indice delle directory web, permettendo a chiunque di scaricare file sensibili come i backup dei database. Un altro grave pericolo deriva dall'utilizzo di componenti obsoleti: la storica vulnerabilità Heartbleed, ad esempio, sfruttava un bug nella libreria crittografica OpenSSL per sottrarre chiavi private e dati direttamente dalla memoria del server in modo del tutto invisibile.

Figura 5: Sensitive Data Exposure. Un caso pratico di vulnerabilità dovuta a un errore di configurazione: l'esposizione accidentale dell'indice di una directory riservata (`/admin/backup`), che consente il download non autorizzato di dump del database.

Un capitolo a parte meritano le cosiddette Injection e gli errori di progettazione del software. La SQL Injection è un attacco devastante che sfrutta la mancata validazione dell'input dell'utente. Se un server non filtra adeguatamente i dati inseriti in un modulo web, un attaccante può inserire caratteri speciali (come l'apice e il doppio trattino `' --`) per chiudere prematuramente la stringa legittima e commentare i controlli di sicurezza, manipolando di fatto la query eseguita sul database per estrarre tutti i record nascosti. Per prevenire questo problema, è necessario utilizzare strati software intermedi come le API e le query parametrizzate.

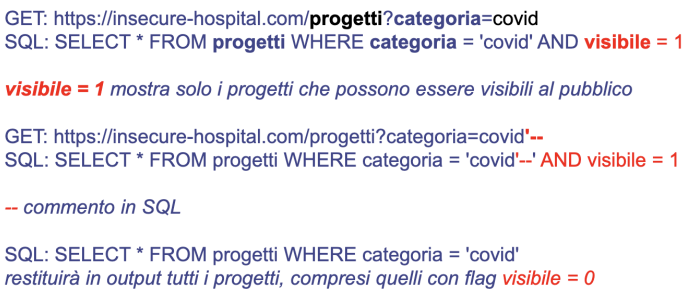

Figura 6: Dinamica di una SQL Injection. Alterazione di una query legittima tramite l'inserimento di input malevolo (es. l'uso del doppio trattino `'--'` per commentare il resto dell'istruzione). Ciò neutralizza le restrizioni di sicurezza (es. `visibile = 1`) e restituisce all'attaccante l'intera base dati.

Un'altra forma diffusa di iniezione di codice è il Cross-Site Scripting (XSS), tramite il quale si esegue codice JavaScript malevolo direttamente nel browser della vittima, permettendo di rubare identificatori di sessione e impersonare l'utente legittimo.

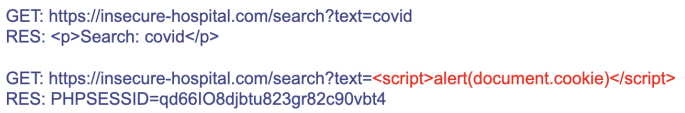

Figura 7: Cross-Site Scripting (XSS). Iniezione ed esecuzione lato client di codice JavaScript malevolo, finalizzata al furto dell'identificativo di sessione della vittima.

È importante in questo contesto chiarire il concetto di Client e Server, che il professore ha voluto sottolineare a lezione: essi non sono semplici macchine fisiche, bensì software. Un server è un programma che eroga un servizio, mentre un client è un programma che lo richiede; entrambi possono tranquillamente coesistere sullo stesso computer fisico comunicando tramite un'interfaccia di loopback locale, come accade quando si installa una suite come XAMPP per simulare un ambiente web offline. Infine, gli errori crittografici rappresentano una minaccia critica, soprattutto quando i dati vengono lasciati in chiaro a riposo. Mentre i protocolli come WPA e HTTPS cifrano i dati durante il transito, le password e le informazioni sensibili archiviate nei database devono essere obbligatoriamente cifrate o processate tramite funzioni di hash irreversibili, nonostante ciò comporti delle difficoltà tecniche per l'indicizzazione e l'ordinamento dei record all'interno dei database relazionali.