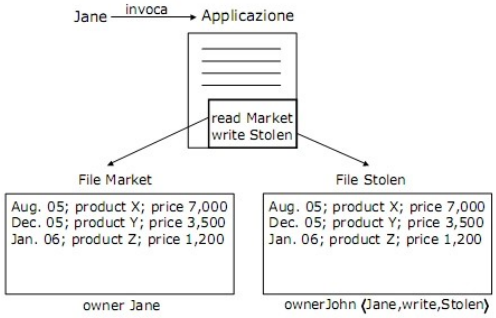
[4] Descrivere in sintesi le caratteristiche principali di un Trojan Horse e mostrare, tramite un esempio, perché e le politiche discrezionarie sono vulnerabili a questa tipologia di attacco. Descrivere inoltre perché tale Trojan Horse fallirebbe in presenza di una politica multilivello.

Un **Trojan Horse** è un programma di utilità che contiene del codice nascosto che esegue funzioni non legittime. Il problema è che le politiche discrezionarie controllano solo gli accessi diretti, non effettuano nessun controllo su cosa accade all'informazione una volta rilasciata. Quindi tutto ciò che l'applicazione esegue(quindi anche il codice nascosto) dipende dall'identità dell'utente che ha invocato l'applicazione e dai suoi permessi. Un trojan horse non può operare sul sistema se non è attivato dall'utente.

Esempio:



Nell'esempio le istruzioni readMarket e writeStolen sono legittime perché ci si basa sull'identità di chi ha lanciato l'applicazione, cioè Jane. Quindi John rispettando la politica discrezionaria riesce ad ottenere informazioni non legittime.

Un Trojan Horse sarebbe bloccato da una politica (di segretezza) mandatoria, visto che a differenza delle politiche discrezionarie che impongono delle regole nella forma soggetto-oggetto-azione, le mandatorie impongono delle restrizioni sul flusso di informazioni, operando una distinzione tra utenti e soggetti. I primi sono fidati mentre i secondi no. In particolare in una politica mandatoria basata sulla segretezza un Trojan non sarebbe in grado di svolgere il suo compito, visto che viene garantita la segretezza impedendo flussi di informazione verso classi di accesso più basse o non compatibile, attraverso due proprietà vincolanti la no write-down e la no read-up. Il covert channel in un contesto di questo tipo riesce però a funzionare.

[2] Descrivere le caratteristiche principali delle politiche discrezionarie. Perché sono politiche discrezionarie?

Le **politiche discrezionarie** controllano l'accesso sulla base dell'identità degli utenti e di regole che stabiliscono chi può o non può eseguire azioni sulle risorse. Prendono il nome dal fatto che ciascun utente può decidere a propria discrezione a chi passare i propri privilegi.
I tre aspetti che definiscono formalmente il sistema sono:

- *entità del sistema*
  - soggetti: entità che possono accedere alle risorse
  - oggetti: risorse che devono essere protette
  - azioni: operazioni che possono essere eseguite sugli oggetti dai soggetti
- *stato di autorizzazione/protezione*:
  insieme dei permessi concessi o negati ad ogni utente.
- *regole di identificazione*:
  assiomi che devono essere soddisfatti affinché i permessi siano concessi

Un primo modello che le rappresenta è il **modello a matrice di accesso** che ha come stato di protezione una tripla(S,O,A) dove S: soggetti; O: oggetti; A: matrice di accesso. Lo stato può essere modificato tramite comandi primitivi.
Il problema è che la matrice è troppo grande e quindi per la memorizzazione ci sono tre alternative:

1. **tabella di autorizzazione**: memorizza le triple(S,O,A) non nulle in una tabella a tre colonne.
2. **ACL**: memorizza per colonne. Ogni riga è una risorsa e ogni colonna è un soggetto che può interagire con la risorsa.
3. **Capability**: memorizza per righe. Ogni riga rappresenta un utente e ogni colonna è una risorsa con le relative

azioni che l'utente può eseguire su di essa.

Debolezze DAC:

- controllano solo gli accessi diretti alle informazioni
- vulnerabili ai trojan horse

ACL e Capability sono due diversi metodi di memorizzazione della matrice di accesso, quindi stiamo parlando di politiche discrezionarie.

- **ACL (access control list)**: memorizza per colonne. E' una lista dove ogni record è formato da un utente con le operazioni che possono essere eseguite sul file.
  In questo caso è obbligatoria l'autenticazione dell'utente perché ciò che può essere eseguito su un oggetto dipende dall'identità dell'utente.
- **Capability List**: memorizza per righe.

ACL

Vantaggi:

- più immediato controllo delle autorizzazioni presenti su un oggetto.
- più efficiente nel controllo dell'accesso e revoca relativi a oggetti in quanto le operazioni per oggetti sono considerate prevalenti.

Svantaggi:

- più complesso recuperare tutte le autorizzazioni di cui gode un soggetto in quanto richiede la ricerca di tali autorizzazioni per ciascun oggetto.
- richiede di autenticare sempre i soggetti.

Capability

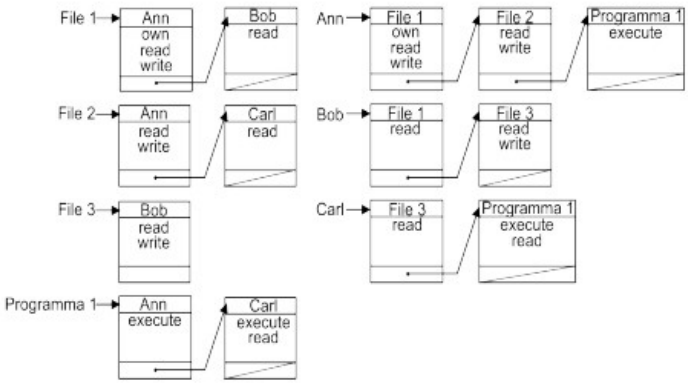
Vantaggi:

- più immediata determinazione dei privilegi di cui gode un soggetto.
- per un soggetto è sufficiente presentare la capability appropriata per guadagnare l'accesso ad un oggetto, questo è vantaggioso nei sistemi distribuiti in quanto permette di evitare le autenticazioni ripetute da parte di un soggetto.

Svantaggi:

- per recuperare tutte le autorizzazioni di accesso su un oggetto occorre esaminare tutte le capability.
- vulerabili all'attacco *forgery*: possono essere copiate e riutilizzate da terze parti che non godono delle autorizzazioni necessarie.

Nell'immagine abbiamo sulla sinistra ACL e sulla destra Capability:



[5] Descrivere la tecnica basata sull'analisi delle risorse del sistema e la tecnica basata sull'analisi del flusso di informazione per l'individuazione di covert channel. Si richiede di fornire degli esempi per entrambe le tecniche.

Un covert channel è un canale di comunicazione che permette ad un processo di trasferire informazioni che violano le politiche di sicurezza. Spesso si accompagna ad altri canali di comunicazione legittimi. Si può quindi pensare di modificare la lunghezza delle linee, inserire degli spazi in posizioni specifiche, al fine di codificare l'informazione che sarà quindi trasportata attraverso un report.

Si può altresi utilizzare uno storage channel che sfrutta la presenza o l'assenza di oggetti in memoria. Possiamo quindi segnalare un 1 o uno 0 al programma spia semplicemente imponendo un lock su un file condiviso, o creando un file talmente grande da esaurire la quota utente. A questo punto il programma spia proverà a scrivere un file se il SO negherà questo per esaurimento quota allora il programma di servizio voleva segnalare un 1 altrimenti si inferisce che il programma di servizio voleva segnalare uno 0.

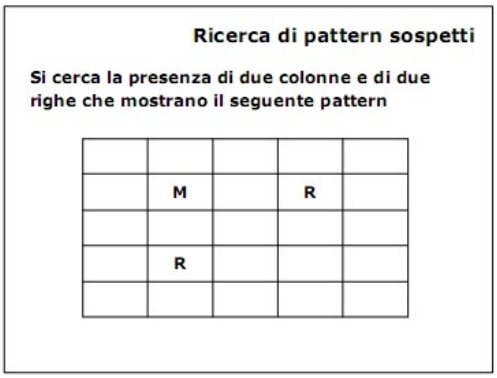
Allo stesso modo si potrebbe utilizzare un timing channel in cui il programma di servizio usa il proprio tempo di CPU per segnalare un 1 o lo ignora e passa avanti per segnalare uno 0.

Ovviamente tutti queste tipologie di canali necessitano di una cognizione di tempo condivisa e di risorse condivise per poter funzionare.

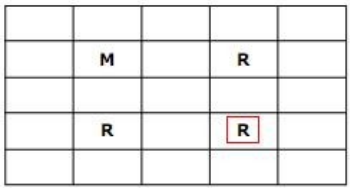
La tecnica basata sull'analisi delle risorse del sistema si basa sulla matrice delle risorse condivise e sfrutta il fatto che la base di ogni covert channel sono le risorse condivise. Si costruisce una matrice M dove le righe sono le risorse e le colonne sono i processi.

- M[i,j]: **R** se il processo j può leggere o osservare la risorsa i
- M[i,j]: **M** se il processo j può creare o modificare o cancellare la risorsa i

Esempio:



E' un pattern sospetto perché la risorsa 2 è condivisa dal processo 2 e dal processo 4(modificata da uno e letta dall'altro). Il processo 2 legge il contenuto della risorsa 4 e la trasmette alla risorsa 2, il processo 4 quindi può leggere indirettamente l'informazione della risorsa 4, cosa che direttamente non potrebbe fare.
Bisogna completare la matrice con il flusso potenziale di informazione e cercare eventuali flussi indesiderati, in questo caso:



L'analisi del flusso di informazioni, invece, cerca di scoprire flussi di informazione non ovvi tra le istruzioni di un programma. Ad esempio:

- B := A realizza un flusso di informazione da A a B (flusso esplicito)
- If D=1 then B:=A realizza due flussi, da A a B ma anche da D a B (flusso implicito)
- B := function(args) realizza un flusso di informazione dalla funzione function a B

[4] Nell'ambito del controllo dell'accesso, definire il concetto di gruppo e ruolo, evidenziando quali sono le differenze tra questi due concetti.

Gruppi e ruoli rappresentano lo stesso concetto, ma sono differenti:

- **Gruppi**: insieme di utenti(statici). Statici significa che se un utente fa parte del gruppo, allora può sempre eseguire le operazioni concesse, senza dover attivare nessun ruolo. Quindi la definizione di gruppo non rimane invariata, si possono aggiungere altri utenti.
- **Ruoli**: insieme di privilegi(dinamici). Dinamici significa che i privilegi dipendono dal ruolo che l'utente ha

attivato.

[3] Nell'ambito del modello di Biba, descrivere la politica low-water mark per soggetti. Questa politica garantisce l'integrità delle informazioni? Si richiede di giustificare la risposta.

Politica **low-water mark per soggetti**:

- il soggetto S può scrivere un oggetto O se e solo se: $\lambda(s) > \lambda(o)$
- il soggetto S può leggere qualsiasi oggetto, però questo va a modificare la sua affidabilità: $\lambda(s) = GLB\{\lambda(s),\lambda(o)\}$

Svantaggio: l'ordine delle operazioni modifica i privilegi del soggetto. Infatti abbiamo visto che il soggetto può leggere qualsiasi cosa, però questo va a modificare la sua classe d'integrità, cioè la sua affidabilità. Quindi può capitare che un soggetto S, che prima poteva scrivere un determinato oggetto O, non sia più in grado di farlo dopo aver letto un altro oggetto che ha fatto diminuire la sua classe di integrità.

Il modello di Biba è un modello per la politica mandatoria per l'integrità.

[2] Descrivere la politica low-water mark per soggetti e per oggetti ed i loro svantaggi.

Secondo la politica **low-water mark per oggetti**:

- un soggetto *s* può leggere un oggetto *o* se e solo se $\lambda(o) \geq \lambda(s)$
- un soggetto *s* può scrivere ogni oggetto *o*, ma dopo l'accesso $\lambda(o) = glb\{\lambda(s) , \lambda(o)\}$

Lo **svantaggio** è che non protegge l'integrità, si limita a segnalare che è stata compromessa.
Questa politica non garantisce l'integrità delle informazioni in quanto controlla solo violazioni di integrità dovute a flussi impropri, mentre l'integrità è un concetto più ampio.

[1] Nell'ambito del modello di Bell e LaPadula, fornire la definizione di simple security property e *-security property.

Secondo il modello, uno stato (b, M, λ) è sicuro se e solo se soddisfa i seguenti criteri(ovvero proprietà che che esprimono i limiti imposti dalla politica mandatoria):

- **Simple property**: uno stato v soddisfa la "simple property" se per ogni $s \in S$, $o \in O$: $(s,o,read) \in b \rightarrow \lambda(s) \geq \lambda(o)$. Corrisponde al principio no-read-up;
- ***-property**(Star property): uno stato v soddisfa la "*-property" se per ogni $s \in S$, $o \in O$: $(s,o,write) \in b \rightarrow \lambda(o) \geq \lambda(s)$. Corrisponde al principio no-write-down.

[1] Nell'ambito del meccanismo di controllo dell'accesso di Apache, descrivere il significato della direttiva Order per le tre diverse tipologie di ordinamento.

La direttiva order imposta l'ordine di valutazione di allow e deny

- order deny, allow = prima valuto le deny, poi le allow => realizzo di fatto una politica aperta
- order allow, deny = prima allow, poi deny => politica chiusa
- order mutual-failure = una richieste deve NON soddisfare deny, e SODDISFARE allow: in pratica non sono ammessi conflitti, se ci sono non va bene e non permetto l'accesso.

[3] Nell'ambito delle politiche per il controllo dell'accesso basato sui ruoli, dire cosa si intende con principio del minimo privilegio (Least privilege). Si richiede di fornire un esempio.

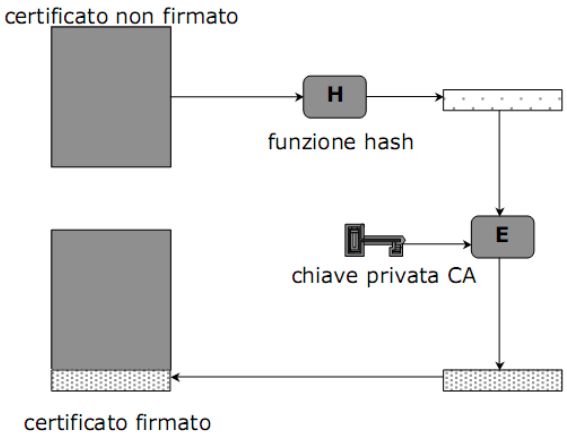
Il principio del minimo privilegio(**Least Privilege**) consiste nel fatto che non bisogna concedere troppi privilegi ma associare il numero minimo indispensabile di privilegi per compiere determinate operazioni in modo efficiente. Si contrastano così l'occorrenza di abusi danni o violazioni .

[2] Dire cosa si intende per certificato digitale e come esso viene creato.

Un certificato digitale è un documento elettronico che attesta, con una firma digitale, l'associazione tra una chiave pubblica e l'identità del soggetto. Esso è garantito da una terza parte fidata nota col nome di Certification Authority (per gli amici CA). Essa rilascia un certificato digitale che contiene le informazioni dell'utente per cui rilascia il certificato, oltre alla chiave pubblica.
Il certificato lega il nome di un soggetto ad una chiave pubblica e per verificarne l'autenticità basta confrontarla con la chiave pubblica della CA.

La creazione:

- Utente prova la propria identità alla CA (procedure non crittografiche)
- CA verifica l'autenticità della chiave
  - coppia chiavi pubblica/privata generata dalla CA
  - coppia chiavi pubblica/privata generata dall'utente
- CA crea un certificato digitale che contiene la chiave pubblica ed i dati identificativi dell'utente
- CA firma il certificato digitale con la propria chiave privata



[2] Nell'ambito del modello a matrice di accesso, dare la definizione di transizione di stato ed enunciare in modo preciso e formale il problema della safety.

[1] Nell'ambito del modello a matrice di accesso, fornire la definizione di stato di protezione e descrivere i comandi primitivi che possono essere usati per cambiare lo stato.

Lo stato di protezione è definito dalla **tripla (s,o,a)** dove:

- **S: insieme dei soggetti** (righe della matrice)
- **O: insieme degli oggetti** (colonne della matrice)
- A: matrice di accesso formata dai soggetti e dagli oggetti e dove **A[s,o]** indica le azioni del soggetto *s* sull'oggetto *o*.

Lo stato di protezione può essere modificato tramite dei **comandi primitivi**:

- enter r into A[s,o]
- delete r from A[s,o]
- create subject S'
- destroy subject S'
- create object O'
- destroy object O'

I comandi hanno la seguente **sintassi**:

```
command c(x1, . . . , xk)
        if r1 in A[xs1, xo1] and r2 in A[xs2, xo2]
           and ... rm in A[xsm, xom]
        then c1
             c2
             ...
             cn
end
```

```
command create(subj,file)
        create object file
        enter own into A[subj,file]
end
```

- $r_1,...,r_m$: azioni
- $s_1,...s_m$ e $o_1,...,o_m$: interi tra 1 e k. Se m=0 il comando non ha la parte condizionale
- $c_1,...c_n$: comandi primitivi

Descrivere a quale possibile attacco è vulnerabile la tecnica di autenticazione basata sul meccanismo di challenge-response. La risposta deve essere ben motivata fornendo anche un esempio.

challenge-response: è una "sfida" tra server e utente basata sul protocollo *challenge response handshake* e caratterizzata dalle seguenti fasi:

1. il server di autenticazione stabilisce una sfida (challenge)
2. il token genera la risposta (diversa per ogni sfida)
3. l'autenticazione avviene con successo se la risposta è corretta

Il challenge response è resistente contro replay attack passivi, ma vulnerabile rispetto ad attacchi attivi come il man in the middle attack.

In questo scenario l'avversario si dichiara al posto del token nei confronti del server e si mostra come server nei confronti del token, questo gli permette di autenticarsi utilizzando la risposta corretta generata direttamente dal token.
Immagine chiarificatrice:

dove **R** = response del token al server
C = challenge del server al token
e l'avversario è ovviamente Babbo Natale.

[1] Descrivere a cosa serve la signature di un virus.

La **signature** di un virus è una stringa di bit usata per individuare ed identificare un virus in modo univoco. Gli antivirus sfruttano questa stringa di bit che caratterizza un particolare tipo di codice malizioso per trovare tale virus e talvolta eliminarlo.
La signature si basa su tre tipi di pattern:

- *di memorizzazione*: quando il codice del virus inizia con una sequenza di bit invariante. La parte di codice non deve essere necessariamente all'inizio, ma anche in altre posizioni però sempre nella stessa posizione. Potrebbe essere anche un programma il cui codice comincia con un'istruzione di jump, saltando subito altrove prima di definire le variabili.
- *di esecuzione*: legato a ciò che il virus esegue, infatti bisogna osservare il comportamento del virus una volta eseguito, in particolare l'entità e la tipologia di danni provocati.
- *di trasmissione*: si basa sul presupposto che un virus adotti sempre la stessa metodologia di diffusione.

[3] Dire cosa si intende per salami attack e fornire un esempio.

Il Salami Attack consiste in un insieme di piccoli crimini informatici che uniti danno luogo a crimini significativi. Parte da qualcosa di insignificante che però applicato molte volte porta a qualcosa di significativo.
Esempio: piccole somme di denaro che vengono trasferite su di un conto(quello dell'attaccante ovviamente). Queste somme prese singolarmente sono talmente piccole che non vengono considerate, passano inosservate e possono essere ad esempio quelle calcolate per gli interessi del conto bancario.
Il Salami Attack solitamente è causato dalle correzioni che i programmatori inseriscono nei programmi per correggere gli inevitabili errori di calcolo.

[2] Dire cosa si intende per segretezza, integrità e disponibilità.

- **segretezza**: si parla di segretezza quando un'informazione viene rilasciata, direttamente o indirettamente solo a utenti autorizzati a riceverla.
  La *privacy* è una specializzazione della segretezza. Se ne parla quando l'informazione fa riferimento a individui e consiste nel diritto dell'individuo di stabilire se, come, quando e a chi l'informazione che lo riguarda può essere rilasciata.
- **integrità**: si parla di integrità quando le informazioni e le risorse non vengono modificate, cancellate o distrutte in modo non autorizzato o improprio.
  Include due aspetti fondamentali: correttezza e fiducia.
- **disponibilità**: si parla di disponibilità quando non vengono in alcun modo impediti agli utenti gli accessi propri e per i quali hanno l'autorizzazione necessaria.

[7] Spiegare in sintesi l'attacco SQL injection fornendo un esempio e quali contromisure possono essere adottate.

L'**SQL injection** è un attacco rivolto alle applicazioni web e coinvolge non solo SQL, ma qualsiasi linguaggio di programmazione e qualsiasi DBMS.
L'attacco consiste nell'inserimento di codice maligno nelle query SQL, sfruttando la mancanza di controlli sui dati da input dell'applicazione web.
L'input può essere trasmesso in tre modi:

- URL (query string)
- form HTML
- cookie costruito su misura

L'attacco provoca:

- manipolazione indesiderata dei dati
- accesso ad aree riservate
- visualizzazione di dati riservati

Esempio:
Abbiamo una variabile *$id* presa in input dalla query string, teoricamente di tipo intero, ma non validata.
$sql = "SELECT * from articoli WHERE id=$id";
se un attaccante in *$id* inserisce
$id = 1; DROP table articoli; la query diventerà:
*$sql = "SELECT * from articoli WHERE id = 1; DROP table articoli;*
che ovviamente provoca la cancellazione della tabella articoli.

Contromisure possibili sono:

- *controlli sul tipo di dato*: tramite l'utilizzo di alcune funzioni si forza una variabile ad appartenere ad un certo tipo.
- *creazione di filtri tramite espressioni regolari*: i dati in input vengono descritti da una espressione regolare. Ad esempio controllare che l'input sia formato solo dalle lettere dalla A alla Z.
- *eliminazione di caratteri potenzialmente dannosi*: si eliminano i caratteri che hanno un significato in un'interrogazione SQL, quindi **, ; "** '
- ***escape(\) di caratteri potenzialmente dannosi*: il carattere sul quale viene effettuato il quoting (\) viene interpretato letteralmente.**

[1]Nell'ambito delle tipologie di crimini informatici, dire cosa si intende per sabotaggio e descrivere come questo attacco può essere classificato.

Per **sabotaggio** si intende qualsiasi azione o iniziativa di disturbo o di danneggiamento intesa ad ostacolare il normale funzionamento del sistema.
Questo attacco può essere classificato in:

- *sabotaggio fisico*: consiste in un vero e proprio danneggiamento fisico
- *sabotaggio logico*: consiste nella modifica o nella distruzione dell'informazione al fine di sabotare il normale funzionamento del sistema
- *sabotaggio psicologico*: consiste nel logoramento psicologico dell'utente.

[1] Descrivere il funzionamento di un virus boot sector.

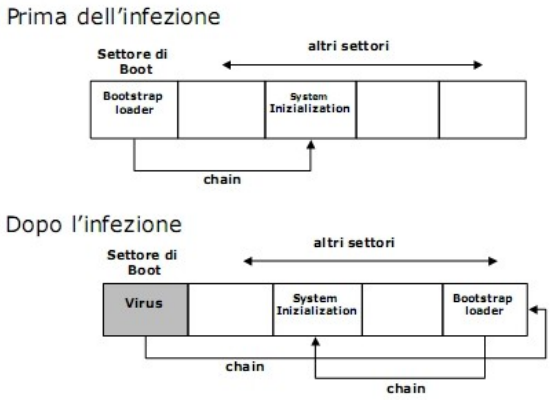
Un virus **boot sector** non può essere individuato da normali software antivirus perché entra in esecuzione prima dell'avvio del sistema e quindi prima degli stessi programmi antivirus.

Normale procedura di bootstrap senza infezione:

- accensione del computer
- test dei componenti hardware
- lettura del boot sector da hard disk
- caricamento del bootstrap loader che carica il sistema operativo

Procedura bootstrap con infezione:

- accensione del computer
- test dei componenti hardware
- esecuzione del virus
- caricamento del bootstrap loader che carica il sistema operativo col virus già in esecuzione



[1] Nell'ambito delle tecniche di autenticazione, descrivere l'autenticazione basata sul possesso.

L'**autenticazione basata sul possesso** si basa appunto sul possesso da parte dell'utente di un Token, cioè qualcosa di tangibile che l'utente utilizza per identificarsi. Ogni token ha una chiave crittografica usata per dimostrare l'identità del token al computer.

- Vantaggi: i token sono più sicuri delle password, mantenendo il controllo sul token l'utente mantiene controllo sull'utilizzo della sua identità.
  Si può usare questo tipo di autenticazione combinata con la Conoscenza, quindi servono token e password per autenticarsi.
- Svantaggi: i token possono essere persi, rubati, falsificati.
  Chiunque acquisisca un token può impersonare l'utente.

I token più utilizzati sono:

- **memory card**: hanno memoria, ma non capacità di processo. Sono usate con il PIN che però non possono controllare e nemmeno crittare per la trasmissione(quindi il PIN viaggia in chiaro). Sono vulnerabili quindi ad attacchi di sniffing.
- **smart token: dispositivi che godono di capacità di processo il cui funzionamento si basa su meccanismi statici(l'utente si autentica al token e il token si autentica al sistema) e dinamici(il token genera periodicamente nuove chiavi)**

[1] Descrivere come viene gestito il processo di revoca dei certificati in X.509.

Lo standard **X.509** supporta due principali sistemi di revoca:

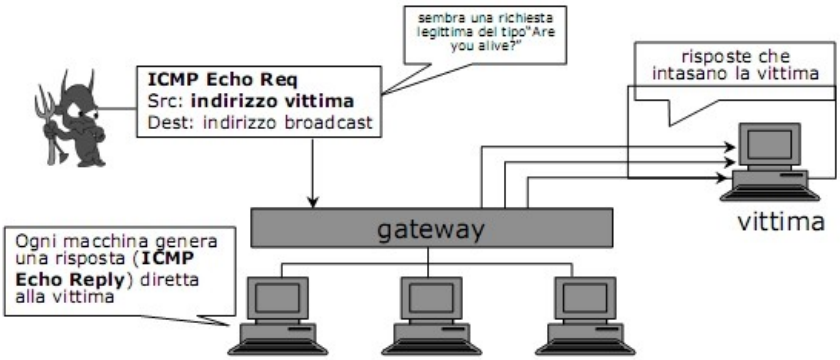
- data di validità
- revoca esplicita

Il CRL è una lista di certificati revocati che viene sempre controllata prima di considerare un certificato valido.
Per ogni certificato revocato il CRL contiene le seguenti informazioni:

- issuer
- last update date
- next update date
- firma CA
- lista dei numeri di serie dei certificati revocati con data di revoca

[5] Descrivere lo smurf attack, come funziona e possibili contromisure.

Nello smurf attack l'attaccante intasa la vittima sfruttando il protocollo ICMP. Infatti invia una ICMP Echo Req con *fonte*: indirizzo vittima e *destinazione*: indirizzo broadcast(quindi tutti gli host della rete locale). Quindi tutte le macchine che ricevono la richiesta generano una ICMP Echo Reply diretta alla vittima, così facendo le risposte generate intaseranno la vittima.
Slide chiarificatrice:

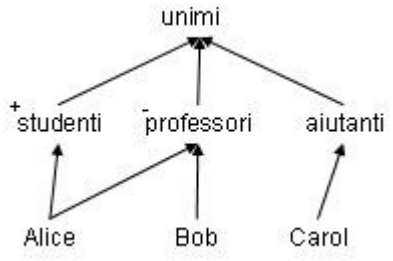


[2] Nell'ambito delle politiche di risoluzione dei conflitti, descrivere le politiche most-specific-takes-precedence (mstp) e most-specific-along-a-path-takes-precedence (msaptp). Si richiede inoltre di fornire un unico esempio di gerarchia utenti-gruppi con relative autorizzazioni e che illustri che per l'utente Alice non c'`e una autorizzazione vincente (deve rimane un conflitto +/ –) sia quando si applica la politica mstp sia quando si applica la politica msaptp.

La politica mstp(most specific takes precedence) comporta che vince l'autorizzazione più specifica rispetto a tutto lo schema. La politica msaptp(most specific along a path takes precedence) invece comporta che vince l'autorizzazione più

specifica rispetto ad un cammino. L'autorizzazione più specifica vince sui cammini che passano dall'autorizzazione.

Esempio utenti-gruppi:



[1] Nell'ambito della classificazione presentata a lezione e relativa ai crimini informatici intenzionali, descrivere cosa si intende per ricerca (fisica e logica) fraudolenta di dati.

Per **ricerca fraudolenta di dati** si intende l'acquisizione di informazioni che si trovano in un elaboratore oppure rilasciate al termine dell'esecuzione di una elaborazione.

- **Fisica**: si cercano informazioni memorizzate su materiale tangibile. (es: listato)
- **Logica**: si cercano informazioni residue nella memoria dell'elaboratore al termine dell'esecuzione di una procedura

[2] Descrivere le informazioni che un firewall di tipo packet filtering può analizzare per filtrare i pacchetti. Questi firewall sono in grado di tracciare delle correlazioni tra i pacchetti di una trasmissione?

Un firewall di tipo packet filtering analizza i pacchetti solo sulla base delle informazioni dell'header:

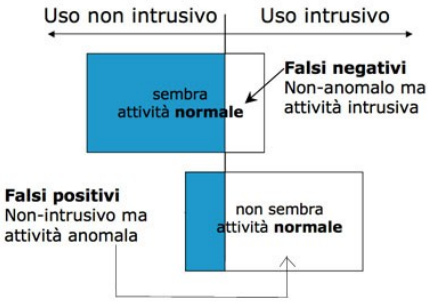
- indirizzo sorgente
- indirizzo destinazione
- porta sorgente
- porta destinazione
- tipo di protocollo
- opzioni di protocollo

Il *packet filtering* non è in grado di tracciare delle correlazioni tra i pacchetti di una trasmissione, mentre lo *stateful packet filtering* si.

[2] Nell'ambito dei sistemi per il controllo delle intrusioni, descrivere le caratteristiche principali dei sistemi misuse detection e anomaly detection.

Gli IDS si possono classificare in due categorie per i metodi utilizzati:

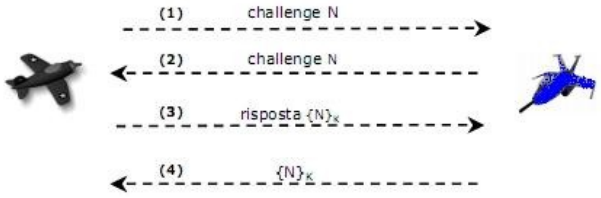
1. **Anomaly detection:** verificano le possibili intrusioni con conoscenze a posteriorio, ovvero ogni utente possiede un profilo dove sono memorizzate le operazioni che generalmente compie. Si riscontra un'anomalia e dunque una possibile intrusione quando un utente compie delle operazioni non abituali (magari effettuate da un intruso con l'identità dell'utente in questione). Questi tipi di IDS sono in grado di riconoscere nuovi tipi di attacco. Lo svantaggio principale degli Anomaly detection è che in alcuni casi la creazione di profili non è banale, in quanto bisogna anche scegliere che tipo azioni bisogna misurare per poi riscontrare azioni anomale.
2. **Misuse detection:** confronta gli eventi con degli schemi predefiniti di attacco, se riscontra delle operazioni che corrispondono a violazioni emette un particolare avviso. Conoscenza a *priori*. I problemi principali del Misuse detection è che bisgona gestire la conoscenza degli attacchi che si vogliono inserire negli schemi, dunque problemi legati a queste attività possono essere anche l'aggiornamento degli attacchi conosciuti. Un altro problema è che una descrizione potrebbe corrispondere a più attacchi.



Come si può vedere dall'immagine esistono casi che in cui si riscontrano **Falsi Negativi**, ovvero casi in cui un'analisi intrusiva non riscontra la presenza di un attaccante. Questi casi sono ovviamente più pericolosi dei **Falsi Positivi**, casi in cui l'analisi potrebbe riscontrare la presenza di un intruso, che in realtà non c'è. Nei Falsi Positivi si hanno comunque delle conseguenze non desiderate, come interruzioni dei servizi che non sono necessarie, ma che comunque sono migliori delle conseguenze che si avrebbero nei Falsi Negativi dove a nostra insaputa è presente un attaccante nel sistema.

[1] Nell'ambito della tecnica di autenticazione challenge-response, descrivere l'attacco denominato reflection attack ed una possibile contromisura.

Il **Reflection Attack** è un attacco che consiste nell'aggirare il problema della challenge riproponendola all'avversario(rimandandogliela indietro), che ci fornirà la risposta corretta alla challenge che quindi rinvieremo indietro.

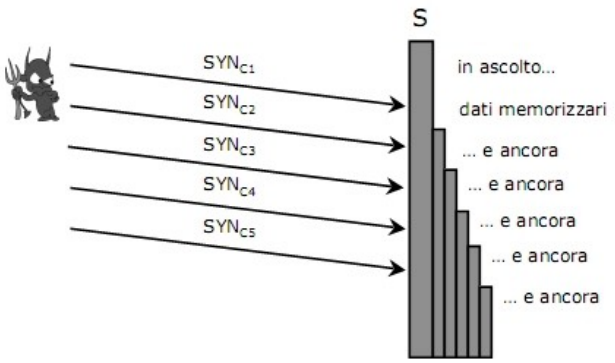


L'aereo blu fornisce la risposta corretta!!

Contromisura: per evitare questo attacco basta inserire insieme alla challenge anche l'identificatore dell'aereo, così che l'avversario non possa rinviarla indietro uguale.

[2] Si richiede di descrivere l'attacco denominato SYN flooding.

Il **SYN flooding** è un attacco nel quale l'attaccante genera un flusso di pacchetti con il flag SYN attivo e con l'indirizzo IP spoofato, per non rivelare il vero indirizzo IP da cui sferra l'attacco e per rendere i pacchetti SYN indistinguibili dai legittimi pacchetti SYN.
Con questo tipo di attacco la coda delle connessioni half-opened dello stack TCP/IP viene saturata, quindi la vittima non accetterà ulteriori connessioni, anche legittime.



Gli stack moderni sono resistenti a questo tipo di attacco.
SYN COOKIE: il server prepara un cookie la cui integrità è garantita attraverso meccanismi di firma, con informazioni relative alla connessione, che rispedisce al client. In questo modo l'allocazione delle risorse viene posposta finché non si riceve il terzo pacchetto (ACK).

[1] Dire cosa si intende per trapdoor

trapdoor: punti di accesso non documentati ad un modulo o ad un programma. Potrebbero essere delle stringhe di codice inserite nella fase di test del programma e poi dimenticate, oppure lasciate volutamente dai programmatori per poter accedere al programma dopo la sua attivazione.
Queste trapdoor sono delle vulnerabilità perché espongono il sistema a modifiche durante il funzionamento oppure possono essere usate come canali nascosti di accesso.
Un esempio di trapdoor è un controllo degli errori povero, che potrebbe causare un attacco di SQL-injection.

[1] Dire in cosa consiste una bomba logica.

Bomba logica: è un programma o parte di programma che viene eseguito al verificarsi di determinate condizioni. La sua esecuzione determina condizioni o stati che facilitano la realizzazione di atti non autorizzati. La bomba logica è il programma contenente la condizione che, se verificata, farà esplodere la bomba.

Si richiede di descrivere cosa si intende per Certificate Revocation List (CRL), di descrivere i principali campi di cui si compone e di indicare quando queste liste devono essere controllate.

Nell'ambito degli Intrusion Detection System, dire cosa si intende per falsi positivi e falsi negativi.

Nell'ambito dei sistemi operativi Unix-based, descrivere il significato dei privilegi setuid e setgid.
Illustrare inoltre come tali privilegi possano consentire ad un utente di modificare la propria password.

[2] In relazione al problema della sicurezza delle password, dire cosa si intende per: spoofing; snooping; sniffing; masqueranding.

- **spoofing**: password acquisite da terze parti che impersonano l'interfaccia di login. Avremo quindi una interfaccia di login fasulla creata dall'attaccante.
- **snooping**: password acquisite da persone che osservano gli utenti legittimi mentre le scrivono.
- **sniffing**: password acquisite da terze parti nella comunicazione lungo la rete.
- **masquerading**: quando un utente in possesso di password altrui impersonifica gli altri utenti.

[3] Nell'ambito dei sistemi operativi Unix-based, descrivere il significato dei privilegi setuid e setgid.

Setuid e Setgid sono privilegi addizionali. Setuid significa che l'UID del processo che esegue il file coincide con l'UID del file.
Setgid significa che il GID del processo che esegue il file coincide con il GID del file.
Entrambi si rappresentano tramite una **s** se l'utente(es: rws --- ---) o il gruppo(es: rwx rws ---) hanno privilegio di esecuzione sul file, tramite una **S** se l'utente(es: r-S --- ---) o il gruppo(es: rwx r-S ---) non hanno privilegio di esecuzione sul file.

[2] Illustrare in modo dettagliato e preciso come la stringa di formato %n pu`o essere utilizzata per sovrascrivere l'indirizzo RET nello stack.

La vulnerabilità delle stringhe di formato consiste nel rischio che, non utilizzando le specifiche di conversione e quindi non validando l'input dell'utente, esso possa inserire dei parametri che vengono poi valutati. Ciò consente gli attacchi di overflow.

L'attaccante inserisce il codice maligno, il programma sovrascrive sulla printf.
Avviene una sovrascrittura dell'indirizzo di ritorno con un indirizzo significativo che non porta più all'applicazione originale, ma all'applicazione dell'attaccante, al codice maligno.

[1] Dare la definizione di virus crittografico e descrivere le tre parti principali di cui tale tipo di virus solitamente si compone.

Un virus crittografico è un virus polimorfo che sfrutta tecniche crittografiche per cambiare forma.
Le tre parti di cui è composto sono:

1. chiave di decrittazione: varia da infezione a infezione
2. codice crittato
3. routine per la decrittazione del codice: questa è la parte costante del virus e per questo può essere considerata la sua *signature*.

Nell'ambito del sistema Kerberos, dire quali sono i tre compiti principali dell'Authentication Server (AS) ed i tre compiti principali del Ticket Granting Server (TGS).

Descrivere le caratteristiche base degli IDS di tipo anomaly detection e misuse detection elencando i vantaggi e svantaggi di ognuno.

Descrivere il processo di autenticazione e segretezza di PGP.

Nell'ambito del modello di Bell e LaPadula, descrivere la tranquility property e fare un esempio del perché è necessario introdurre tale proprietà.

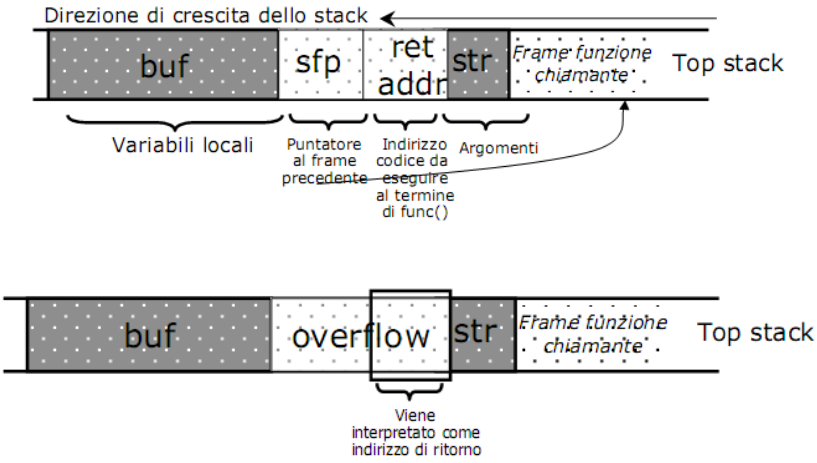
Descrivere il buffer overflow illustrando anche un esempio.

Il **buffer** è un area di memoria che serve per memorizzare temporaneamente dei dati. Nello **stack** essa ha una certa dimensione assegnata, ma confina con l'area destinata all'indirizzo di ritorno, ovvero l'indirizzo della funzione che ha chiamato la parte di codice proprietaria di questo stack.

Il **buffer overflow** sfrutta una caratteristica di diverse funzioni che utilizzano buffer per immagazzinare dati. Esse si aspettano una certa quantità di dati in ingresso. E quelle scritte male, anche se ne arrivano di più, non fanno nessuna storia. Ma il sistema ha assegnato al buffer una quantità fissa di spazio. Se arriva più roba, dove la metto? Semplice: sovrascrivo le aree dello stack circostanti! Con un po' di abilità e fortuna, posso inviare dati maliziosi i quali sovrascrivano allegramente l'indirizzo di ritorno, mettendoci al suo posto l'indirizzo del mio codice malizioso, o comunque dati che possono inficiare il buon funzionamento del programma. Per metterci un altro indirizzo, occorre però sapere esattamente dove in memoria il mio codice si trovi, e non è affatto una cosa banale. Posso anche modificare i valori di ritorno, causando chissà quali danni. O mettere addirittura il codice eseguibile nello stack, passandolo come

parametro.

Il buffer overflow dipende quindi dall'implementazione delle funzioni in un particolare linguaggio di programmazione. Per evitare debolezze del genere, si possono usare linguaggi come Java, che sono sicuri da questo punto di vista, o stabilire che lo stack non sia eseguibile.



[2] Definire la politica aperta e la politica chiusa. Indicare per quale ragione si considerano le autorizzazioni negative ed i problemi che queste portano.

- **politica aperta**: autorizzazioni specificano negazioni all'accesso, quindi il soggetto può fare tutto ciò che eredita dalla sua gerarchia a meno di eccezioni.
- **politica chiusa**: autorizzazioni specificano permessi di accesso, quindi ogni permesso viene negato a meno di eccezioni.

Si usano principalmente le politiche chiuse perché negano l'accesso per default e quindi assicurano una maggiore protezione.
Il problema si viene a creare quando ad esempio un'organizzazione vuole permettere l'accesso ai propri 100 membri, tranne 10; significa che sarà necessario specificare 90 autorizzazioni positive, cioè permessi concessi. In questo caso sarebbe ovviamente più comoda una politica aperta.

[4] Nell'ambito del controllo dell'accesso, a cosa servono le politiche amministrative? Si richiede inoltre di descrivere almeno tre diverse politiche amministrative.

Le **politiche amministrative** definiscono a chi spetta il compito di concedere e/o revocare le autorizzazioni. Le autorizzazioni sono determinate dalle classificazioni assegnate a soggetti e oggetti del sistema:

- classificazioni dei soggetti sono determinate dall'amministratore di sicurezza
- classi degli oggetti sono determinate dal sistema in base alle classi dei soggetti che li hanno creati.

L'amministratore di sicurezze è l'unico che può cambiare le classi degli oggetti e dei soggetti.
I diversi tipi di politiche amministrative sono:

- **amministrazione centralizzata**: un singolo amministratore ha il controllo su tutto il sistema
- **amministrazione gerarchica**: un singolo amministratore è responsabile per assegnare responsabilità amministrative ad altri
- **amministrazione cooperativa**: diversi amministratori cooperano nella definizione delle autorizzazioni
- **ownership**: ogni oggetto ha un proprietario che lo amministra
- **amministrazione decentralizzata**: amministrazione suddivisa fra più utenti. Spesso associata all'ownership. Flessibile, ma con diverse complicazioni.

[1] Spiegare in sintesi quali conseguenze per la sicurezza pu`o implicare una erronea validazione dei dati di input di una applicazione Web.

L'**erronea validazione** dei dati di input può comportare il Cross-site scripting(XSS) e l'SQL injection. L'erronea validazione si verifica quando i siti web si fidano dei dati che arrivano dall'esterno e non li controllano, anzi spesso li

visualizzano. Ciò provoca:

- *XSS*: attacco che permette ad un malintenzionato di inserire codice arbitrario come input in un'applicazione web, consentendogli di raccogliere dati, leggere cookie, visualizzare falsa pubblicità.
- *SQL injection*: attacco che consiste nella manipolazione di interrogazioni SQL costruite sulla base di input passato da un utente(URL, form HTML, cookie ad hoc). Comporta la manipolazione indesiderata dei dati, accesso ad aree riservate, visualizzazione dati riservati.

[2] Nell'ambito degli attacchi a protocolli di rete, descrivere in maniera chiara e sintetica in cosa consiste un attacco di tipo Denial-of-Service (DoS). Quali sono le differenze tra DoS e DDoS (Distributed Denial-of-Service)?

Un attacco **DoS (Denial of Service)** ha come obiettivo quello di far saltare il servizio offerto da un sistema, quindi vedranno negato l'accesso al sistema(alle informazioni e ai servizi che offre) anche utenti autorizzati.
Un attacco di questo tipo viene realizzato inondando di richieste casuali la macchina obiettivo che non riuscirà più a sopportare il carico di richieste e smetterà di funzionare.
I diversi tipi di DoS sono:

- ping of death
- SYN flooding
- smurf attack
- DDoS (distributed denial of service)

Nel DDoS gli attori sono:

- **vittime primarie**:
  - *macchine zombie* elevato numero di macchine ignare di essere usate per l'attacco.
  - *macchine master* minor numero di macchine infettate o macchine dell'attaccante che sincronizzano le zombie.
- **vittime finali** che subiscono richieste dalle macchine zombie.

Le fasi che caratterizzano questo attacco sono due:

1. Vengono infettate molte macchine che ricoprono il ruolo di zombie e master, questa fase può durare anche mesi.
2. Le vittime finali vengono inondate di richieste e pacchetti dalle zombie.

L'attacco DDoS è quindi un particolare tipo di DoS in cui il malintenzionato infetta altri computer sulla rete per fare in modo di non essere egli stesso colui che effettua direttamente l'attacco, ma altri computer di ignari utenti.
La **differenza** tra i due è che in DoS è l'utente malizioso a sferrare l'attacco mentre in DDoS l'utente lo scatena ma non lo effettua in prima persona.

[2] Nell'ambito delle tecniche di autenticazione basate su caratteristiche dell'utente, descrivere in cosa consiste la fase di enrollment .

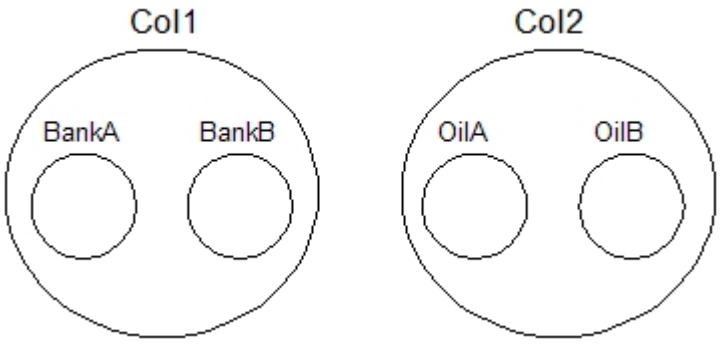
La **fase di Enrollment** è la fase iniziale dell'autenticazione basata sulle caratteristiche. Durante questa fase vengono prese più misurazioni della stessa caratteristica (es: più misurazioni della voce dell'utente) e viene definito un *template*, cioè un valore medio delle misurazioni effettuate. Il template serve per poter successivamente effettuare l'autenticazione tramite confronto con la misurazione effettuata.
Nell'ambito delle politiche mandatorie per l'integrità, spiegare i principi in base ai quali si verifica se le operazioni di letture e scritture possono essere permesse, supponendo di applicare una politica:

i) low water mark per oggetti
ii) low water mark per soggetti

[1] Nell'ambito dei meccanismi per il controllo dell'accesso, elencare e spiegare le cinque caratteristiche base di un reference monitor

- tamper-proof: non può essere alterato
- non-bypassabile: media tutte le richieste di accesso al sistema e alle sue risorse
- security kernel: deve essere confinato in una parte limitata del sistema
- piccolo: deve poter essere verificato con metodi rigorosi e formali
- resistente a covert channel.

[1] Data una politica ChineseWall dove i company dataset sono BankA, BankB, OilA, OilB e le classi di conflitto di interesse sono CoI1 = {BankA, BankB} e CoI2 = {OilA, OilB}, si supponga che un soggetto S1 esegua una operazione di lettura su un oggetto del dataset BankA ed un oggetto del dataset OilB. Si richiede di indicare i dataset che contengono oggetti che S1 pu`o scrivere e di motivare la risposta.



All'inizio un utente può leggere qualsiasi cosa. Se legge da BankA significa che escludiamo BankB, perché fa parte della stessa classe di conflitto di interessi, mentre rimangono accessibili OilA e OilB. Successivamente se legge da OilB significa che dobbiamo escludere OilA, ancora perché fa parte della stessa classe di conflitto di interessi.
L'utente S1 non può scrivere da nessun parte se non all'interno di BankA e OilB, perché i company dataset rimanenti(BankB e OilA) sono stati esclusi per conflitto di interessi.

[1] Nell'ambito delle tipologie di attacco alle reti, descrivere in cosa consiste l'attacco denominato masquerade.

Il termine **masquerade** indica la circostanza in cui una macchina dichiara di essere una macchina diversa. Sotto questo nome rientrano i seguenti casi:

- *URL confusion*: i nomi di dominio si possono facilmente confondere(es: xyz.com viene mascherato con xyz.it)
- *Phishing*: attraverso messaggi di posta elettronica fasulli opportunamente creati per apparire autentici si cerca di estorcere dati personali all'utente con la finalità di furto d'identità(es: la mail fasulla delle poste italiane che vi segnalava che eravate in rosso sul conto).

[1] Si supponga di avere a disposizione un web server Apache. Si richiede di definire delle regole di controllo dell'accesso che permettano l'accesso al contenuto della directory /usr/pippo/www solo all'utente esame05 (esame05 ` e la login dell'utente) le cui richieste di accesso possono essere accettate solo se provengono dalla macchina con IP 134.123.786.9.

order allow, deny
allow from 134.123.786.9
require user esame05
satisfy all

[1] Descrivere le componenti principali di SAML.

Il protocollo SAML (*Security Assertion Markup Language*) è un insieme di specifiche chiamate *asserzioni*, usate per lo scambio di informazioni. Sono basate sul linguaggio XML e sono progettate per funzionare sui meccanismi di trasporto più comuni, come HTTP, SMTP, FTP e altri ancora. Le asserzioni possono essere di tre tipi:

- di *autenticazione*, che identificano un utente
- di *attributi*, che contengono le informazioni specifiche dell'utente
- di *permessi*, che definiscono quali operazioni può compiere un utente e quali no

Indipendentemente dal tipo, ogni assezione deve contenere il suo identificatore univoco, la data e l'ora di creazione, il periodo di validità e cosa essa dichiara, oltre ad informazioni addizionali come ad esempio la dipendenza da altre asserzioni.
Infine è importante sottolineare che SAML di per sé non autentica né autorizza nessuno, si limita esclusivamente a trasportare informazioni tra i domini.

[4] Descrivere le caratteristiche principali di un virus polimorfo.

Il **virus polimorfo**, può assumere un numero elevato o illimitato di forme diverse. Non è sufficiente che utilizzi stringhe casuali se poi le inserisce sempre nella stessa posizione, o si potrebbe considerare come signature le parti che rimangono invariate; un virus polimorfo come si deve invece deve cambiare, suddividere e distribuire il proprio codice in varie porzioni del programma da infettare.
Un ulteriore raffinamento per rendere il rilevamento ancora più difficoltoso è fare sì che il virus polimorfo adoperi tecniche crittografiche per cambiare forma.

[1] Descrivere la differenza tra utenti e soggetti nelle politiche mandatorie.

Nelle politiche mandatorie gli **utenti**(od oggetti) sono le persone fisiche, sono entità passive che possono godere di autorizzazioni e che possono connettersi al sistema. Una volta connessi al sistema, gli utenti originano processi (i **soggetti**) che si eseguono automaticamente e quindi sottopongono richieste al sistema.
Le politiche mandatorie prendono in considerazione tale distinzione tra utente e soggetto e valutano tutte le richieste sottoposte dal processo per conto di qualche utente verificandone la classe di sicurezza a cui appartengono.
L'assunto fondamentale è che gli utenti sono fidati, se hanno l'autorizzazione ad accedere alle informazioni, mentre processi non lo sono.

[1] Descrivere i concetti di separazione statica dei privilegi e di separazione dinamica dei privilegi , mostrando anche un esempio per ognuno di essi. Quale di questi due concetti `e specificatamente supportato dal modello Chinese Wall? Si richiede di motivare la risposta.

La separazione dei privilegi deriva dal principio per cui nessun utente dovrebbe essere fornito di privilegi sufficienti per poter fare cattivo uso del sistema.
Separazione Statica dei Privilegi: consiste nell'assegnare a priori le autorizzazioni in modo da non dare troppi privilegi ad un singolo utente. Ad esempio una politica tale che gli utenti appartenenti alla categoria "impiegato" non possono accedere a documenti appartenenti alla categoria "top secret".
Separazione Dinamica dei Privilegi: in principio l'utente non ha limitazioni, ma i suoi privilegi vengono ristretti man mano che compie operazioni. Quindi l'utente è libero di scegliere quali accessi fare (non tutti), il sistema gli negherà gli altri di conseguenza. Ad esempio una politica tale che in principio chiunque può leggere qualunque documento, ma nel momento in cui legge il Necronomicon non può più avere accesso a nessun file di qualsiasi altro autore.

Il Chinese Wall è una politica mandatoria che applica una separazione dinamica dei privilegi, infatti inizialmente ciascun utente può accedere a qualsiasi oggetto, mentre le restrizioni si attivano una volta che l'utente ha effettuato l'accesso ad un oggetto e vengono applicate agli oggetti che si trovano all'interno della stessa classe di conflitto di interessi, ma non nello stesso company dataset.

Si supponga di avere n utenti ed m oggetti. Rispondere alle seguenti domande.

- **Quante sone le celle della matrice di controllo degli accessi? Quante sono le ACL? Quante sono le capa- bility? Nel caso in cui si aggiungesse un nuovo utente al sistema, quante sono, nel caso peggiore, le ACL che devono essere modificate? Quante sono le capability che devono essere create?**
- **Alice non vuole pi`u che altri utenti siano in grado di modificare un suo oggetto. Cosa deve fare Alice per revocare l'accesso a tale oggetto? Si confronti lo sforzo necessario nel caso in cui il sistema sia basato su ACL oppure su capability.**

- Celle della matrice del controllo degli accessi: $n*m$
- Se si intende le righe della ACL: m
- Se si intende le righe della capabilty: n

- caso peggiore: m
- 1
- ACL è sicuramente più efficiente per le revoche relative ad un oggetto: per Alice sarà sufficiente eliminare la riga relativa alla risorsa a cui non vuole più consentire l'accesso da parte degli altri utenti. Capability invece è più orientato ai soggetti, quindi per revocare l'accesso all'oggetto Alice dovrà eliminare per ciascun utente l'elemento dalla lista delle operazioni consentite che riguarda la risorsa in questione.

[2] Nel caso in cui un sistema supporti sia autorizzazioni positive sia autorizzazioni negative, dire cosa si intende per inconsistenza e non completezza. Si richiede inoltre di descrivere come si possono risolvere questi problemi.

L'uso combinato di autorizzazioni positive e negative conduce al problema di come le due autorizzazioni dovrebbe essere trattate nelle seguenti circostanze:

- **non completezza**: per un accesso non abbiamo né autorizzazioni negative né autorizzazioni positive
  - *Contromisure*: la completezza può essere ottenuta assumendo di default una politica chiusa o aperta.
- **inconsistenza**: per un accesso ci sono sia autorizzazioni positive che negative
  - *Contromisure*: non ha un'unica risposta perché è più difficile da trattare. Le soluzioni possibili dipendono dalle situazioni in cui ci troviamo e corrispondono a diverse politiche che possono essere implementate.
    - denials take precedence: autorizzazioni negative vincono
    - most specific takes precedence: l'autorizzazione più specifica vince
    - most specific along a path takes precedence: l'autorizzazione più specifica vince solo sui cammini che passano dall'autorizzazione
    - chinese wall: politica mandatoria che applica la separazione dinamica dei privilegi per prevenire flussi di informazione che possono causare conflitti di interesse tra gli utenti.

[1] Nell'ambito del controllo dell'accesso, descrivere il significato di politica discrezionale, politica mandatoria e politica basata sui ruoli .

[4] Nell'ambito delle tecniche di autenticazione, descrivere il protocollo di autenticazione delle smart token basato su challenge-response.

smart token: dispositivo che gode di capacità di processo il cui funzionamento si basa su meccanismi statici e dinamici e sul challenge-response:

- *aspetto statico*: l'utente si autentica al token e il token si autentica al sistema (sono necessarie quindi due autenticazioni)
- *aspetto dinamico*: il token genera periodicamente nuove chiavi

challenge-response: è una "sfida" tra server e utente basata sul protocollo *challenge response handshake* e caratterizzata dalle seguenti fasi:

1. il server di autenticazione stabilisce una sfida (challenge)
2. il token genera la risposta (diversa per ogni sfida)
3. l'autenticazione avviene con successo se la risposta è corretta

Il challenge response è resistente contro replay attack passivi, ma vulnerabile rispetto ad attacchi attivi(reflection attack) e man in the middle attack.

[1] Nell'ambito delle tecniche di autenticazione, dire in cosa consiste il single sign-on. Quali sono i vantaggi di questa tecnica di autenticazione rispetto ad una tecnica tradizionale basata su password.

Grazie ad *internet* è possibile accedere a servizi di ogni tipo, su macchine remote dislocate qua e là su sistemi distribuiti. L'utente sarà dunque costretto ad autenticarsi nell'ambito di ciascun dominio con cui interagisce, dovrà cioè fornire user ID e password per ogni servizio di cui vorrà usufruire. Ciò oltre a comportare la duplicazione delle

informazioni personali, costringe l'utente a ricordarsi più password (che spesso utilizzerà sempre la stessa).

I sistemi **Single Sign On** (*SSO*) sono stati concepiti per risolvere questa situazione, consentendo all'utente di autenticarsi una volta per tutte a più servizi distribuiti utilizzando un'unica credenziale (login e password). La robustezza del sistema rimane legata al processo di autenticazione utilizzato, che deve essere particolarmente forte data la sensibilità delle informazioni da condividere su più sistemi distribuiti.
In un SSO esiste un *dominio primario*, presso cui l'utente si registra e si autentica. Ad esso sono collegati una serie di domini secondari, con i quali esistono delle relazioni di trust per garantire la sicurezza, ai quali il dominio principale si preoccuperà di comunicare le informazioni rilasciate dall'utente che ha usato il servizio di sign on.

Un sistema *Single Sign On* può avere due tipologie: **centralizzata** e **federativa** (in realtà esiste anche quella *cooperativa*, che non sarà trattata in questa sede).
In un *SSO centralizzato* esiste un'autorità centrale e una repository centralizzata che gestiscono le informazioni di autenticazione di TUTTI gli utenti, e che comunica con i domini secondari attraverso protocolli come il *SAML* o il *SOAP*. Il vantaggio di questa tipologia è la facilità di gestione e di controllo dell'accesso, a costo però di avere un *single point of failure*: un attacco diretto al dominio primario danneggerebbe l'intero sistema.

Negli *SSO federati* l'autenticazione è invece realizzata da più server SSO indipendenti, ognuno dei quali possiede un'identità parziale dell'utente. La combinazione delle identità indipendenti registrate nei vari domini forma la ***network identity***; in altre parole, essa è l'insieme di tutte le informazioni dell'utente (username, indirizzo, telefono, ...) che egli ha rilasciato ai diversi service provider per autenticarsi. La ricostruzione della network identity è possibile solo se esiste una *federazione* tra i vari service provider, quindi solo in seguito ad accordi stipulati che l'utente deve rispettare e accettare. Una di queste federazioni è la *Liberty Alliance*, che ha creato una serie di tracciati (*circoli*) che mettono in comunicazione i service provider facendo agire alcuni di essi come server SSO di autenticazione (*identity provider*). Le identità locali vengono combinate in modo che l'utente possa utilizzare i servizi di un circolo dopo l'autenticazione presso uno degli identity provider.

[2] Discutere i principali approcci che possono essere usati per autenticare gli utenti, evidenziando i loro vantaggi e svantaggi.

Ci sono tre tipi di autenticazione:

- **Basata Sulla Conoscenza**.
  Si basa sulla coppia *login*(l'utente si identifica) *password*(l'utente prova la sua identità).
  - Vantaggi: è il più semplice, il più economico, il più implementabile.
  - Svantaggi: è il più debole, perché le password possono essere acquisite in molti modi (guessing; spoofing; sniffed; snooping)
- **Basata Sul Possesso**
  Si basa sul possesso da parte dell'utente di un Token, cioè qualcosa di tangibile che l'utente utilizza per identificarsi. Ogni token ha una chiave crittografica usata per dimostrare l'identità del token al computer.
  - Vantaggi: i token sono più sicuri delle password, mantenendo il controllo sul token l'utente mantiene controllo sull'utilizzo della sua identità.
    Si può usare questo tipo di autenticazione combinata con la Conoscenza, quindi servono token e password per autenticarsi.
  - Svantaggi: i token possono essere persi, rubati, falsificati.
    Chiunque acquisisca un token può impersonare l'utente.
- **Basata Su Caratteristiche**
  Si basa sulle caratteristiche fisiche(impronte digitali; impronta della retina) e comportamentali(timbro della voce; firma) dell'utente. Richiede una fase iniziale(**fase di enrollment**) di misurazione e di definizione di un template, cioè un valore medio delle misurazioni effettuate.
  - Vantaggi: è la forma di autenticazione più forte, infatti elimina le vulnerabilità dovute a impersonificazioni.
  - Svantaggi: troppo costosa, intrusiva, potenziale mancanza di privacy.

[3] Descrivere l'attacco ping of death.

L'attacco **ping of death** è un tipo di Denial of Service e si basa sul protocollo ICMP, specialmente sul comando *ping*. Il comando ping viene usato per determinare se un sistema remoto è raggiungibile o meno dal proprio sistema locale. Il problema quindi non riguarda solo i computer, ma tutto ciò che si connette direttamente alla rete (es: stampanti, router). Le specifiche del TCP/IP fissano la grandezza massima di un datagramma IP in 65536 bytes, di questi solo 20 sono

riservati all'intestazione del pacchetto. Però all'interno del pacchetto risiede una richiesta ICMP costituita da 8 bytes di header seguita dal numero di byte riservati per i dati, la cui dimensione massima è 65507 bytes. Il problema del ping of death nasce dalla possibilità di generare una richiesta ICMP_ECHO in un pacchetto con più di 65507 byte di campo dati.
Da notare che macchine simili possono reagire in modo diverso, in quanto in alcuni casi il fenomeno si può verificare in condizioni di carico gravoso su macchine che in altre condizioni non avevano presentato il problema. In alcuni casi si può anche arrivare al crash e al reboot del sistema.

Quindi riassumendo l'attacco ping of death tramite una richiesta ICMP echo manda in crash il sistema perché la richiesta generata supera il limite di bytes concessi dalla specifica TCP/IP per il campo dati dei pacchetti.