

# Sistemi Informatici - Anonimato in Rete

Michele CORRIAS

Lombardia Plus 2019 Cyber Security - GALDUS

5/12/2019

# Index

- 1 Introduzione
- 2 Anonimato in Rete & HTTP
- 3 HTTPS, Proxy, VPN, TOR, p2p
- 4 Esercizi

# Anonimato in Rete

## Censura e Controllo in Rete

### Controllo in rete

- le reti telematiche sono uno strumento di libertà
- sono esposte al rischio di controllo di massa
  - censura
  - content filtering
  - tracking & profiling

# Anonimato in Rete

## Diritti Fondamentali

### Art. 12 della Dichiarazione Fondamentale dei Diritti dell'Uomo

*No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.*

*Nessun individuo potrà essere sottoposto ad interferenze arbitrarie nella sua vita privata, nella sua famiglia, nella sua casa, nella sua corrispondenza, né a lesione del suo onore e della sua reputazione. Ogni individuo ha diritto ad essere tutelato dalla legge, contro tali interferenze o lesioni.*

# Anonimato in Rete

## Diritti Fondamentali

### Art. 12 della Dichiarazione Fondamentale dei Diritti dell'Uomo

*No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.*

*Nessun individuo potrà essere sottoposto ad interferenze arbitrarie nella sua vita privata, nella sua famiglia, nella sua casa, nella sua corrispondenza, né a lesione del suo onore e della sua reputazione. Ogni individuo ha diritto ad essere tutelato dalla legge, contro tali interferenze o lesioni.*

# Anonimato in Rete

## Diritti Fondamentali

### Articolo 15 della Costituzione Italiana

*La libertà e la segretezza della corrispondenza e di ogni altra forma di comunicazione sono inviolabili.*

*La loro limitazione può avvenire soltanto per atto motivato dell'autorità giudiziaria [cfr. art. 111 c. 1] con le garanzie stabilite dalla legge.*

# Anonimato in Rete

## Tutela

### Diritti dell'utente finale

Agli utenti di una rete spetta di:

- vedere tutelati i propri diritti stabiliti dalla legge
- usare la tecnologia per potersi difendere dentro una rete

# Anonimato in Rete

## Privacy-Enhancing Technologies

### PET

Tecnologia progettata per tutelare la privacy (privatezza, riservatezza):

- non solo in informatica (porta di un bagno ...)
- in informatica
  - tecniche per minimizzare o eliminare i dati personali
  - tecniche per evitare il controllo delle attività



# Anonimato in Rete

## Privacy e Sicurezza

### Punti in comune

La privacy è importante anche per la sicurezza:

- identity theft
- controllo e repressione del dissenso (più efficace della tortura: The Man in the Snow White Cell, CIA)
- diritto all'oblio: le persone cambiano, ma i dati restano

# Anonimato in Rete

## Minimizzazione dei Dati

### Come trattare i dati

- ogni servizio dovrebbe richiedere e raccogliere solo il minimo dei dati necessari al suo funzionamento
- i dati personali e sensibili dovrebbero essere raccolti solo se realmente necessari (ed in tal caso protetti)

# Anonimato in Rete

## Sanitization

### Come trattare i dati

*Sanitizzare* vuol dire eliminare dai dati le caratteristiche che li rendono personali o sensibili:

- molto difficile (dovrebbero risultare anonime anche le aggregazioni statistiche)
- l'anonimato può necessitare spesso di una grande quantità di dati

# Anonimato in Rete

## Protezione

### Come trattare i dati

Ogni volta che dati personali e sensibili vengono conservati, elaborati e trasmessi dovrebbero essere protetti:

- controllo degli accessi
- crittografia
- *shredding*

# Anonimato in Rete

## Protezione

```
dreamer@dreamer-desktop:~$ shred -zuvn20 '/home/dreamer/Text File'
shred: /home/dreamer/Text File: pass 1/21 (random)...
shred: /home/dreamer/Text File: pass 2/21 (777777)...
shred: /home/dreamer/Text File: pass 3/21 (492492)...
shred: /home/dreamer/Text File: pass 4/21 (111111)...
shred: /home/dreamer/Text File: pass 5/21 (aaaaaa)...
shred: /home/dreamer/Text File: pass 6/21 (333333)...
shred: /home/dreamer/Text File: pass 7/21 (c92492)...
shred: /home/dreamer/Text File: pass 8/21 (444444)...
shred: /home/dreamer/Text File: pass 9/21 (555555)...
shred: /home/dreamer/Text File: pass 10/21 (dddddd)...
shred: /home/dreamer/Text File: pass 11/21 (random)...
shred: /home/dreamer/Text File: pass 12/21 (6db6db)...
shred: /home/dreamer/Text File: pass 13/21 (b6db6d)...
shred: /home/dreamer/Text File: pass 14/21 (666666)...
shred: /home/dreamer/Text File: pass 15/21 (ffffff)...
shred: /home/dreamer/Text File: pass 16/21 (000000)...
shred: /home/dreamer/Text File: pass 17/21 (249249)...
shred: /home/dreamer/Text File: pass 18/21 (924924)...
shred: /home/dreamer/Text File: pass 19/21 (db6db6)...
shred: /home/dreamer/Text File: pass 20/21 (random)...
shred: /home/dreamer/Text File: pass 21/21 (000000)...
shred: /home/dreamer/Text File: removing
shred: /home/dreamer/Text File: renamed to /home/dreamer/000000000
shred: /home/dreamer/0000000000: renamed to /home/dreamer/000000000
shred: /home/dreamer/000000000: renamed to /home/dreamer/00000000
shred: /home/dreamer/00000000: renamed to /home/dreamer/000000
shred: /home/dreamer/000000: renamed to /home/dreamer/00000
shred: /home/dreamer/00000: renamed to /home/dreamer/0000
shred: /home/dreamer/0000: renamed to /home/dreamer/000
shred: /home/dreamer/000: renamed to /home/dreamer/00
shred: /home/dreamer/00: renamed to /home/dreamer/0
shred: /home/dreamer/Text File: removed
dreamer@dreamer-desktop:~$
```

Figura 1: Shred

# Anonimato in Rete

## Anonimato

### Controllo

- la difesa rispetto i pericoli di controllo è l'anonimato
- tutti i tentativi di controllo politico di Internet cercano in vari modi di limitare l'accesso anonimo
- controversia: l'anonimato perfetto permette azioni non perseguibili
- possibilità di contrasto tra legalità e diritti fondamentali

# Anonimato in Rete

## Caratteristiche

### Unobservability

Inosservabilità: un utente utilizza una risorsa, senza che terzi siano possibilitati a osservarne l'utilizzo

- *perfect unobservability*: nessuna osservazione è in grado di cambiare la probabilità a posteriori di un evento

### Unlinkability

Incollegabilità: un utente utilizza diverse risorse o servizi, senza che sia possibile ricollegare i diversi utilizzi

- unlinkability tra mittente e destinatario in una comunicazione
- A e B comunicano: che A comunica è osservabile, che B comunica anche ... non è osservabile che A comunica con B

# Anonimato in Rete

## Caratteristiche

### Unobservability

Inosservabilità: un utente utilizza una risorsa, senza che terzi siano possibilitati a osservarne l'utilizzo

- *perfect unobservability*: nessuna osservazione è in grado di cambiare la probabilità a posteriori di un evento

### Unlinkability

Incollegabilità: un utente utilizza diverse risorse o servizi, senza che sia possibile ricollegare i diversi utilizzi

- unlinkability tra mittente e destinatario in una comunicazione
- A e B comunicano: che A comunica è osservabile, che B comunica anche ... non è osservabile che A comunica con B



# Anonimato in Rete

## Caratteristiche

### Anonymity

Anonimato: un utente utilizza una risorsa, senza rendere nota la propria identità

### Pseudonymity

Pseudonimo: un utente utilizza una risorsa identificandosi con uno pseudonimo

- pseudonimo costante
- può essere legato ad un ruolo
- non è possibile (o solo alcuni possono) collegarlo all'identità reale

# Anonimato in Rete

## Caratteristiche

### Anonymity

Anonimato: un utente utilizza una risorsa, senza rendere nota la propria identità

### Pseudonymity

Pseudonimo: un utente utilizza una risorsa identificandosi con uno pseudonimo

- pseudonimo costante
- può essere legato ad un ruolo
- non è possibile (o solo alcuni possono) collegarlo all'identità reale

# Anonimato in Rete

## Privacy e Web

### Web

Caricare una pagina web da un server espone moltissimi *dati personali*, alcuni addirittura *sensibili*:

- IP del client
- IP del server
- identità (<https://panopticklick.eff.org>)
- dati del browser (cronologia, ...)
- *system profile*
- dati trasmessi con form ...

# Anonimato in Rete

## HTTP

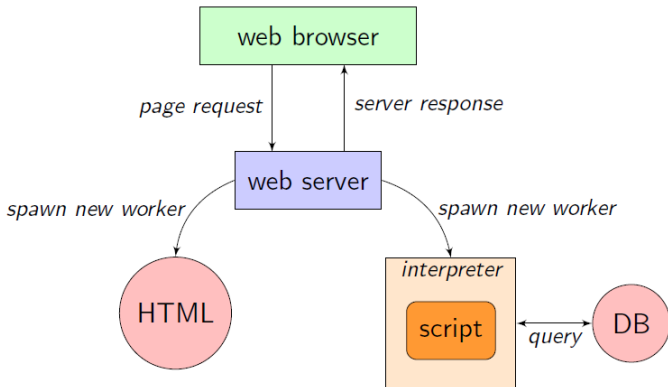


Figura 2: Architettura Infrastruttura Web

# Anonimato in Rete

## HTTP

### HyperText Transfer Protocol

- elemento fondamentale del World Wide Web
- protocollo di livello applicazione, usato per trasferire dati tra client e server
- *text-based* e *stateless*
- versioni: 1.0 (RFC 1945), 1.1 (RFC 7231), 2.0 (RFC 7540)
- incapsulato all'interno di connessioni TCP, di default su porta 80
- oggi utilizzato per trasportare anche altre informazioni (SOAP ...)

# Anonimato in Rete

Richiesta HTTP `http://www.example.com/test.html`

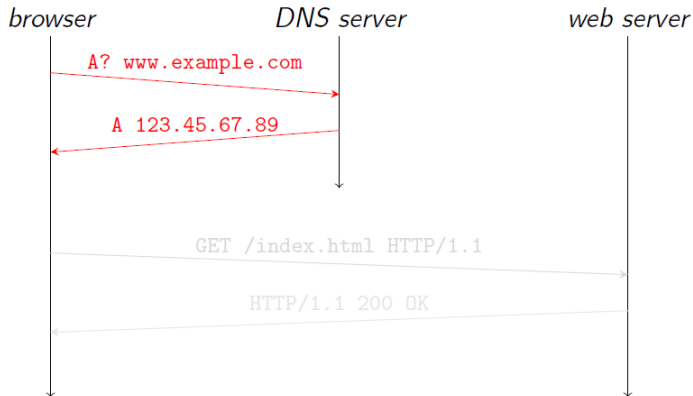


Figura 3: browser interroga DNS per ottenere indirizzo IP del server

# Anonimato in Rete

Richiesta HTTP `http://www.example.com/test.html`

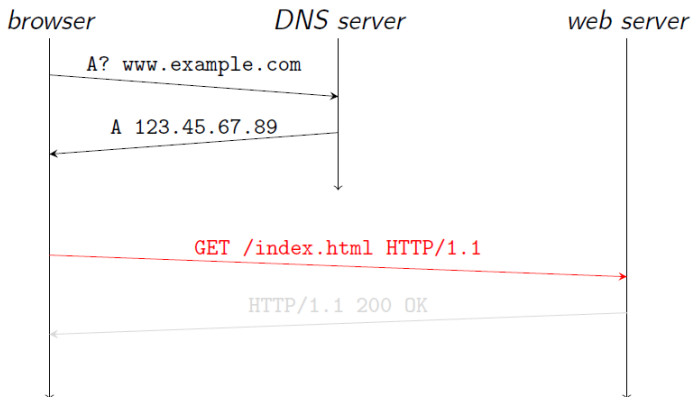


Figura 4: browser si collega alla porta TCP 80 del server e invia una richiesta HTTP

# Anonimato in Rete

Richiesta HTTP `http://www.example.com/test.html`

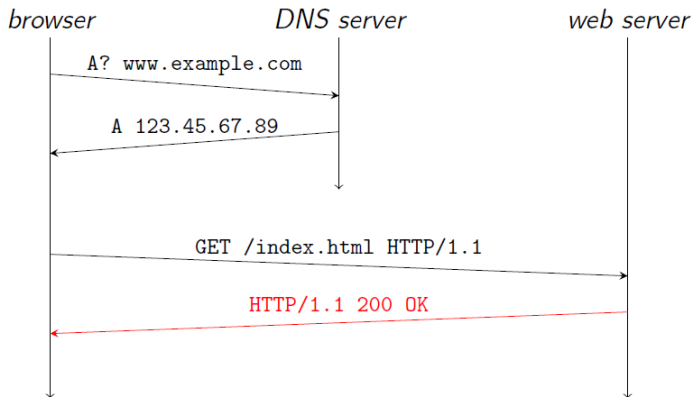


Figura 5: server processa la richiesta ricevuta e restituisce una risposta HTTP (pagina HTML ...)



# Anonimato in Rete

## HTTP Request

### Struttura

- *request line* (GET /index.html HTTP/1.1 ...)
- *header*: opzionali (User-Agent: Mozilla/5.0 (X11; U; Linux i686) ...) ma Host obbligatorio da HTTP 1.1
- linea vuota
- corpo del messaggio (opzionale)

### Note

- *request line* e *header* terminati da CRLF: *carriage return + line feed*: \r\n
- linea vuota formata da CRLF
- implementazioni piuttosto flessibili: req. accettate anche con linee terminate solo da LF

# Anonimato in Rete

## HTTP Request

```
GET /index.html HTTP/1.1
Host: security.di.unimi.it
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64 ...) ...
Accept: text/html,application/xhtml+xml,application/xml; ...
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Referer: http://security.di.unimi.it/index.html
Upgrade-Insecure-Requests: 1
If-Modified-Since: Wed, 06 Mar 2019 13:59:31 GMT
If-None-Match: "16fe-5836d661fdc47-gzip"
Cache-Control: max-age=0
```

# Anonimato in Rete

## HTTP Response

### Struttura

- *status line* (HTTP/1.1 200 OK ...)
- *header*: opzionali (Apache/2.4.25 (Debian) ...)
- linea vuota
- corpo del messaggio (opzionale)

# Anonimato in Rete

## HTTP Response

```
HTTP/1.1 200 OK
Date: Mon, 09 Dec 2019 08:21:27 GMT
Server: Apache/2.4.25 (Debian)
Last-Modified: Wed, 06 Mar 2019 13:59:31 GMT
ETag: "16fe-5836d661fdc47-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Length: 5886
Connection: close
Content-Type: text/html
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
```

# Anonimato in Rete

## HTTP: GET/POST

### GET

GET è un metodo per inviare dati usando il protocollo HTTP. Secondo la specifica del protocollo di questo metodo, i dati sono preceduti dall'indirizzo della pagina richiesta e un punto interrogativo

# Anonimato in Rete

## HTTP: GET/POST

### 1: passaggio parametri tramite *form*

```
<form action="submit.php" method="get">  
  <input type="text" name="var1" />  
  <input type="hidden" name="var2" value="b" />  
  <input type="submit" value="invia" />  
</form>
```

### 2: parametri *embedded* nell'URL

```
<a href="submit.php?var1=a&var2=b">link</a>
```

### Richiesta HTTP corrispondente

```
GET /submit.php?var1=a&var2=b HTTP/1.1  
Host: www.example.com  
...
```

# Anonimato in Rete

## HTTP: GET/POST

### POST

POST è un metodo per inviare dati usando il protocollo HTTP. Secondo la specifica del protocollo di questo metodo, i dati sono inviati dopo che tutti gli header sono stati inviati dal client al server

# Anonimato in Rete

## HTTP: GET/POST

### passaggio parametri POST

```
<form action="submit.php" method="post">  
  <input type="text" name="var1" />  
  <input type="text" name="var2" />  
  <input type="submit" value="invia" />  
</form>
```

```
POST /submit.php HTTP/1.1  
Host: localhost  
...  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 13
```

```
var1=a&var2=b
```

### GET + POST

```
<form action="test.php?var3=c&var4=d"  
method="post">  
  <input type="text" name="var1" />  
  <input type="text" name="var2" />  
  <input type="submit" value="invia" />  
</form>
```

```
POST /test.php?var3=c&var4=d HTTP/1.1  
Host: localhost  
...  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 13
```

```
var1=a&var2=b
```



# Anonimato in Rete

## HTTP

### Problema

- *stateless*: ogni richiesta è indipendente dalle precedenti
- le applicazioni web dinamiche richiedono concetto di *sessione*

### Cookie

- dati creati dal server e memorizzati sul client
- trasmessi tra client e server utilizzando header HTTP
- creazione cookie: `Set-Cookie: param=value<CRLF>`
- il client memorizza localmente l'info e nelle req. successive aggiunge: `Cookie: param=value<CRLF>`
- cookie standardizzati in RFC 2109 (*HTTP State Management Mechanism*)

# Anonimato in Rete

## HTTP

### Sessione

Una sessione permette di gestire l'interazione tra client e server web (*stateful*)

### Componenti

- variabili di sessione
- identificativo di sessione

### Caratteristiche

- informazioni e stato devono essere memorizzati
- ogni richiesta HTTP deve contenere un ID di sessione
- le sessioni devono avere un timeout

# Anonimato in Rete

## HTTP

### Sessione

- il concetto di *sessione* è implementato dall'applicazione web
- le informazioni di sessione devono passare tra client e server
- la trasmissione può avvenire tramite:

❶ header HTTP

```
GET /page.php HTTP/1.1
```

```
Host: www.example.com
```

```
...
```

```
Cookie: sessionid=7456
```

```
...
```

❷ URL

```
http://www.example.com/page.php?sessionid=7456
```

❸ payload HTTP

```
<INPUT TYPE="hidden" NAME="sessionid" VALUE="7456">
```

# Anonimato in Rete

## HTTP

### Sessione

- è un elemento critico
  - bypass del sistema di autenticazione
  - deve essere valida per un periodo di tempo limitato
  - attacchi:
    - intercettazione → SSL/TLS
    - predizione → *strong pseudonumber*
    - brute force → ID length
    - session fixation → controllo IP, Referer, rigenerazione ID,
- ...

# Anonimato in Rete

## HTTP

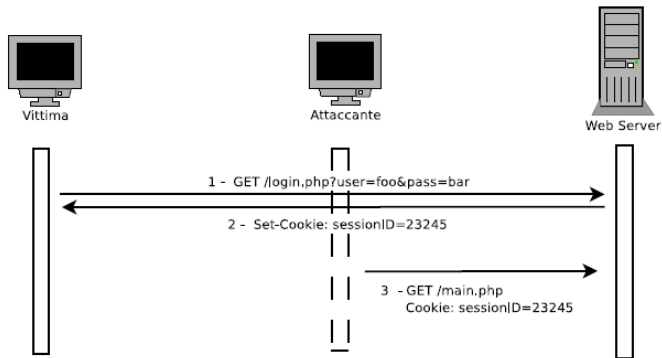


Figura 6: Session Hijacking

# Anonimato in Rete

## HTTP

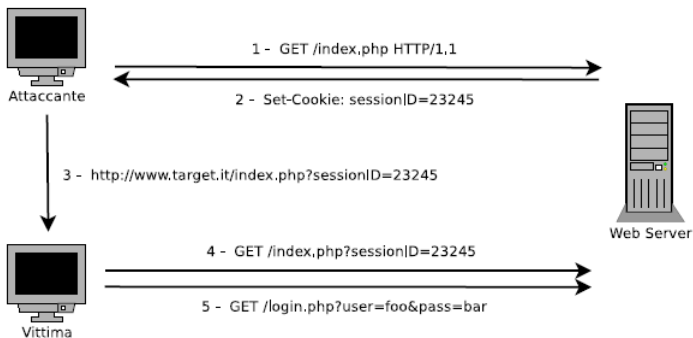


Figura 7: Session Fixation

# Anonimato in Rete

## HTTP

### Analisi di traffico HTTP

- payload HTTP incapsulato dentro il segmento TCP
- comunicazione *in chiaro*
- osservazione del traffico HTTP per analisi
- analisi tramite strumenti di *sniffing*: Wireshark, ...

# Anonimato in Rete

## HTTP

### Chi è il nemico?

- *eavesdropper*: osserva il traffico, in segreto e senza consenso
- server web conserva i dati dei client
- il gestore della rete osserva e/o censura il traffico

### Come difendersi?

- 1 HTTPS
- 2 proxy
- 3 VPN
- 4 *onion routing*



# Anonimato in Rete

## HTTP

### Chi è il nemico?

- *eavesdropper*: osserva il traffico, in segreto e senza consenso
- server web conserva i dati dei client
- il gestore della rete osserva e/o censura il traffico

### Come difendersi?

- ① HTTPS
- ② proxy
- ③ VPN
- ④ *onion routing*

# HTTPS, Proxy, VPN, TOR, p2p

## HTTPS

### HyperText Transfer Protocol over Secure Socket Layer

- anche noto come HTTP Secure
- protocollo per la comunicazione sicura, attraverso una rete di computer, utilizzato su Internet
- la porta utilizzata generalmente è la 443
- comunicazione tramite protocollo HTTP all'interno di una connessione criptata con crittografia asimmetrica
- Transport Layer Security (TLS) o il suo predecessore Secure Sockets Layer (SSL): protocolli crittografici di presentazione (operano al di sopra del livello di trasporto)

# HTTPS, Proxy, VPN, TOR, p2p

## HTTPS

### HyperText Transfer Protocol over Secure Socket Layer

- garantisce:
  - autenticazione del sito web visitato
  - protezione della privacy
  - integrità dei dati scambiati
- protegge da attacchi *man-in-the-middle* e *eavesdropping*
- attenzione: anche con traffico cifrato, gli IP e il *system profile* rimangono accessibili

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

### Intermediario

- componente che media la comunicazione tra due parti comunicanti
- separa la comunicazione tra due componenti, ponendosi in mezzo e disaccoppiandola, rendendola quindi indiretta
- agisce sia da client (rispetto al server originale) che da server (rispetto al client originale)

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

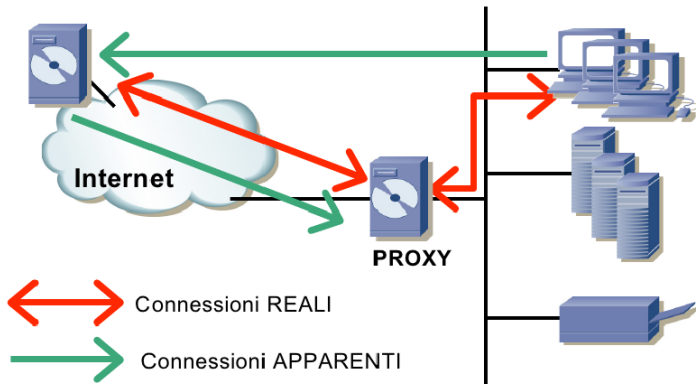


Figura 8: Proxy

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

### Categorie di proxy

- *web proxy*: funziona come cache di pagine web
- *anonymizing proxy*: garantisce l'anonimizzazione di una connessione web
- *forward proxy*: fornisce ad utenti interni ad una rete l'accesso alle risorse esterne a tale rete
- *reverse proxy*: fornisce ad utenti esterni ad una rete l'accesso alle risorse interne di tale rete (tipicamente aziendale)
- *proxy firewall*: simile ad un firewall stateful, collocato però a livello di protocollo applicativo, gestisce aspetti di sicurezza dei protocolli

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

### HTTP header

HTTP\_FORWARDED  
HTTP\_VIA  
HTTP\_X\_FORWARDED\_FOR  
HTTP\_X\_FORWARDED\_HOST  
HTTP\_X\_FORWARDED\_PROTO

### Livelli di anonymizing

- *transparent proxy*: in HTTP\_X\_FORWARDED\_FOR rimane visibile l'indirizzo IP del client originale che fa partire la richiesta
- *anonymous proxy*: nasconde l'indirizzo del client originale
- *distorting proxy*: falsifica l'indirizzo del client originale
- *high anonymity proxy*: HTTP\_VIA vuoto

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

### Privoxy

Web proxy con capacità elevate di filtraggio, progettato appositamente per la privacy:

- gestisce i cookie
- javascript
- altamente *customizzabile*
- *protocol cleaner*



# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

### HTTP proxy

- funziona come *man-in-the-middle* tra il browser e l'applicazione target
- modifica del traffico HTTP/HTTPS
- indipendenti dall'applicazione
- intercettando traffico HTTPS, il browser notifica l'errore di verifica del certificato SSL

### Esempi di HTTP proxy

- Burp - <https://portswigger.net/burp>
- WebScarab - <https://www.owasp.org>
- proxy - <https://code.google.com/archive/p/proxy>

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

### HTTP proxy

- funziona come *man-in-the-middle* tra il browser e l'applicazione target
- modifica del traffico HTTP/HTTPS
- indipendenti dall'applicazione
- intercettando traffico HTTPS, il browser notifica l'errore di verifica del certificato SSL

### Esempi di HTTP proxy

- Burp - <https://portswigger.net/burp>
- WebScarab - <https://www.owasp.org>
- proxy - <https://code.google.com/archive/p/proxy>

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

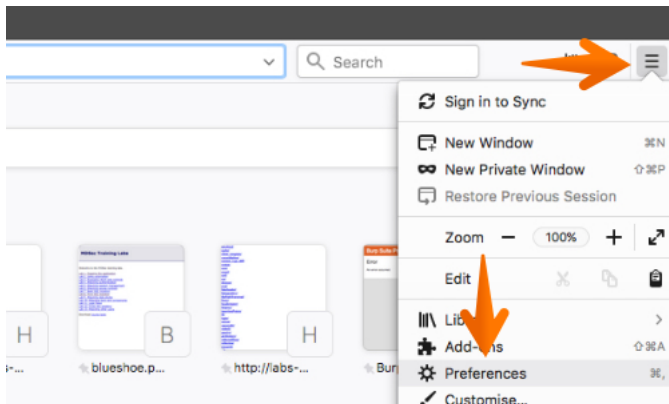


Figura 9: Burp - fase 1

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

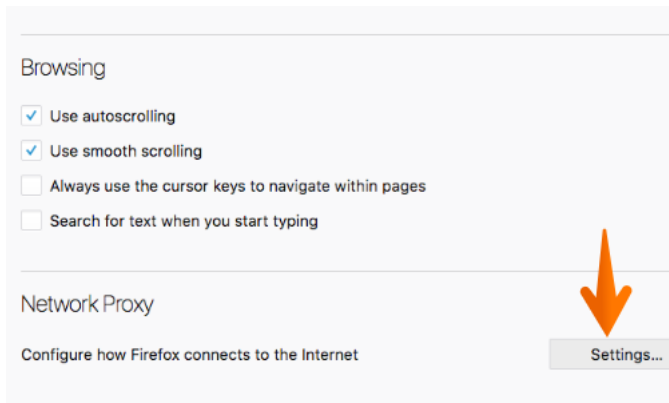
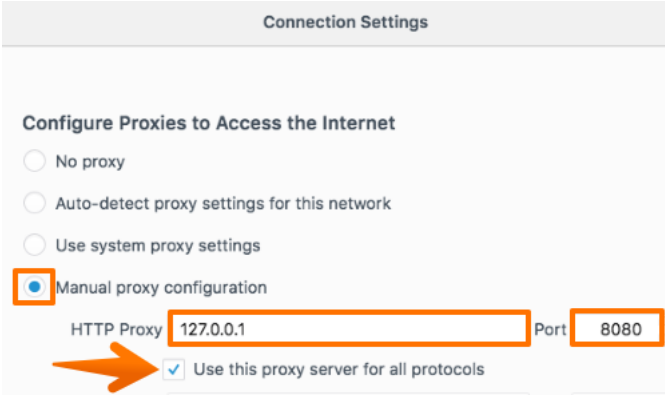


Figura 10: Burp - fase 2

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy



The screenshot shows the 'Connection Settings' window in Burp Suite. Under the heading 'Configure Proxies to Access the Internet', the 'Manual proxy configuration' option is selected and highlighted with an orange box. Below this, the 'HTTP Proxy' field contains '127.0.0.1' and the 'Port' field contains '8080', both highlighted with orange boxes. A large orange arrow points to the checkbox labeled 'Use this proxy server for all protocols', which is checked.

Figura 11: Burp - fase 3 (there is no place like 127.0.0.1)

# HTTPS, Proxy, VPN, TOR, p2p

## Proxy

No Proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Automatic proxy configuration URL

Reload

Do not prompt for authentication if password is saved

Proxy DNS when using SOCKS v5

Help Cancel OK




Figura 12: Burp - fase 4

# HTTPS, Proxy, VPN, TOR, p2p

## VPN

### Virtual Private Network

- rete *overlay*
- forma un dominio amministrativo in sostanza indipendente dall'effettiva topologia della rete fisica sottostante
- comunicazioni criptate
- permette l'accesso alle risorse di una rete aziendale agli utenti *roaming*
- estensione geografica di una grande rete locale privata e sicura
- utilizza *routing* tramite IP per il trasporto su scala geografica, implementando una rete LAN virtuale e privata

# HTTPS, Proxy, VPN, TOR, p2p

## VPN

### Tunnel

- VPN basate sul concetto di tunnel
- *tunneling*: incapsulare pacchetti di uno specifico protocollo, non supportato, in un altro protocollo che rispetta la topologia della rete fisica sottostante (es.: protocollo IPv6 in una rete IPv4 compatibile)
- tunnel *virtuale* protetto e sicuro, supportato da Internet, esattamente come fosse il cavo fisico di cablaggio alla LAN

### Implementazioni

- tramite OS più comuni: Windows, Linux, macOS, ...
- tramite software di terze parti: Cisco VPN Client, OpenVPN ...



# HTTPS, Proxy, VPN, TOR, p2p

## VPN

### Autenticazione

- *qualcosa che sai*: password, PIN, ...
- *qualcosa che hai*: smartcard, ...
- *qualcosa che sei*: biometria

### Encryption

Autenticazione, confidenzialità e integrità del messaggio possono avvenire a diversi livelli:

- IPsec: IP Security, protocollo standard per connessioni sicure su reti IP
- SSL/TLS: protocolli standard crittografici di livello presentazione
- protocolli proprietari

# HTTPS, Proxy, VPN, TOR, p2p

## VPN

### Autenticazione

- *qualcosa che sai*: password, PIN, ...
- *qualcosa che hai*: smartcard, ...
- *qualcosa che sei*: biometria

### Encryption

Autenticazione, confidenzialità e integrità del messaggio possono avvenire a diversi livelli:

- IPsec: IP Security, protocollo standard per connessioni sicure su reti IP
- SSL/TLS: protocolli standard crittografici di livello presentazione
- protocolli proprietari

# HTTPS, Proxy, VPN, TOR, p2p

## VPN

### Tipologie di VPN

- *Trusted VPN*
  - utilizzo in maniera esclusiva del circuito
  - importanza e fiducia sul percorso in cui si muovono i dati
- *Secure VPN*
  - attenzione alla sicurezza (cifratura dei dati)
  - non assicurano qualità sui percorsi (*QoS* ...)
- *Hybrid VPN*
  - Secure VPN adoperata come Trusted VPN
  - recentemente introdotta sul mercato

# HTTPS, Proxy, VPN, TOR, p2p

## VPN

### OpenVPN

- programma *open-source* che permette di costruire VPN basate su tunnel TCP o UDP
- 1194: porta assegnata da IANA
- implementazione sfrutta l'interfaccia TUN/TAP: driver che permettono la creazione di periferiche di rete virtuali
  - interfaccia TUN: livello network
  - interfaccia TAP: livello link
- autenticazione con chiave condivisa oppure certificati e scambio di chiavi

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Onion Routing (OR)

- sviluppato dal *Naval Research Laboratory* di Washington
- traffico instradato attraverso sequenza variabile di *onion router*, per complicare tracciamento attività
- *onion router* = *mix di Chaum*

### Mix Networks di David Chaum

Un *mix*:

- 1 riceve messaggi di lunghezza fissata
- 2 li cifra
- 3 attende di averne abbastanza da garantire anonimato
- 4 inoltra i messaggi (in ordine arbitrario) ad altri *mix*

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Onion Routing (OR)

- sviluppato dal *Naval Research Laboratory* di Washington
- traffico instradato attraverso sequenza variabile di *onion router*, per complicare tracciamento attività
- *onion router* = *mix* di Chaum

### Mix Networks di David Chaum

Un *mix*:

- ① riceve messaggi di lunghezza fissata
- ② li cifra
- ③ attende di averne abbastanza da garantire anonimato
- ④ inoltra i messaggi (in ordine arbitrario) ad altri *mix*

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Onion Router

- *onion router* (or) intermedi non hanno informazioni sufficienti per tracciare sender, receiver e traffico
- criticità degli **entry node** e (soprattutto) degli **exit node**

### TOR

The Onion Router project: evoluzione più recente del concetto di *onion routing*:

- sviluppato da *NRL*
- *open-source*
- supportato da *Electronic Frontier Foundation (EFF)*
- prodotto utilizzabile anche da utenti meno esperti: alto numero di utenti è bene per anonimato

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Onion Router

- *onion router* (or) intermedi non hanno informazioni sufficienti per tracciare sender, receiver e traffico
- criticità degli **entry node** e (soprattutto) degli **exit node**

### TOR

The **Onion Router project**: evoluzione più recente del concetto di *onion routing*:

- sviluppato da *NRL*
- *open-source*
- supportato da *Electronic Frontier Foundation (EFF)*
- prodotto utilizzabile anche da utenti meno esperti: alto numero di utenti è bene per anonimato



# HTTPS, Proxy, VPN, TOR, p2p

## TOR

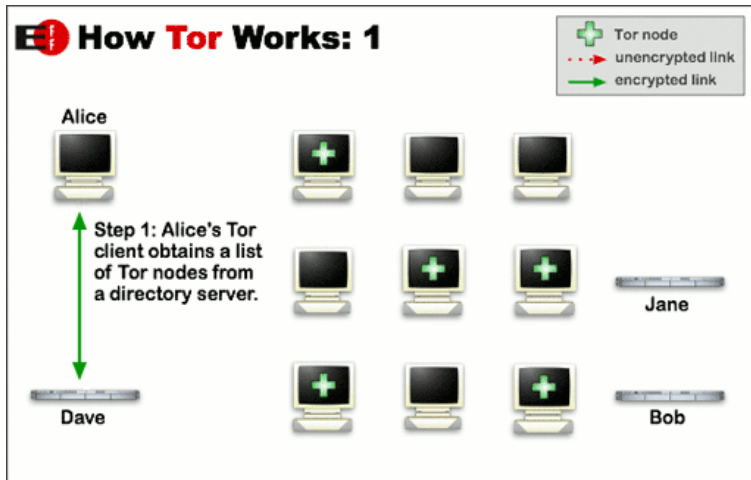


Figura 13: TOR - fase 1

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

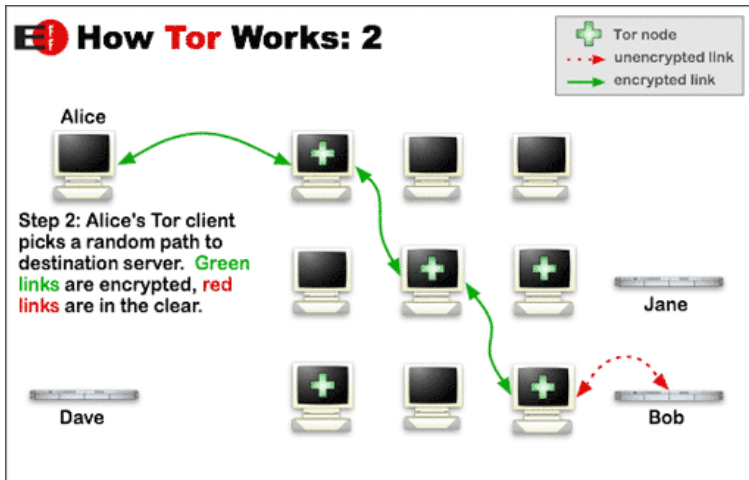


Figura 14: TOR - fase 2

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Set-up

- Alice: sender
- Jane e Bob: public server, raggiungibili in Internet
- Dave: *Tor Directory server*, fornisce ed aggiorna periodicamente la lista dei nodi TOR

### Connessione Alice - Bob

- Alice deve contattare Bob
- il nodo TOR di Alice si costruisce una catena di nodi attraverso i quali invierà la richiesta
- ogni collegamento tra **or** nella rete TOR è cifrato
- l'ultimo **or** contatta Bob al di fuori della rete TOR e non è garantito che il collegamento sia ancora cifrato

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

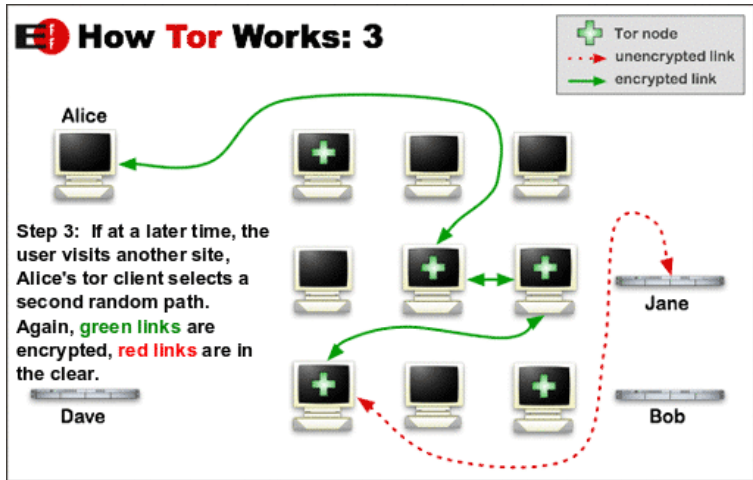


Figura 15: TOR - fase 3

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Circuito

L'instradamento di ogni messaggio viene detto *circuito*:

- ogni nodo del circuito conosce solo il nodo precedente e successivo (non sender e receiver)
  - *Guard Node*: nodo di guardia, primo nodo della catena; molto importanti, sono *trusted* e operano da directory server iniziali
  - *Middleman Node*: nodo intermedio nella catena
  - *Exit Node*: nodo di uscita dalla catena; molto importanti, sono i più critici e contattano le macchine all'esterno della rete TOR
- diverse richieste incorporate (*multiplexing*) in un unico circuito
- robusto a introduzione di nodi maligni o compromissione

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Struttura di una rete TOR

- nodi utenti hanno un *onion proxy* (op) (Alice)
- or connessi tra loro con TLS
- or hanno *long-term-identity key* e *short-term-onion key*
- PDU → *cella* (dimensione fissa 512 byte)

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Uso delle chiavi

- *long-term-identity key* per firma digitale di:
  - descrizione dei router (certificati SSL/TLS, chiavi, metadati, ...)
  - elenchi dei router
- *short-term-onion key* per:
  - cifratura/decifratura di richieste di circuiti
  - negoziazione di chiavi *una tantum*, che garantiscono *forward secrecy* (segretezza in avanti: se una chiave di cifratura viene violata le chiavi generate da essa non risultano compromesse)

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Circuiti

- in una rete TOR almeno tre nodi di default
- op costruisce un circuito in background
- più stream utente vengono combinati e trasmessi (*multiplexati*) sullo stesso circuito
- creato nuovo circuito ogni minuto
- op sceglie quale or fa da exit node (diverse *exit policy*)
- solo TCP stream (UDP problematico)
- *protocol cleaner* necessario per evitare che informazioni rilevanti finiscano nello stream



# HTTPS, Proxy, VPN, TOR, p2p

TOR

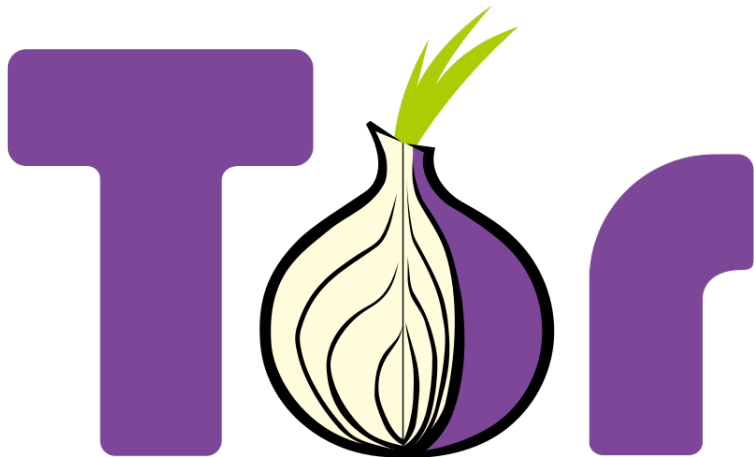


Figura 16: Logo

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Resistenza agli attacchi di analisi del traffico

- *onion* → messaggio incapsulato in diversi strati di crittografia (come strati di una cipolla)
- ogni *or sbuccia* via un singolo strato di crittografia, quello più esterno
- ogni *or*, con la decifratura, scopre il prossimo nodo intermedio destinatario
- ogni *or* conosce solo il nodo immediatamente precedente e immediatamente successivo
- il messaggio originale rimane nascosto per tutto il tragitto attraverso gli *or*
- nessun *or* conosce il percorso completo che il pacchetto ha preso

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Resistenza agli attacchi di analisi del traffico

- ogni pacchetto nel circuito viaggia protetto e cifrato con la chiave di un router
- nuovo scambio di chiavi crittografiche tra due nodi consecutivi, per ogni salto lungo il circuito
- ogni strato è relativo ad un nodo, che infatti riesce a *sbucciare* unicamente il proprio strato, ovvero lo strato più esterno della cipolla

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Resistenza agli attacchi di analisi del traffico

- ➊ ALICE invia messaggio a GUARD, protetto con la sua chiave: solo lui può capire strato più esterno
- ➋ GUARD legge che deve inviare il messaggio a MIDDLEMAN
- ➌ MIDDLEMAN riceve il messaggio: solo lui è in grado di capire il suo strato, che ora è il più esterno
- ➍ MIDDLEMAN legge che deve spedire il messaggio ad EXIT
- ➎ EXIT riceve il messaggio: solo lui è in grado di capire il suo strato, che ora è il più esterno
- ➏ EXIT legge strato più esterno, decifrabile solo da lui, e scopre che deve fare una richiesta esterna a BOB

# HTTPS, Proxy, VPN, TOR, p2p

TOR

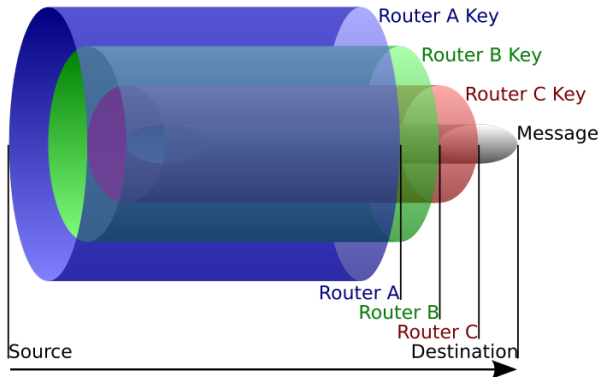


Figura 17: Onion Routing

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Resistenza agli attacchi di analisi del traffico

- 1 SOURCE invia una *onion* al router A, cioè un dato coperto da diversi strati di crittografia
- 2 A rimuove uno strato di crittografia: scopre nodo successivo a cui inviare il dato e da quale nodo proveniva *onion*
- 3 A invia *onion*, privata di uno strato, al router B
- 4 B decripta (*sbuccia*) un altro strato di crittografia
- 5 B invia *onion*, privata di un secondo strato, al router C
- 6 C rimuove lo strato di crittografia finale e trasmette quindi il dato finale decifrato a DESTINATION

# HTTPS, Proxy, VPN, TOR, p2p

## TOR

### Riferimenti

- <https://www.torproject.org/>
  - Tor Browser
  - Orfox
  - ...
- Facebook + TOR: <https://www.facebookcorewwi.onion>

# HTTPS, Proxy, VPN, TOR, p2p

## Reti p2p e privacy

### peer-to-peer

Gruppo di nodi all'interno di una rete che operano sia come client che come server: ogni nodo è in grado di svolgere in maniera paritetica le stesse operazioni

- *Napster-like*: indice dei servizi conservato da un server centrale
- *Gnutella-like*: indice dei servizi distribuito tra i vari peer
- indice servizi (chi fornisce che cosa) sostanzialmente pubblico
- fruizione di servizi tramite HTTP
- in alcuni casi (**BitTorrent**, ...) metadati contengono molte informazioni personali



# HTTPS, Proxy, VPN, TOR, p2p

## Reti p2p e privacy

### Freenet

Progetto che nasce proprio come tentativo di realizzare un sistema di pubblicazione di contenuti *copyright-resistant* (*resistente alle censure*):

- permette la pubblicazione e la fruizione di qualsiasi tipo di informazione
- peer-to-peer e completamente decentralizzato
- dati cifrati e replicati su molti nodi
- estremamente difficile sapere chi ha che cosa
- i singoli nodi non hanno modo di sapere cosa mettono a disposizione

# HTTPS, Proxy, VPN, TOR, p2p

## Reti p2p e privacy

### Architettura di Freenet

- ogni nodo mantiene una tabella dove associa i documenti immagazzinati con le chiavi di ricerca
- ogni contenuto identificato solo da un hash SHA-256
- *vicinato*: ogni nodo conosce soltanto i nodi che può raggiungere direttamente
- ogni nuovo nodo può essere *vicino* di un altro, non esiste struttura gerarchica
- contenuti passati ai vicini (memorizzati in una cache locale) senza sapere quale sia destinazione finale
- *key-based routing* euristico

# HTTPS, Proxy, VPN, TOR, p2p, p2p

## Reti p2p e privacy

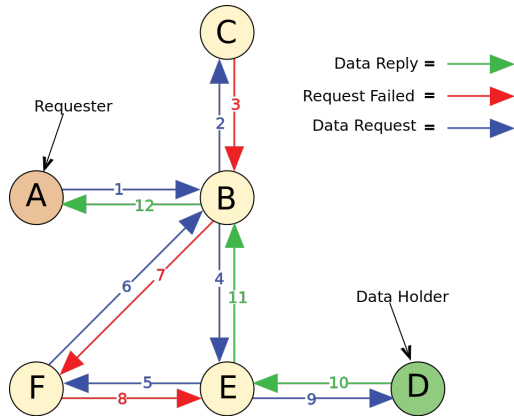


Figura 18: Freenet routing

# HTTPS, Proxy, VPN, TOR, p2p

Reti p2p e privacy

## Architettura di Freenet

- un nodo che inserisce un file nella rete può eventualmente disconnettersi: il file viene suddiviso e conservato tra i peer attivi
- contenuti maggiormente richiesti vengono inseriti più frequentemente nelle cache
- contenuti non richiesti tendono a sparire
- supporto di connessioni *Opennet* (chiunque può connettersi) e *Darknet* (rete tra trusted node)

# HTTPS, Proxy, VPN, TOR, p2p

## Reti p2p e privacy

### Invisible Internet Project (I2P)

Rete per servizi anonimi, con possibilità di gateway verso l'internet tradizionale:

- 2003: parziale spin-off di *Freenet* e *Invisible IRC*
- *overlay network*: la comunicazione avviene tramite I2Ptunnel (equivalenti ai circuiti TOR)
- tunnel cambiati ogni dieci minuti
- per usare I2P le applicazioni devono essere riscritte, utilizzando un'apposita API: Simple Anonymous Messaging oppure Basic Open Bridge

# HTTPS, Proxy, VPN, TOR, p2p

## Reti p2p e privacy

### eepsite

Sono i siti web di I2P e identificati da chiavi crittografiche, anziché normali indirizzi IP:

- i domini terminano con .i2p
- *eeproxy*: necessario per collegarsi con un normale browser a qualsiasi eepsite
- topologia della rete e risoluzione dei nomi simbolici avviene tramite un *Network Tracking Database (NetDB)*: una base di dati distribuita gestita con modalità DHT, simili a quelle viste per Freenet

# HTTPS, Proxy, VPN, TOR, p2p

Reti p2p e privacy

## Obiettivi e Criticità

I2P complementare a TOR: l'obiettivo è creare una rete alternativa il più possibile anonima

- attacchi *Sybil* → permettono di controllare il NetDB, gestendo una porzione di nodi
- molto facile da usare, ma livello di anonimato inferiore rispetto a TOR

# Esercizi

## Natas

0x00

- connettersi a `http://natas0.natas.labs.overthewire.org`: la password per ogni livello successivo deve essere trovata
- username: `natas0`
- password: `natas0`

### Soluzione

- vedere il sorgente dello script: la password per il livello successivo è nei commenti
- `<!--The password for natas1 is gtVrDuiDfck831PqWsLEZy5gyDz1clto -->`



# Esercizi

## Natas

### 0x00

- connettersi a `http://natas0.natas.labs.overthewire.org`: la password per ogni livello successivo deve essere trovata
- username: `natas0`
- password: `natas0`

### Soluzione

- vedere il sorgente dello script: la password per il livello successivo è nei commenti
- `<!--The password for natas1 is gtVrDuiDfck831PqWsLEZy5gyDz1clto -->`

# Esercizi

## Natas

### 0x01

- connettersi a  
`http://natas1.natas.labs.overthewire.org`
- username: natas1
- password: gtVrDuiDfck831PqWsLEZy5gyDz1clto

### Soluzione

- il sito non permette *right click* per vedere il file sorgente
- possibile controllare i sorgenti per mezzo dei *Web Developer Tools*
- `<!--The password for natas2 is  
ZluruAthQk7Q2MqmDeTiUij2ZvWy2mBi ->`

# Esercizi

## Natas

### 0x01

- connettersi a  
`http://natas1.natas.labs.overthewire.org`
- username: natas1
- password: gtVrDuiDfck831PqWsLEZy5gyDz1clto

### Soluzione

- il sito non permette *right click* per vedere il file sorgente
- possibile controllare i sorgenti per mezzo dei *Web Developer Tools*
- `<!--The password for natas2 is  
ZluruAthQk7Q2MqmDeTiUij2ZvWy2mBi ->`

# Esercizi

## Natas

### 0x02

- connettersi a  
`http://natas2.natas.labs.overthewire.org`
- username: natas2
- password: ZluruAthQk7Q2MqmDeTiUij2ZvWy2mBi

### Soluzione

- nel file sorgente dell'index c'è un tag ``
- `robots.txt`: `disallow /s3cr3t/`
- `/s3cr3t/users.txt`
- `natas4:Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ`

# Esercizi

## Natas

### 0x03

- connettersi a  
`http://natas3.natas.labs.overthewire.org`
- username: natas3
- password: sJIJNW6ucpu6HPZ1ZAchaDtwd7oGrD14

### Soluzione

- nell'index è presente il suggerimento: `<!-- No more information leaks!! Not even Google will find it this time... ->`
- `robots.txt`: `disallow /s3cr3t/`
- `/s3cr3t/users.txt`
- `natas4:Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ`

# Esercizi

## Natas

### 0x04

- connettersi a  
`http://natas4.natas.labs.overthewire.org`
- username: natas4
- password: Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ

### Soluzione

- nella home è presente l'avviso: Access disallowed. You are visiting from "" while authorized users should come only from  
`"http://natas5.natas.labs.overthewire.org/"`
- *Referer* di HTTP: cambiare valore in  
`http://natas5.natas.labs.overthewire.org/`



# Esercizi

## Natas

### 0x04

- connettersi a  
`http://natas4.natas.labs.overthewire.org`
- username: natas4
- password: Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ

### Soluzione

- nella home è presente l'avviso: Access disallowed. You are visiting from "" while authorized users should come only from  
`"http://natas5.natas.labs.overthewire.org/"`
- *Referer* di HTTP: cambiare valore in  
`http://natas5.natas.labs.overthewire.org/`