Sicurezza dei sistemi e delle reti[1]

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

a.a. 2013/14

Lezione XXI: Censura e controllo in rete

Censura e controllo in rete

Le reti telematiche risultano essere uno strumento di libertà, ma sono esposte al rischio di controllo di massa da parte dei *carrier* e dei governi.

- Censura
- Content filtering
- Tracking delle abitudini

Anche quando ci possono essere buone ragioni, i filtri possono essere imprecisi.

Diritti fondamentali

Dichiarazione universale dei diritti dell'uomo, art. 12

No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation.

Costituzione italiana, art. 15

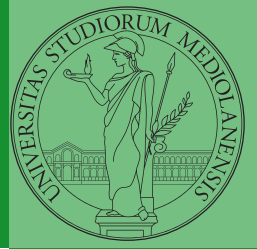
La libertà e la segretezza della corrispondenza e di ogni altra forma di comunicazione sono inviolabili. La loro limitazione può avvenire soltanto per atto motivato dell'autorità giudiziaria con le garanzie stabilite dalla legge.

Tutela

Gli utenti delle reti hanno quindi diritto di

- veder tutelati i loro diritti dalla legge
- usare la tecnologia in modo da difendersi all'interno di una rete (quindi in potenziale conflitto con l'amministratore della rete stessa)

Privacy-Enhancing Technologies (PET)



PET: tecnologia progettata allo scopo di tutelare la privacy (privatezza, riservatezza)

- Non solo reti: le porte dei bagni sono PET...
- In campo informatico:
  - tecniche per minimizzare o eliminare i *dati personali*
  - tecniche per evitare il controllo delle attività

Privacy e sicurezza

La privacy è importante anche per la sicurezza

- identity theft
- controllo e repressione del dissenso (piú efficace della tortura, vedi *The Man in the Snow White Cell*, CIA `http://ur1.ca/6lef8`)
- le persone cambiano, ma i dati restano (diritto all'oblio)

Minimizzazione dei dati

- Ogni servizio dovrebbe richiedere e raccogliere solo l'insieme minimo di dati necessario a fornirlo
- I dati personali (o addirittura *sensibili*) dovrebbero essere raccolti solo quando strettamente necessari (e conservati in maniera adeguatamente protetta)

Sanitization

La sanitization consiste nel eliminare dai dati le caratteristiche che li rendono personali o sensibili.

- Molto difficile: anche le aggregazioni statistiche dovrebbero risultare anonime
- L'anonimato richiede spesso grandi quantità di dati (es. un nero di 30-40 abitante a Dalvík, Islanda).

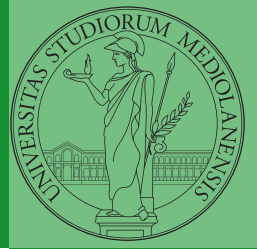
Protezione

Ogni volta che dati personali/sensibili sono conservati, processati o trasmessi dovrebbero essere protetti

- Controllo degli accessi
- Crittografia e *shredding*

In Italia norme di legge piuttosto precise: vedi Decreto legislativo 30 giugno 2003, n. 196 *Codice in materia di protezione dei dati personali*.

Riassumendo



La Rete è senz'altro uno strumento di libertà d'espressione, ma si presta a un controllo sistematico e potenzialmente oppressivo.

- Censura
- Content filtering
- Privacy

Lezione XXII: Anonimato in rete

Anonimato

La difesa rispetto ai pericoli di controllo è l'anonimato (non a caso tutti i tentativi di controllo politico di Internet cercano, in un modo o nell'altro, di limitare l'accesso anonimo)
Tema assai controverso, perché l'anonimato perfetto permette azioni non perseguibili (e in effetti in alcuni casi la legalità *locale* potrebbe essere in contrasto con i diritti fondamentali).

Inosservabilità

Un utente usa una risorsa senza che terze parti siano in grado di osservare l'uso
Per un evento E, Se O_A è l'insieme di eventi osservabili dall'attaccante A

Unobservability

$\forall \omega \in O_A : 0 < P(E|\omega) < 1$

Perché sia efficace $0 \ll P(E|\omega) \ll 1$

Inosservabilità perfetta

Se nessuna osservazione è in grado di cambiare la probabilità a posteriori di un evento, si parla di inosservabilità perfetta.

Perfect Unobservability

$$\forall \omega \in O_A : P(E) = P(E|\omega)$$

Incollegabilità

Un utente usa diverse risorse o servizi senza che sia possibile collegare i diversi usi.
Per due eventi E, F, con una caratteristica comune (link) $L_{E,F}$

Unlinkability

$\forall \omega \in O_A : 0 < P(L_{E,F}|\omega) < 1$

Perché sia efficace $0 \ll P(L_{E,F}|\omega) \ll 1$
Perfect Unlinkability: $\forall \omega \in O_A : P(L_{E,F}) = P(L_{E,F}|\omega)$

Incollegabilità di mittente e destinatario

Un caso particolare è l'unlinkability fra mittente e destinatario in una comunicazione.

- A, B comunicano
- A *comunicante* è osservabile, B *comunicante* è osservabile...
- ...ma non è osservabile il fatto che A comunica con B

Anonimato

Un utente usa una risorsa senza rendere nota la propria identità. Si definisce rispetto al ruolo $R_{U,E}$ dell'utente U nell'evento e un insieme W di identità

Anonymity

$\forall \omega \in O_A,\ \kappa \in W : 0 < P(R_{\kappa,E}|\omega) < 1$

In pratica deve essere $0 \ll P(R_{\kappa,E}|\omega) \ll 1$

Anonimato perfetto: $\forall \omega \in O_A,\ \kappa \in W : P(R_{\kappa,E}) = P(R_{\kappa,E}|\omega)$

Pseudonymity

Un utente usa una risorsa identificandosi con uno pseudonimo.

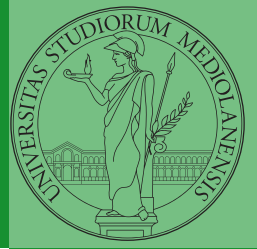
- Lo pseudonimo rimane costante
- ma non è possibile (o solo alcuni sono in grado di farlo) collegarlo all'identità reale
- può essere legato ad un ruolo

Riassumendo

La principale difesa rispetto ai pericoli di controllo è l'anonimato:

- Inosservabilità, unlinkability
- Anonimato e pseudonimi

Virtual Private Network



VPN

Una rete "overlay" che costituisce un dominio amministrativo sostanzialmente indipendente dalla topologia effettiva della rete sottostante

- Le comunicazioni sono criptate
- Molto usate per permettere a utenti roaming di accedere alle risorse delle reti aziendali

Tunnel

Si basano sul concetto di tunnel: ossia incapsulano i pacchetti in un altro protocollo (che rispetta la topologia della rete "fisica" sottostante)
Differiscono per il livello di rete virtuale che offrono.

Encryption

Anche la confidenzialità e integrità dei pacchetti può essere ottenuta a diversi livelli

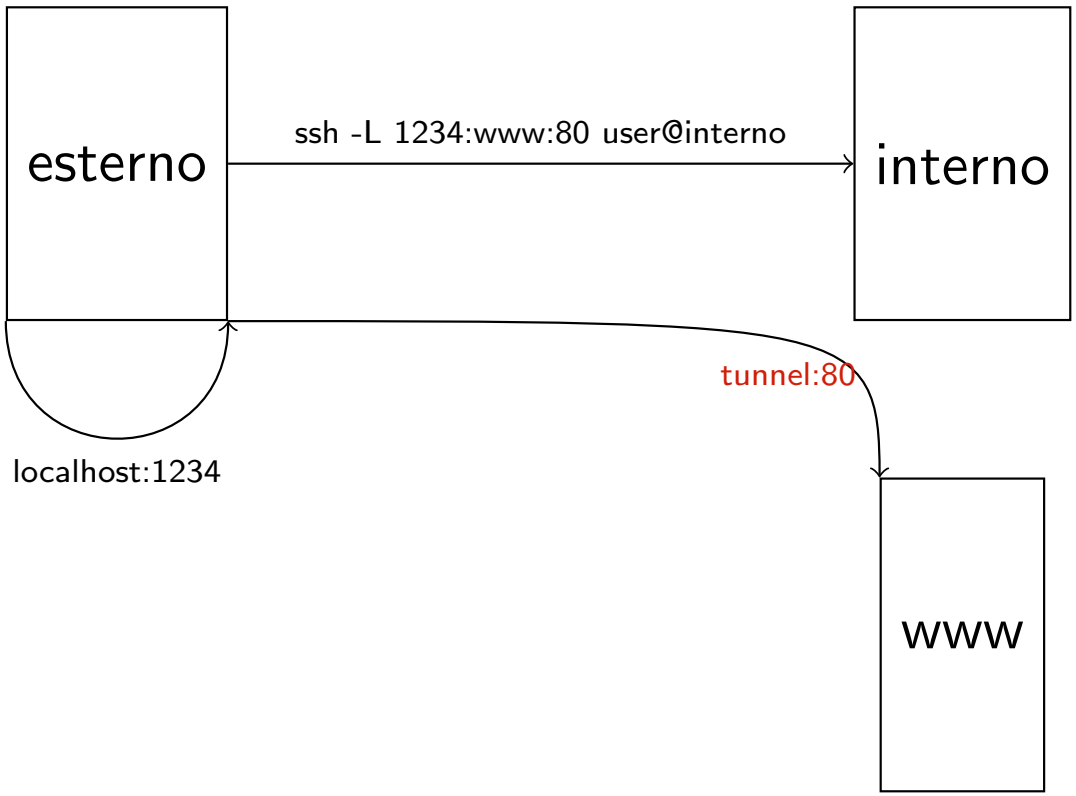
- IPSec
- SSL/TLS
- Protocolli proprietari
- SSH

SSH

OpenSSH può essere usato per fare tunneling di altri protocolli in SSH

- Port forwarding
- Vera e propria VPN

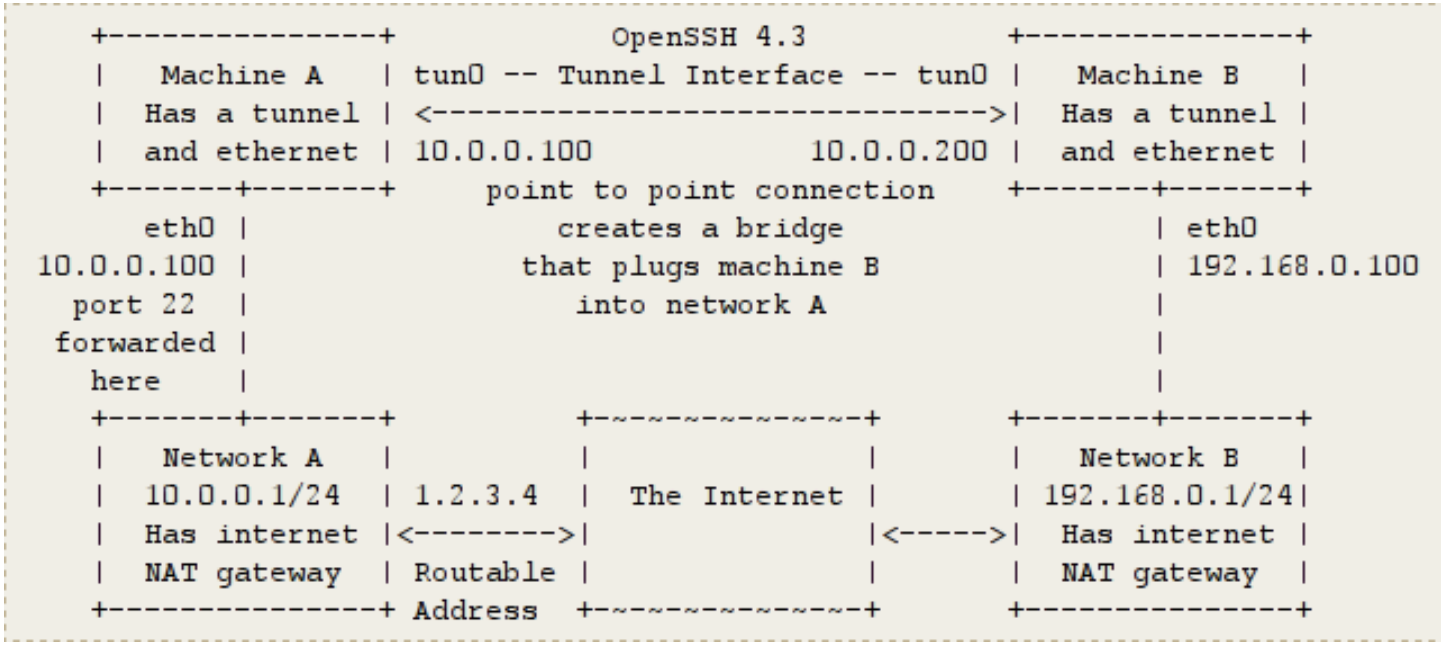
Port forwarding



www vede una connessione da interno, il pezzo criptato è solo esterno/interno

Una vera VPN con SSH

Per avere una vera VPN bisogna appoggiarsi sui livelli piú bassi dello stack. Con Linux esiste un device `tun` che serve proprio a questo scopo (Tunneling/NAT regolato da applicazioni utenti)



Setup di TUN/TAP

Se non è già attiva, bisogna creare l'interfaccia TUN/TAP
In Linux potrebbe essere necessario

```
1 mknod /dev/net/tun c 10 200
2 modprobe tun
```

Configurazione

```
# A
iface tun0 inet static
 address 10.0.0.100
 pointopoint 10.0.0.200
 up arp -sD 10.0.0.200 eth0 pub
# B
iface tun0 inet static
 pre-up ssh -f -w 0:0 1.2.3.4 'ifdown tun0; ifup tun0'
 address 10.0.0.200
 pointopoint 10.0.0.100
 up ip route add 10.0.0.0/24 via 10.0.0.200
 up ip route add 1.2.3.4/32 via 192.168.0.1
 up ip route replace default via 10.0.0.1
 down ip route replace default via 192.168.0.1
 down ip route del 10.0.0.0/24 via 10.0.0.200
 down ip route del 1.2.3.4/32 via 192.168.0.1
```

Riassumendo

Le VPN

- Permettono di usare una rete potenzialmente ostile, con alcune garanzie di confidenzialità e integrità
- Si basano sul tunnel di protocolli e possono essere realizzate in molti modi diversi

SSHuttle

Iptables + SSH tunnelling

```
1  Chain PREROUTING (policy ACCEPT 33 packets, 1604 bytes)
2   pkts bytes target prot opt in out source destination
3      33 1604 sshuttle-12300 all -- * * 0.0.0.0/0 0.0.0.0/0
4
5  Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
6   pkts bytes target prot opt in out source destination
7
8  Chain OUTPUT (policy ACCEPT 24 packets, 1568 bytes)
9   pkts bytes target prot opt in out source destination
10     48 3008 sshuttle-12300 all -- * * 0.0.0.0/0 0.0.0.0/0
11
12 Chain POSTROUTING (policy ACCEPT 48 packets, 3008 bytes)
13  pkts bytes target prot opt in out source destination
14
15 Chain sshuttle-12300 (2 references)
16  pkts bytes target prot opt in out source destination
17     0 0 RETURN tcp -- * * 0.0.0.0/0 127.0.0.0/8
18    24 1440 REDIRECT tcp -- * * 0.0.0.0/0 0.0.0.0/0 TTL match TTL != 42 redir ports 12300
```

OpenVPN

Un programma open-source che permette di costruire VPN basate su tunnel TCP o UDP. (La porta assegnata da IANA a OpenVPN è 1194)
Anche in questo caso l'implementazione sfrutta il driver TUN/TAP.

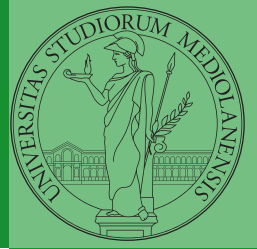
Autenticazione

L'autenticazione può avvenire in due modi

Static key pre-shared key

TLS tramite una sessione SSL/TLS (su UDP) con certificati e scambio di chiavi

Tunnel



Il tunnel può essere TCP (sconsigliato per ragioni di efficienza) o UDP (default)
Si può anche scegliere se usare l'interfaccia TUN (livello network) TAP (livello link).

Configurazione

Punto a punto, TAP (indirizzi assegnati con DHCP)

```
1 client
2 remote 172.16.0.3
3 dev tap
4 tls-client
5 ca client.crt
6 auth-user-pass
```

```
1 dev tap
2 remote 172.16.0.1
3 tls-server
4 ca server.crt
5 auth-user-pass-verify script
```

con la modalità server è possibile definire VPN con topologie complicate

Firewall

OpenVPN è facile da gestire con i firewall, perché usa un'unica porta (generalmente UDP 1194)

```
iptables -A INPUT -p udp --dport 1194 -j ACCEPT
```

E poi

```
iptables -A INPUT -i tun+ -j ACCEPT
```

(o l'equivalente con tap)

HMAC

Ogni pacchetto ha un message authentication code calcolato con HMAC, per cui i pacchetti non integri vengono scartati

$HMAC = H(K \oplus OPAD | H(K \oplus IPAD | m))$

(RFC2104: $OPAD = \texttt{0x5c5c5c} \ldots \texttt{5c5c}$, $IPAD = \texttt{0x363636} \ldots \texttt{3636}$))

- Autenticazione piú forte che basata su IP sorgente
- Ogni pacchetto che arriva a tun/tap è già stato controllato

Perché HMAC

È opportuno usare HMAC e non MAC piú semplici perché essendo i pacchetti di dimensione fissa sarebbero possibili attacchi del tipo hash extension, se l'hash è di tipo Merkle-Damgård (MD4-5, SHA0-2)

- $H(K|m)$: extension attack se nota la lunghezza di m
- $H(m|K)$: meglio, ma dipende fortemente dalle tecniche di collisione di H (birthday attack)
- $H(K|m|K)$ ancora meglio, ma manca una dimostrazione del grado di efficacia

Extension attack

Reso famoso da un attacco a Flickr (2009. . . vulnerabilità nota dal 1992!)

Se si sa $len(K|m)$ è possibile generare $H(K|m|m')$ senza conoscere K. Il motivo è che gli H Merkle-Damgård spezzano lo stream in blocchi e applicano una funzione di compressione ad ogni blocco indipendentemente.

Con lunghezza si sa quale funzione applicare a blocchi aggiunti dall'attaccante (estensione).

Riassumendo

OpenVPN permette di costruire VPN

- autenticazione con chiave condivisa o certificati
- tunnel UDP o TCP