Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure

Sicurezza dei sistemi e delle reti[1]

Mattia Monga

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

a.a. 2014/15

Sicurezza delle reti

Monga

DNS cache poisoning

DNSSEC

BGP

Vulnerabilità

Attacchi a BGP

Prefix hijacking

Prefix de-aggregation

Flapping attack

Contromisure

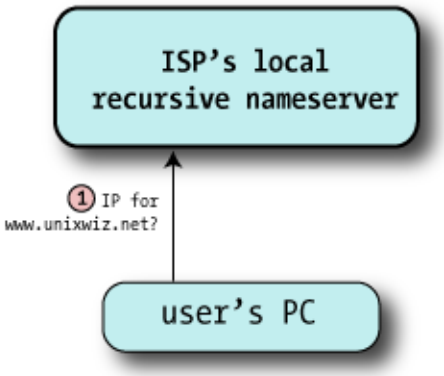
Lezione XVII: Attacchi al DNS

Risoluzione di una query

1. Si vuole risolvere `www.example.com`
2. Richiesta al server DNS configurato
3. Il DNS esamina se è risolvibile localmente o se la query è *ricorsiva:* in questo caso consulta l'elenco dei root server che conosce
4. Il root DNS non conosce l'indirizzo, ma risponde con un record di tipo `NS` corrispondente ai Global Top Level Domain (gTLD) server di `.com`
5. si ripete fin quando si ottiene il ns autorevole (authoritative) per *example.com*

Risoluzione

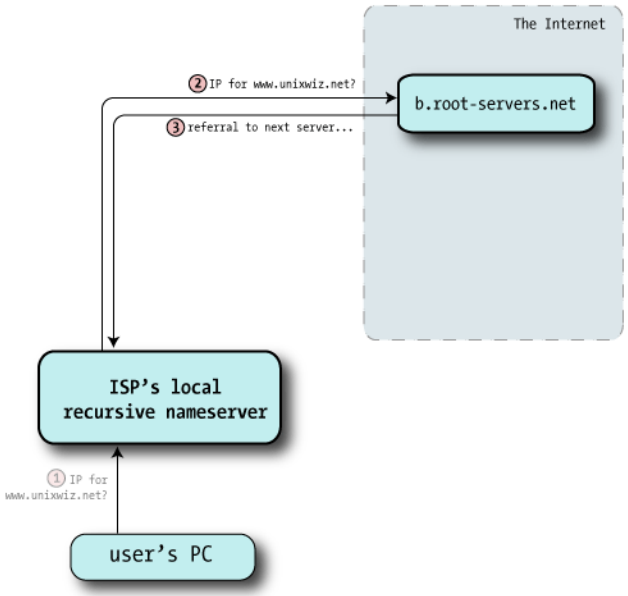
da http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html



L'utente chiede la risoluzione di `www.unixwiz.net` al DNS del proprio ISP (`dnsr1.sbcglobal.net`)

Risoluzione

da http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html



13 root server

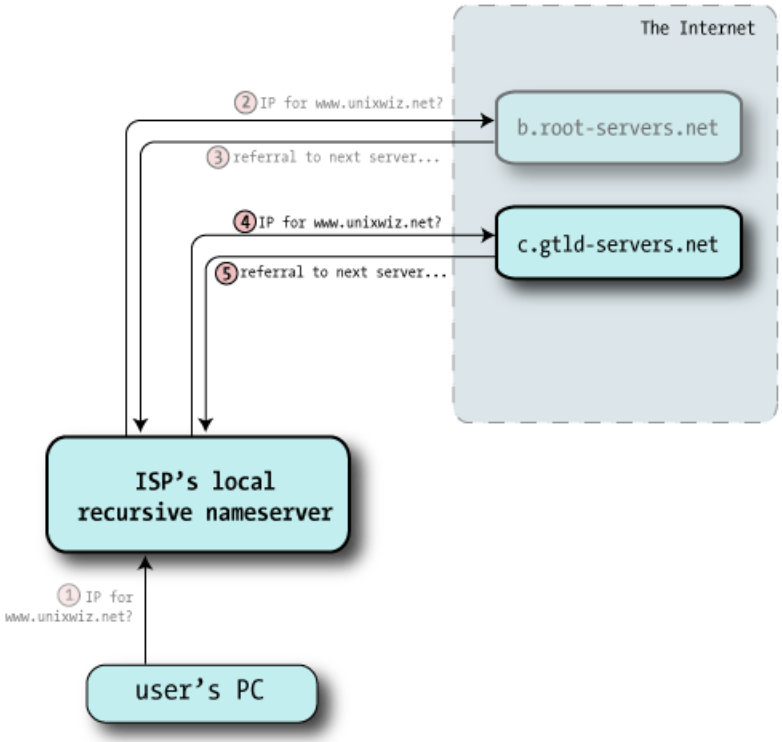
```
A.ROOT-SERVERS.NET.   IN   A   198.41.0.4
B.ROOT-SERVERS.NET.   IN   A   192.228.79.201
C.ROOT-SERVERS.NET.   IN   A   192.33.4.12
...
M.ROOT-SERVERS.NET.   IN   A   202.12.27.33
```

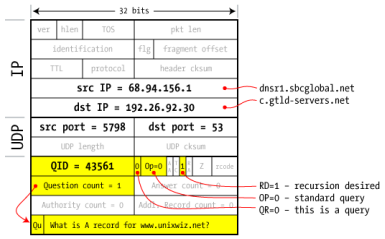
e i name server di .net

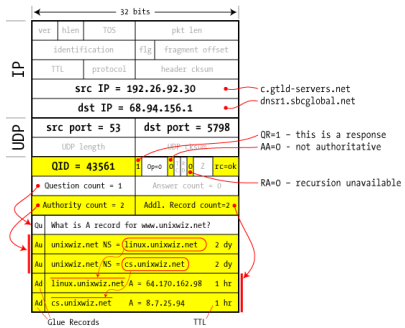
```
/* Authority section */
NET.                  IN   NS A.GTLD-SERVERS.NET.
                      IN   NS B.GTLD-SERVERS.NET.
                      IN   NS C.GTLD-SERVERS.NET.
                      ...
                      IN   NS M.GTLD-SERVERS.NET.

/* Additional section - "glue" records */
A.GTLD-SERVERS.net.   IN   A   192.5.6.30
B.GTLD-SERVERS.net.   IN   A   192.33.14.30
C.GTLD-SERVERS.net.   IN   A   192.26.92.30
...
M.GTLD-SERVERS.net.   IN   A   192.55.83.30
```

Risoluzione

da http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

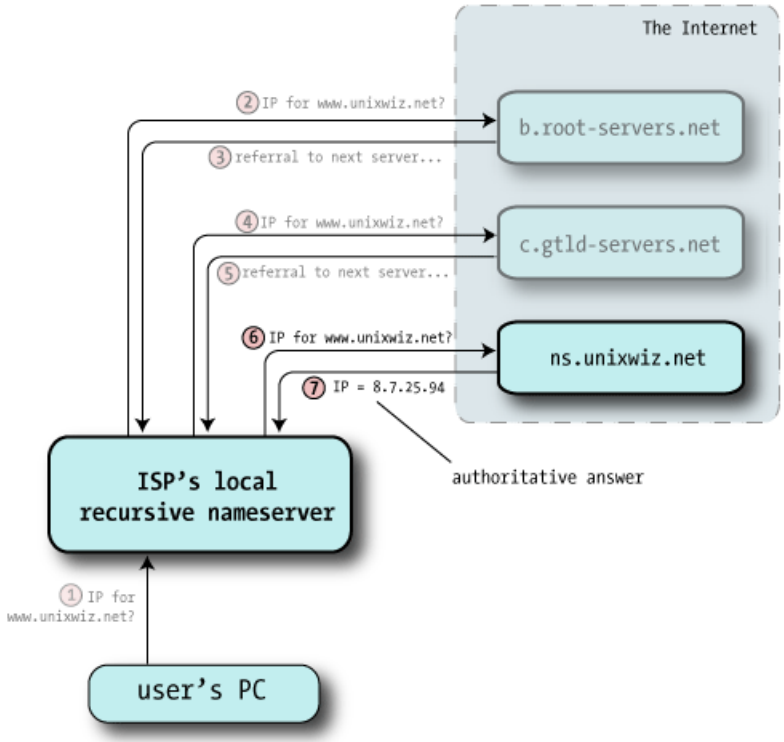


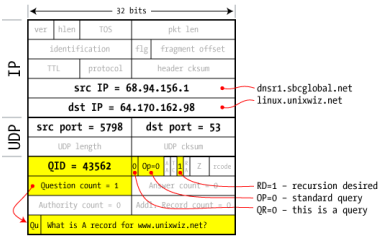


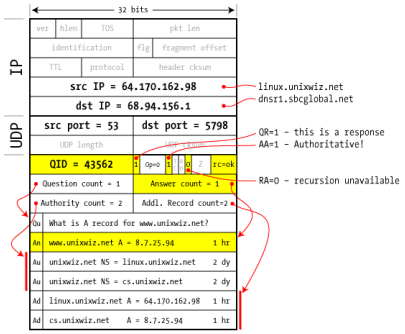


Risoluzione

da http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

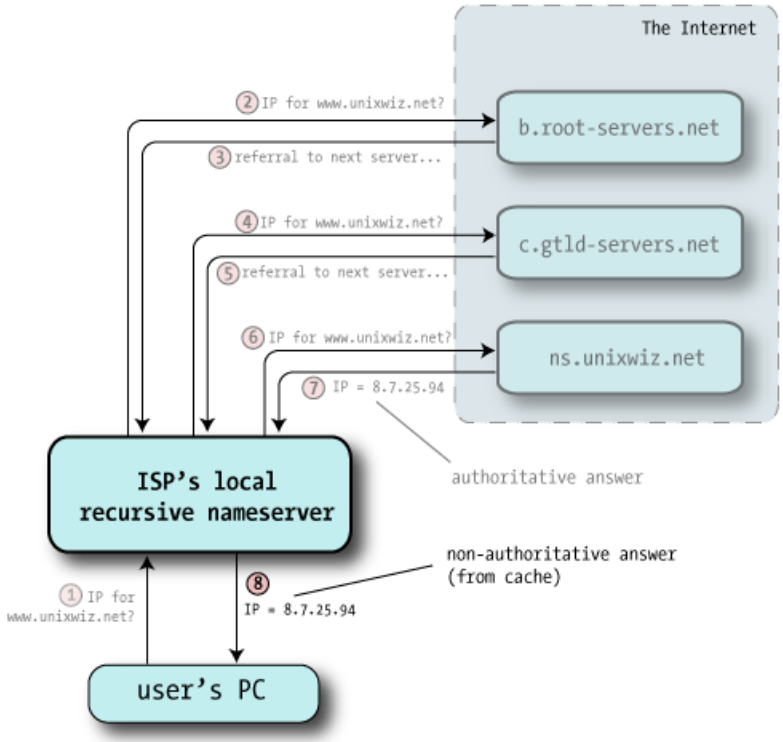






Risoluzione

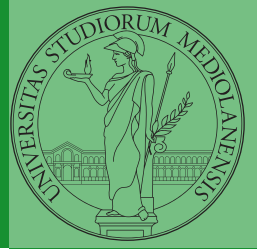
da http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html



Il numero IP cercato è `8.7.25.94`. Questo dato può essere conservato (per un tempo pari al TTL) in una cache locale del name server ricorsivo per rendere piú efficiente il processo.

Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure

Fragilità del sistema



Sicurezza delle reti

Monga

DNS cache poisoning

DNSSEC

BGP

Vulnerabilità

Attacchi a BGP

Prefix hijacking

Prefix de-aggregation

Flapping attack

Contromisure

Il sistema è estremamente efficiente e piuttosto resistente ai guasti, ma nella versione originaria non prevede nessuna tecnica di autenticazione e integrità delle informazioni.

Fragilità del sistema

- Il name server x di `unixwiz.net` potrebbe ospitare le associazioni per i nomi della *zona* `bancaditalia.com`, anche se nessun gTLD ne delegherebbe la risoluzione a x
- Un attacco possibile è l'avvelenamento della cache (cache poisoning)

Cache poisoning

Un attaccante riesce ad alterare la cache di un DNS ricorsivo, che pertanto restituisce un'associazione scorretta.
Come fa il DNS ricorsivo ad "autenticare" la risposta che riceve da `ns.unixwiz.net`?

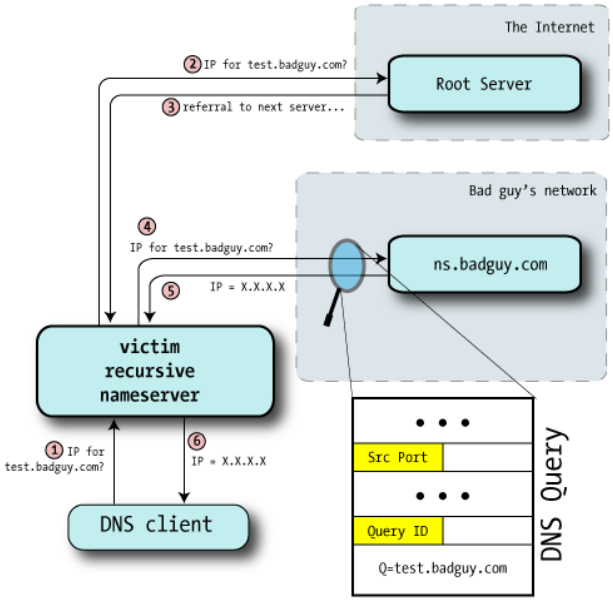
Cache poisoning

1. La risposta deve arrivare con la stessa porta UDP sorgente della richiesta. altrimenti viene scartata
2. La sezione Question coincide con quella della richiesta
3. Il query ID corrisponde a quello della richiesta
4. La risposta contiene dati riguardanti nodi nella zona (non `bancaditalia.com` per esempio)

Se l'attaccante riesce a prevedere questi dati, può alterare la cache

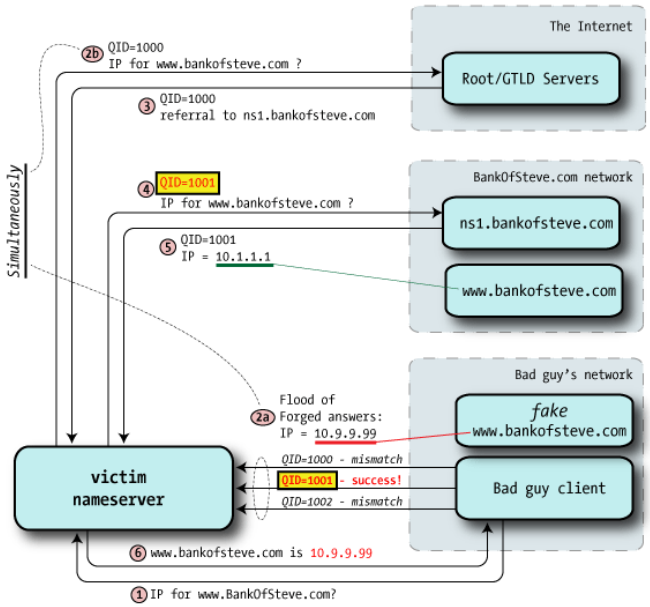
Come indovinare il Query ID?

Spesso è semplicemente un contatore, quindi basta intercettare il traffico di richieste lecite



Caso semplice

Se la porta UDP utilizzata è sempre la stessa (cosí in molte implementazioni) l'attacco è semplice



Caso semplice

Ovviamente non funziona se il nome è già nella cache. Le chance dell'attaccante sono minori se il dns authoritative è piú *vicino* al dns vittima.

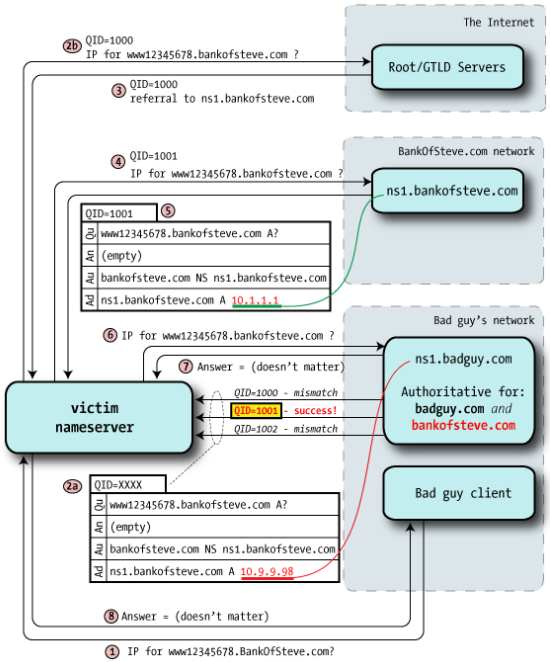
Difesa

La difesa principale è la randomizzazione del query ID

- Con ID sequenziali l'attaccante prova una ventina di ID
- Con ID random (su 16 bit) occorre provare 64K (prima che il dns authoritative risponda)

L'attacco di Dan Kaminsky

L'idea di base è la stessa, ma amplia l'impatto falsificando il dns authoritative stesso. L'attaccante ne allestisce uno proprio, che però normalmente non riceverebbe richieste.



Chance dell'attacco

Con la randomizzazione dei query ID sembra difficile fare le 64K prove necessarie in tempo utile (prima che arrivi la vera risposta).
Ma nella versione Kaminsky l'attaccante genera tanti nomi casuali (p.es. `www12345678.bankofsteve.com`).

Chance dell'attacco

Anche se riesce a fare solo poche (50?) risposte finte prima di essere superato dal vero authoritative, può comunque ripetere questo tentativo tante volte con nomi diversi: ogni tentativo ha probabilità di riuscita $\frac{50}{65536}$

Con 100 tentativi: $7,3\%$, con 1000: $53,4\%$, con 10000: $99,9\%$

Un possibile miglioramento si ha randomizzando anche la porta UDP.

Se la porta è random su 65535 ci vogliono almeno $60 \cdot 10^6$ tentativi.

(MS DNS server sceglie fra 2500 porte, quindi in realtà "bastano" $2,3 \cdot 10^6$)

Riassumendo

Il protocollo DNS

- non prevede autenticazioni nelle query
- l'associazione query/risposta si basa sul numero di porta
- se le porte sono prevedibili, si possono facilmente avvelenare le cache dei DNS

DNSSEC

Sicurezza delle reti

Monga

DNS cache poisoning

DNSSEC

BGP

Vulnerabilità

Attacchi a BGP

Prefix hijacking

Prefix de-aggregation

Flapping attack

Contromisure

DNSSEC è uno standard retro-compatibile che aggiunge autenticazione e controllo d'integrità alle query DNS. Prima versione 1997, rivisto nel 2005 e nel 2008.

DNSSEC

- È considerato un elemento fondamentale nelle strategie globali della cosiddetta *trusted* Internet
- In realtà la sua adozione langue:
  - Complessità delle configurazioni
  - Aumento del traffico
  - Perplessità di una parte della comunità sull'efficacia

Nel 2009 risultavano 274 domini firmati su circa 80'000'000 di `.com`

DNSSEC, punti fondamentali

Il concetto fondamentale è che le risposte dei DNS authoritative sono firmate digitalmente

- La chiave pubblica di una zona viene distribuita dalla zona gerarchicamente superiore (la chiave pubblica di `.net` è distribuita da un root server, ecc.)
- C'è la possibilità di avere risposte di non esistenza ("Authenticated denial of existence")

Complicazioni

Il deployment è reso complicato soprattutto dall'esigenza di ruotare le firme che scadono (di solito ogni 30 giorni), per evitare *replay attack*.

Limiti di DNSSEC

Secondo D. J. Bernstein, autore di `djbdns` e di una proposta alternativa (DNSCurve), è mal progettato:

- L'assunzione di base è che non è pensabile usare la crittografia in ogni pacchetto, per ragioni di efficienza.
- Non c'è crittografia dei dati (solo integrità)
- Le firme sono precalcolate (e quindi occorre ruotare le chiavi per limitare i replay)
- Tutti i tool per la modifica dei dati DNS devono essere 'signature aware'
- Non c'è protezione contro DoS

Ulteriori pericoli

Bernstein identifica anche vulnerabilità di DNSSEC

1. Ogni pacchetto di risposta DNS è firmato: se i dati sono alterati andrebbe scartato
   - So che i dati potrebbero essere falsi, ma non ho comunque i dati veri (denial of service)
   - Di fatto, al momento la maggior parte dei server non lo fa

Ulteriori pericoli

2. Vengono firmate solo le associazioni di cui un ns è authoritative: i glue record rimangono falsificabili
3. Il protocollo permette l'amplificazione di DDos (una query di 78 byte si può trasformare in una risposta da 3113 byte)

Riassumendo

DNSSEC è un protocollo che cerca di rendere sicura la risoluzione

- forte pressione per l'adozione
- richiede una complessa gestione delle chiavi
- non risolve tutti i problemi

L'infrastruttura di Internet

Internet è una reti di reti locali.
Il routing a livello globale, però è gestito fra Autonomous System (AS), insiemi di reti locali con un'autonomia amministrativa.

In un AS valgono routing policy specifiche, non necessariamente concordate con gli altri
Il routing fra AS è affidato a protocolli particolari.

BGP

Il Border Gateway Protocol è un protocollo usato per il routing fra AS

- path vector: l'instradamento è fatto conoscendo una serie di path
- le decisioni non sono prese con riferimento alle "distanze", ma a politiche di routing

Testo di riferimento: A. Wong, A. Yeung, *Network Infrastructure Security*, Springer

Concetti di BGP

- I nodi indirizzabili da un AS sono quelli con un determinato *prefisso*
- Un *AS path* è la lista degli AS da attraversare per raggiungere un nodo con un dato prefisso
  1. Un AS A annuncia (`UPDATE`) ai vicini quali prefissi x sa indirizzare ($A\,x$)
  2. Il vicino B annuncia ($B\,A\,x$)
  3. Chi riceve un path che contiene sè stesso non lo riannuncia
  4. I path contengono anche attributi utilizzabili nelle policy

BGP

Le comunicazioni BGP fra AS avvengono tramite una connessione TCP (porta 179). I principali pericoli sono:

- Alterazione dei dati di routing (subverted link)
- Router maligni (subverted router)

Attacchi esterni

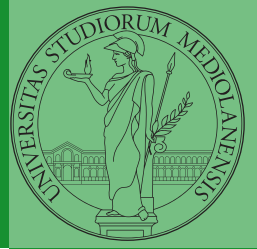
Dai subverted link ci si può difendere con un'infrastruttura a chiavi asimmetriche (non presente nel protocollo di base). Come al solito, la gestione delle PKI è complessa, ma molto efficace. (Non difende dall'*interruzione* del collegamento, naturalmente)

Attacchi interni

Un router maligno, per:

- compromissione
- spoofing (se non c'è PKI)
- mal configurato

Vulnerabilità di BGP



Senza opportune precauzioni (estensioni PKI), BGP:

- non prevede autenticazione della sorgente, né integrità dei messaggi
- non c'è controllo sull'ownership dei prefissi
- non c'è controllo sulle informazioni di path

Controlli di coerenza

A volte è possibile rilevare un'incoerenza nelle informazioni di routing

- non sono necessariamente dovute a compromissioni
- quasi mai si riesce a determinare l'informazione corretta
- se l'attaccante conosce la topologia della rete, generalmente può produrre informazioni false, ma coerenti

Riassumendo

Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure

Il protocollo BGP per il routing fra AS

- È un protocollo path vector che permette di fare routing in base a policy complesse (non solo secondo la "distanza")
- Nella versione base non prevede garanzie di sicurezza

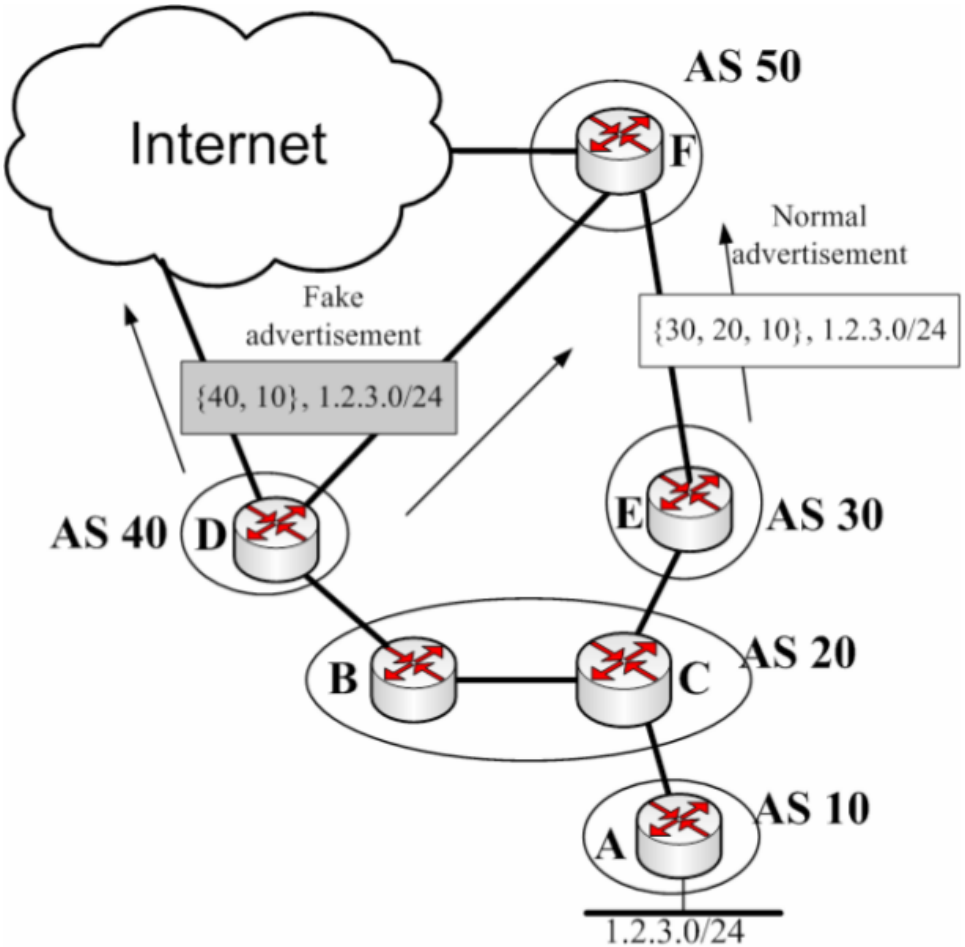
Obiettivi degli attacchi

La falsificazione delle informazioni di routing può servire per

- Redirezione del traffico
- Instabilità del routing
- Black hole

Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure

Prefix Hijacking



- attaccante D
- Fa finta di controllare il prefisso di A
- Se $AS\,50$ preferisce i path corti, D ha successo nella redirezione

D potrebbe anche attribuirsi i prefissi di $AS\,20$

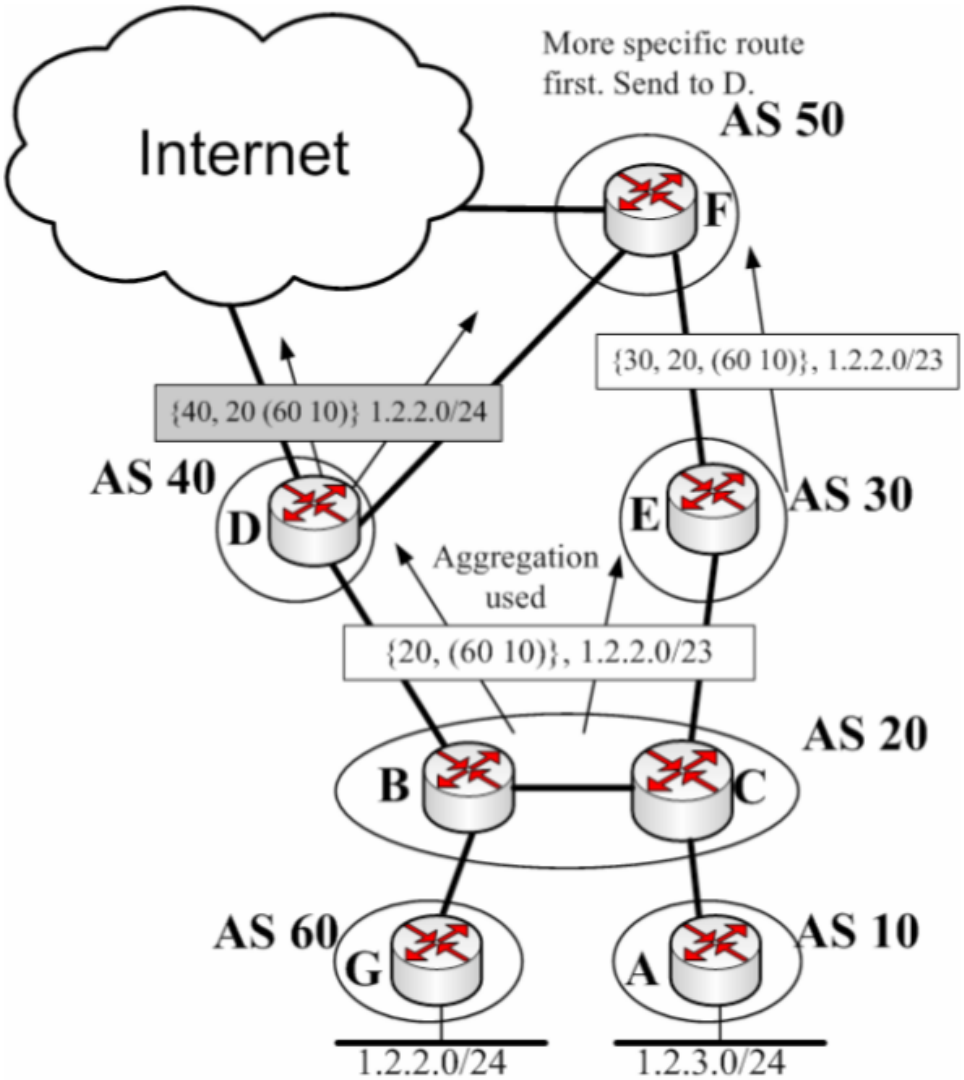
Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure

Prefix aggregation

Quando piú prefissi condividono un certo numero di bit è conveniente aggregarli

- `10.42.2.0/24` e `10.42.3.0/24` condividono i primi 23 bit
- aggregati in `10.42.2.0/23` (o `10.42.3.0/23`) permettono di accorciare i path
- allo scopo si usa un *AS set*

Prefix De-aggregation



- attaccante D
- $AS\,50$ riceve da $AS\,40$ una rotta piú specifica
- Il traffico passa per D

D potrebbe anche attribuirsi il prefisso `1.2.3.0/24`

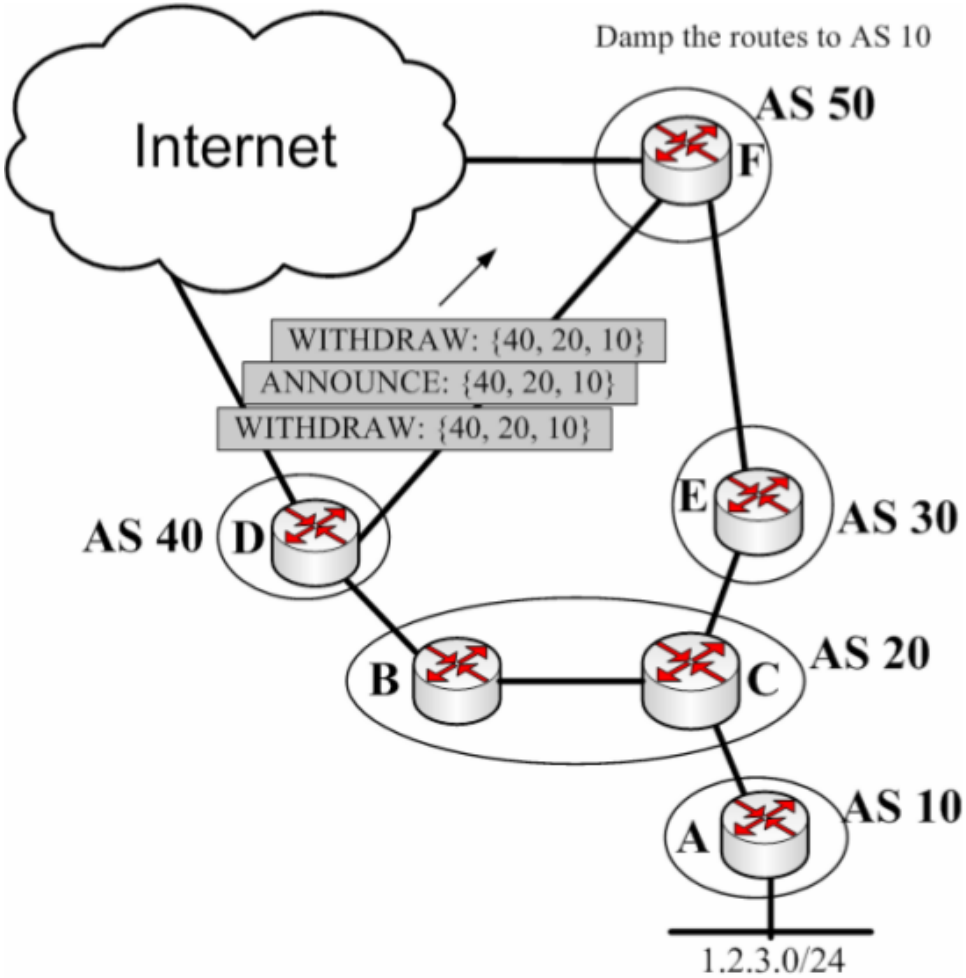
Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure

Route flapping

A livello Internet è perfettamente normale avere una topologia estremamente dinamica: BGP permette di scartare e annunciare nuove rotte con facilità.

- link flapping un link viene disattivato e poi riattivato (normale)
- Se succede spesso però, crea instabilità nella rete perché gli instradamenti sono in continua variazione
- route damping la riattivazione di una rotta viene accettata con tempi crescentemente piú lunghi

Flapping attack

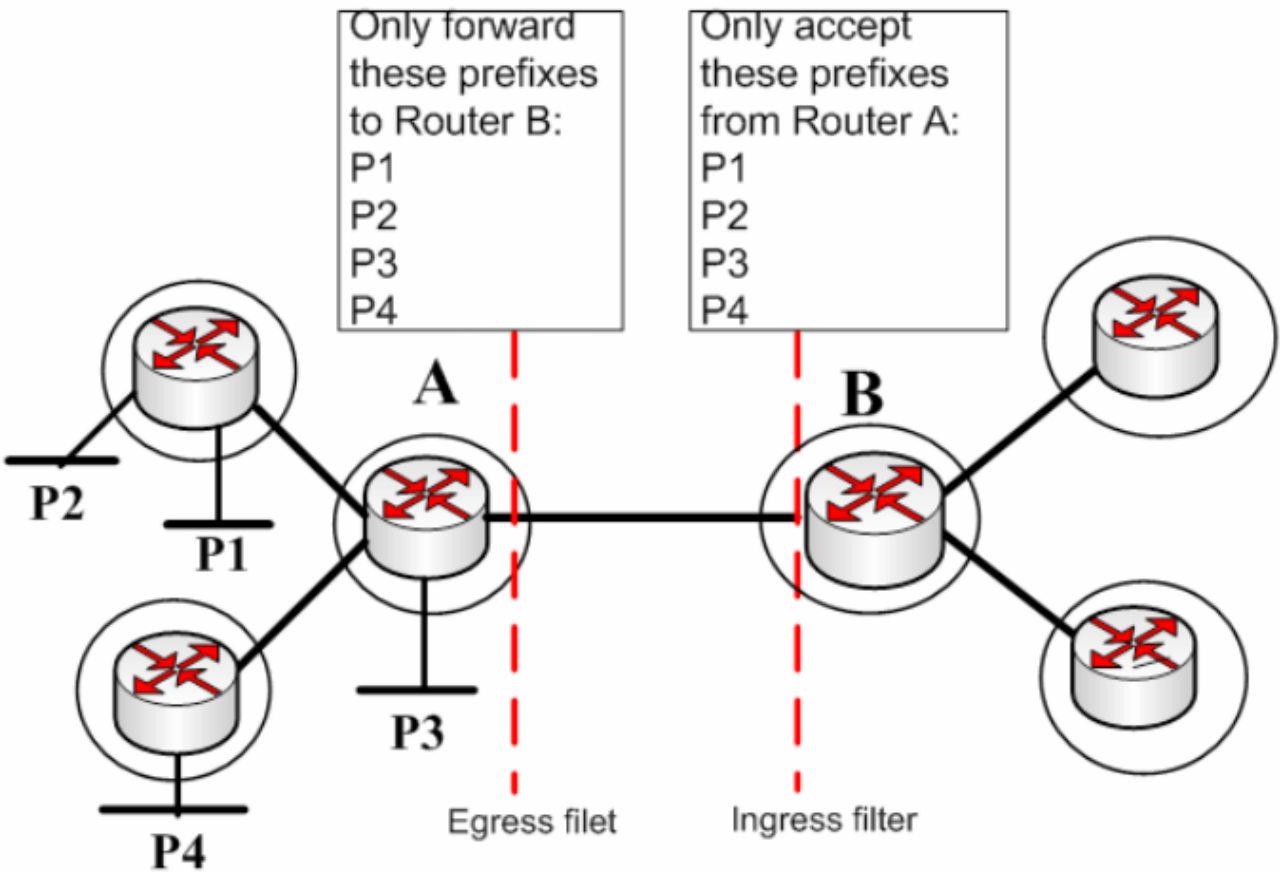


- attaccante D
- $AS\,50$ si convince che il link è flapping
- $AS\,10$ diventa irraggiungibile da $AS\,50$ a causa del damping

Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure

Contromisure

La contromisura piú semplice è l'attivazione di filtri ingress e egress che scartano i path relativi a prefissi "imprevisti" Internet Routing Registry (IRR) (http://www.irr.net)



Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure

Versioni sicure di BGP

Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure

Ci sono diverse evoluzioni sicure di BGP

- S-BGP: PKI e IPsec
- Secure Origin BGP (Cisco): PKI, nuovi messaggi BGP
- IRV: indipendente dal protocollo (non solo BGP), basta un livello di trasporto sicuro

Riassumendo

Il protocollo BGP senza precauzioni è vulnerabile

- Prefix hijacking
- Prefix de-aggregation
- Flapping attack

Sicurezza delle reti
Monga
DNS cache poisoning
DNSSEC
BGP
Vulnerabilità
Attacchi a BGP
Prefix hijacking
Prefix de-aggregation
Flapping attack
Contromisure