DNS cache poisoning e Bind

Il Domain Name System è fondamentale per l'accesso a internet in quanto risolve i nomi degli host nei corrispondenti numeri IP. Se qualche malintenzionato riesce a cambiare la risoluzione di un nome (ad esempio windowsupdate.microsoft.com) le conseguenze potrebbero essere almeno spiacevoli. In questo articolo consideriamo un caso particolare di attacco al protocollo DNS, bisogna infatti osservare che alla base di questi attacchi vi è una debolezza intrinseca del protocollo stesso. Per capire come siano possibili questi tipi di attacchi e perciò cosa si può fare per prevenirli, dobbiamo spiegare un attimo come funziona una parte del protocollo DNS.

1. Come funziona

Un server DNS di norma svolge due funzioni[1]

1. essere *autoritativo* per alcuni domini, cioè avere le tavole di traduzione da nome a numero IP (e viceversa) per tutti nomi di host appartenenti a questi domini; spesso queste tavole sono create e modificate manualmente in file di testo sul server DNS
2. risolvere i nomi in numeri IP per un certo numero di propri client, ad esempio gli host appartenenti ai domini di cui è autoritativo.

Vi sono due modi di fare una ricerca di risoluzione di nome in numero IP:

1. i client fanno delle richieste *ricorsive* al proprio server DNS, in questo caso il server DNS si incarica di contattare tutti i server DNS necessari sino a che non trova quello autoritativo per il nome o numero cercato
2. i server DNS fanno delle richieste *non-ricorsive* verso altri server DNS, in questo caso se il server interrogato non conosce la risposta, indica un altro server che potrebbe conoscerla.

Ad esempio, un client che cerca la risoluzione del nome host.mio.dominio.it, fa una richiesta al proprio server DNS che a sua volta contatta nell'ordine: un root-server, il server della zona .it., il server della zona dominio.it., il server della zona mio.dominio.it che fornisce la risposta, che a sua volta il nostro server DNS invia al client. Ovviamente il nostro server DNS è un po' più furbo ed agisce secondo la seguente logica quando riceve una richiesta da un client:

1. se è autoritativo per il nome richiesto poiché la risoluzione è nei file di configurazione del server, invia la risposta direttamente al client
2. se la risoluzione è presente nella propria cache (la stessa richiesta è stata fatta da qualcun altro precedentemente) invia la risposta direttamente al client; è da notare che le risoluzioni vengono

1 In sistemi medio grandi le funzioni possono essere divise e distribuite su diversi server, anche per aumentare la sicurezza del sistema.

di solito mantenute in cache per un periodo da un giorno ad una settimana

3. altrimenti contatta in modo non-ricorsivo i server DNS che potrebbero conoscere la risposta (ovviamente non è necessario ripartire tutte le volte dai root-server) ed una volta ottenuta la risoluzione la invia al client e la inserisce nella cache.

2. L'attacco

Un possibile, e purtroppo spesso usato, attacco consiste nell'inserire nella cache di un server DNS la risoluzione fasulla di un nome. Non entreremo qui nei dettagli tecnici di come è possibile fare questo, ma illustreremo solo lo schema d'azione dell'attaccante:

1. l'attaccante ha un client in grado di fare richieste di risoluzione al server DNS sotto attacco ed invia molte richieste contemporaneamente per la risoluzione fasulla che vuole inserire nella cache del nostro server DNS
2. se la risposta è già nella cache del server DNS l'attacco fallisce poiché il server risponde con l'informazione presente in cache e non carica la risoluzione fasulla
3. se la risposta non è nella cache, l'attaccante invia la risoluzione fasulla al nostro server DNS mentre questo sta interrogando i server DNS autoritativi.

Dalla descrizione è ovvio che vi sono due punti su cui gioca l'attaccante:

1. una *race-condition* per cui, sapendo l'attaccante quando il server DNS effettuerà la richiesta di risoluzione verso i server DNS autoritativi visto che è l'attaccante stesso la sorgente della richiesta di risoluzione, l'attaccante cerca di inviare la risposta prima che arrivi dai server autoritativi
2. l'attaccante deve prendere l'*identità* del vero server autoritativo in modo che il server DNS sotto attacco accetti la risposta fasulla.

Il primo punto è facilmente risolvibile per l'attaccante se al tempo stesso dell'attacco al nostro server DNS fa anche un attacco di Denial-of-Service sul vero DNS autoritativo. In questo modo l'attaccante ha tutto il tempo necessario per inviare la risoluzione fasulla al nostro server DNS. Le tecniche per il secondo punto sono più complesse, discuteremo più avanti i punti di attacco e come fare a proteggersi.

3. Cosa succede se l'attacco va a buon fine?

Se l'attacco va a buon fine, l'attaccante può utilizzarlo per molti scopi. Ne citiamo solo due quali esempi:

1. l'attaccante può preparare un sito identico a windowsupdate.microsoft.com, inserire nella cache del nostro server DNS la risoluzione di questo nome al proprio sito e far scaricare a tutti i client

del nostro server DNS del codice maligno invece degli updates di Microsoft

2. l'attaccante può inserire nella cache del nostro server DNS la risoluzione di un dominio utilizzato a scopi illegali, ad esempio pedo-pornografia, e poi indicare il nostro server come autoritativo per questo dominio; in questo modo tutti gli host in internet interrogheranno il nostro server DNS per risolvere questo dominio e prima o poi saremo coinvolti in una indagine di polizia per il reato di pedo-pornografia.

E' da notare che il secondo metodo è molto usato dagli spammer per creare un falso dominio che gli permetta di inviare i messaggi di posta elettronica. Va notato anche che, come detto, i dati rimangono nella cache per un tempo limitato e l'attaccante deve rifare il proprio attacco una volta che le risoluzioni fasulle sono state cancellate dalla cache.

4. Come difendersi

Recentemente sono stati introdotti i protocolli DNSSEC e TSIG che dovrebbero garantire la sicurezza delle comunicazioni tra server DNS, ovvero evitare che l'attaccante possa prendere l'identità del server autoritativo. Purtroppo questi protocolli sono ancora poco adottati anche perché sono ancora giovani e poco sperimentati. Nel frattempo possiamo migliorare la nostra protezioni adottando le seguenti procedure:

1. se utilizziamo bind, è sicuramente consigliato di passare alla versione 9 ed in generale, qualunque sia il software adottato, è consigliato utilizzare l'ultima versione (stabile) disponibile in quanto gli sviluppatori stanno introducendo tecniche che rendono più difficile all'attaccante inviare pacchetti fasulli che il nostro server DNS accetti[2]
2. è assolutamente necessario configurare la nostra rete in modo che al nostro server DNS non possano arrivare pacchetti con indirizzo IP sorgente di un nostro client inviati da un attaccante da internet, in altre parole che vi siano rigidi filtri *anti-spoofing*
3. limitare l'accesso in modalità ricorsiva al server DNS ai soli numeri IP dei veri client e non, come è cattiva abitudine, lasciare l'accesso in modalità ricorsiva a tutto il mondo.

I punti 2 e 3 garantiscono che solo un nostro client, e non un attaccante esterno che utilizza come IP sorgente quello di un nostro client, sia abilitato ad inviare una richiesta ricorsiva al nostro server DNS. In questo modo solo i nostri client possono popolare la cache del nostro server DNS. Ovviamente questa configurazione lascia libero di agire un attaccante interno, ovvero che è un nostro client legittimo o qualcuno che si è appropriato di un nostro client. Nella tabella 1 riportiamo un parziale esempio di configurazione di bind 9 che soddisfa alle richieste del punto 3. Si noti che il default per tutte le `query` (nella sezione `options`) è ristretto ai nostri client, e solo

2 In particolare si cerca di utilizzare porte UDP sorgenti casuali e numeri casuali per le transaction-ID.

i domini per cui il server è autoritativo sono disponibili a tutti.

Andrea Pasquinucci
Libero Professionista in Sicurezza Informatica
pasquinucci@ucci.it

Riferimenti Bibliografici:

[1] Lo standard per il DNS è negli RFC 1034, 1035 e molti aggiornamenti
[2] *Attacking the DNS Protocol*, SANS Institute, http://sainstitute.org/articles/AttackingtheDNSProtocol.pdf
[3] Joe Stewart, *DNS Cache Poisoning - The Next Generation*, SecurityFocus, http://www.securityfocus.com/guest/17905
[4] Un esempio di configurazione di bind-9 di Rob Thomas http://www.cymru.com/Documents/secure-bind-template.html
[5] P. Albitz e C. Liu, *DNS and Bind,* O'Reilly

```
// bind 9 – PARTIAL simple config example - DO NOT USE AS IT IS
acl our-customers {
        localhost;
        192.168.1.0/24;
};
options {
        allow-query { our-customers; };
        allow-recursion { our-customers; };
};
// for recursive queries
zone "." IN {
        type hint;
        file "root.servers";
};
zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};
zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "named.local";
        allow-update { none; };
};
// Our authoritative domains
zone "my.domain.it" IN {
        type master;
        file "mydomain";
        allow-update { none; };
        allow-query {any;};
};
zone "1.168.192.in-addr.arpa" IN {
        type master;
        file "192.168.1";
        allow-update { none; };
        allow-query {any;};
};
```

Tabella 1. Esempio parziale di configurazione di Bind-9