Corrado Giustozzi è uno dei maggiori esperti italiani di Sicurezza Informatica

Verso una Rete inaffidabile?

Il DNS, il sistema da cui dipende il funzionamento di tutta Internet, è a rischio. Una vulnerabilità di recente scoperta ne affligge il corretto funzionamento. Il problema è stato (per ora) risolto. Ma in futuro?

Nello scorso mese di luglio è rimbalzato per tutta la Rete l'allarme, ripreso ed amplificato anche dalla stampa generalista, riguardo una gravissima vulnerabilità insita nella maggior parte dei DNS, il sistema che permette di "risolvere" i nomi degli host. Dopo molti mesi di lavoro segreto e con uno sforzo coordinato senza precedenti, tutti i principali produttori di software hanno contemporaneamente emesso le patch necessarie a rimuovere il problema dai propri prodotti. Infine, ad agosto, dopo un opportuno periodo di embargo, lo scopritore del problema *Dan Kaminsky*, nel corso della *Black Hat Convention 2008* (**Figura 1**) ha pubblicamente spiegato in cosa consiste. E la situazione non è affatto semplice... Come da molti sospettato si trattava di una nuova, ma micidiale, possibilità di inquinamento della cache (**cache poisoning**) dei DNS.

Inquinare la cache

L'inquinamento della cache consiste nel far sì che un *name server* accetti per buone, ed inserisca quindi nella sua cache locale per poterle riutilizzare in futuro, informazioni di risoluzione dei nomi che sono in realtà fasulle in quanto non provenienti da un *server autoritativo*. In questo modo un malintenzionato può indurre in errore tutti i client che si servono di quel *name server* per la risoluzione dei nomi. Ora, tutte le tecniche sinora note di cache poisoning avevano come effetto l'inserimento nella cache della vittima di una o più associazioni fasulle relative a singoli host, non ad interi domini. **Sfruttando invece la vulnerabilità di Kaminsky un malintenzionato potrebbe spacciare un proprio server DNS come autoritativo per qualsiasi zona** ed inserire questa informazione nella cache della vittima, col risultato di poter redirigere a piacimento verso la propria rete tutto il traffico originariamente diretto verso ogni host di ogni possibile dominio esistente. In pratica ciò significa semplicemente che **non si potrebbe più avere la certezza dell'identità di nessun host della Rete**, dato che teoricamente ogni *name server* potrebbe essere stato "taroccato" ad arte da qualcuno.

Figura 1 - Un momento della conferenza nella quale Dan Kaminsky ha rivelato i dettagli della vulnerabilità dei DNS da lui scoperta

Come e perché funziona l'attacco?

Tutto si basa sul modo in cui agisce il DNS (**Figura 2**). Un *name server* che riceve una query da un client (I), e non ha la risposta già disponibile localmente, non fa altro che rivolgere la stessa query ad un *root server* esterno (II); se questi ha la risposta pronta la fornisce direttamente, se invece non la ha fornisce un riferimento (*delegation*) ad un ulteriore name server che potrebbe conoscerla (III). Il *name server* di origine invia allora la propria query a questo nuovo *name server* delegato, ed il processo si ripete finché non viene raggiunto un name server autoritativo per la zona ricercata (IV), il quale può infine fornire la risposta definitiva alla query originale (V). A questo punto il name server di partenza non solo è in grado di fornire tale risposta al client che l'aveva originariamente richiesta, ma provvederà anche a memorizzarla nella propria cache locale per un tempo determinato (stabilito dall'amministratore della zona in questione) in modo da poter soddisfare direttamente ulteriori query che dovessero riguardare la medesima zona. Il problema generale dell'inquinamento della cache consiste nel fatto che è relativamente facile per un attaccante inviare ad un *name server* delle false risposte ad una sua query, costringendolo così a memorizzare nella propria cache informazioni errate. Ciò accade in quanto il protocollo di dialogo tra i name server non prevede alcuna forma esplicita di autenticazione dei corrispondenti o di validazione del contenuto, al di là di alcuni semplici accorgimenti finalizzati alla verifica ed alla prevenzione degli errori. Ma, quali sono i requisiti per cui una risposta viene presa per buona da un name server? Innanzitutto deve provenire dall'**indirizzo IP dell'host** cui è stata rivolta la query; in secondo luogo deve essere indirizzata alla specifica **porta UDP** da cui il richiedente ha inviato la query; ed infine deve portare con sé uno **speciale identificatore** (chiamato **QueryID** o **TXID**), impostato originariamente dal server che ha inviato la query, il quale contraddistingue una query effettivamente inoltrata ed ancora "aperta". Chiunque sia in grado di conoscere (o

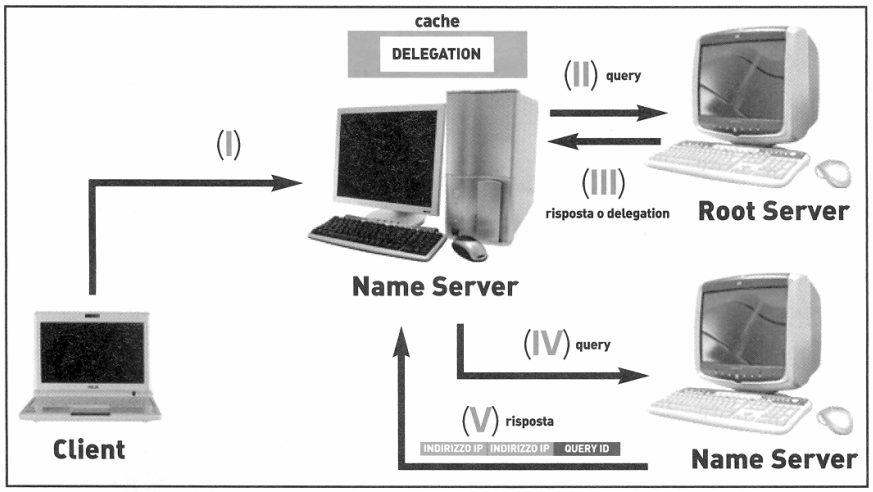

Figura 2 - Schema del processo di risoluzione dei domini Internet su cui si basa il funzionamento dei DNS

supporre, o indovinare) tali tre valori può dunque costruire ad arte un pacchetto di risposta fasullo ma formalmente valido, e sottoporlo al *name server* che ha fatto la query: se tale pacchetto arriva prima della reale risposta inviata dal *name server* cui era effettivamente stata inviata la query, esso verrà preso per buono dal server richiedente ed utilizzato per aggiornare la propria cache; il vero pacchetto di risposta, quando arriverà, verrà invece scartato in quanto corrispondente ad una query non più attiva.

Prevedere il futuro?

Riuscire a conoscere i valori giusti e creare un pacchetto di risposta accettabile non è cosa complicata. La porta sorgente del richiedente solitamente, ad esempio, è fissa e spesso corrisponde addirittura alla **well-known-port 53** del DNS. Per conoscere l'IP di provenienza basta fare in modo che il server oggetto di attacco indirizzi la sua query ad un server autoritativo noto, del quale si può utilizzare l'indirizzo inserendolo come mittente nei pacchetti fasulli. Il valore QueryID di una query, solitamente viene incrementato di 1 rispetto al precedente, il che rende molto facile poterlo indovinare. Inoltre, occorre tener presente che l'attaccante può bombardare la sua vittima con una raffica di risposte tutte leggermente diverse, nella speranza che almeno una di esse possa effettivamente avere i valori giusti ed essere quindi accettata per buona. Anche la generazione di un valore casuale (e non incrementale) del QueryID da parte di un *name server,* col tempo non si rivelato un metodo idoneo a contrastare l'individuazione del QueryID, in quanto si è scoperto che in moltissimi casi il generatore di numeri casuali utilizzato non era dotato di adeguate proprietà e bastava una semplice analisi statistica dei pacchetti per poter comunque prevedere con buona approssimazione i prossimi QueryID.

Entra in gioco Kaminsky

La vulnerabilità scoperta da Kaminsky, anziché inserire nella cache della vittima una falsa informazione relativa ad un singolo server, mira ad inserirvi una falsa delega: ossia l'informazione relativa ad un server autoritativo per un'intera zona, il che di fatto consente ad un malintenzionato di potersi "appropriare" di un intero dominio. L'attacco sfrutta il fatto che i *name server* mantengono nella propria cache anche le informazioni di delega: in altre parole un server, una volta avuta l'informazione (vera o falsa) che un certo DNS esterno è autoritativo per una certa zona, manterrà nella sua cache anche questa informazione ed in seguito interrogherà direttamente tale server tutte le volte che vorrà risolvere un nome appartenente al suo dominio, senza passare nuovamente per i *root server*. Il problema è ovviamente come inviare una falsa informazione di delega alla vittima facendogliela accettare come buona. Il trucco consiste nel costringere il *name server* vittima a generare query per le quali sicuramente non ha la risposta in cache, inondandolo poi di false risposte contenenti un'informazione di delega formalmente relativa alla zona richiesta, ma che punti tuttavia ad un name server opportunamente taroccato dal malintenzionato. Purtroppo, dato che l'attacco sfrutta una caratteristica intrinseca del protocollo DNS, non è realmente possibile chiudere la falla senza modificare il protocollo stesso: cosa ovviamente impossibile a farsi quantomeno in tempi brevi ed in maniera semplice. La migliore soluzione identificata, ed appunto implementata nelle le patch, è dunque in grado solo di mitigare la vulnerabilità ma non di risolverla definitivamente. L'obiettivo della correzione è essenzialmente quello di rendere più difficile la vita ai potenziali attaccanti, mettendoli in condizione di **non poter più predire i valori di QueryID e soprattutto di porta sorgente** che servirebbero loro per poter costruire pacchetti di risposta formalmente accettabili dal name server oggetto dell'attacco. Il concetto chiave è randomizzazione: così come fatto qualche anno fa per il campo QueryID, la modifica attualmente applicata ai server DNS fa sì che essi utilizzino per inviare le proprie query ogni volta una porta sorgente diversa, scelta completamente a caso nel più ampio range disponibile (e naturalmente controllando che anche il campo QueryID sia davvero casuale). Così facendo si amplia enormemente lo spazio delle possibili combinazioni dei due fattori, costringendo l'attaccante ad un compito molto più duro. Non è il massimo ma certamente è meglio di niente.

C'è andata bene...

La conclusioni tratte da Kaminsky al termine della sua conferenza sono: *"Per questa volta ci è andata bene, ma non è detto che in futuro saremo ancora così fortunati"*. Purtroppo introdurre come retrofit funzionalità di sicurezza nei protocolli vigenti è impresa ardua se non impossibile, mentre sostituirli con protocolli nuovi ed intrinsecamente sicuri (ad esempio DNSSEC nel caso del DNS) è quantomeno lento e difficoltoso. Tuttavia non abbiamo alternative, e dobbiamo fare in fretta se non vogliamo rischiare l'integrità dell'intera Rete. Praticamente tutti i software DNS del mondo si sono dimostrati vulnerabili all'attacco di *Kaminsky* tranne uno: **DJBDNS**, scritto dall'autore di **qmail** *Daniel J. Bernstein*. Si tratta di un server open source per Unix molto diffuso sviluppato proprio per essere intrinsecamente sicuro contro ogni forma di attacco noto e non. Ed infatti sin dall'inizio esso ha implementato la randomizzazione non solo dei valori di QueryID ma anche delle porte sorgenti utilizzate per inoltrare le query. Segno che una sana attitudine paranoica, ed un'attenta programmazione difensiva, possono risparmiare un sacco di guai futuri nei confronti di minacce ancora al di là da venire.

Corrado Giustozzi