Pesquisa sobre ataques DoS em DNS

Autor: Matheus Rotta Alves (184403)

Introdução:

[4] O DNS (Domain Name System) é um elemento fundamental da infraestrutura da internet. Até mesmo uma pequena parte da infraestrutura DNS estar indisponível por um pequeno período de tempo poderia potencialmente irritar toda a Internet e portanto é completamente inaceitável.

[1] 2 tipos de ataques DoS:

1. Ou o atacante cria um pacote de maneira precisa para explorar uma vulnerabilidade no software ou protocolo na vítima;
2. Ou então o atacante faz flooding para sobrecarregar um serviço crítico que um servidor provê. Um exemplo é o RDDoS (Reflection Distributed Denial of Service).

Vamos focar no tipo 2:

Funcionamento do ataque:

[1]O ataque ocorre da seguinte maneira: o atacante falsifica o campo de endereço fonte no datagrama IP para ser o de um host na rede da vítima. Usando o endereço falsificado, uma requisição DNS que pede um recurso válido é elaborada e mandada a um “name server” intermediário. Esse último normalmente é um servidor DNS recursivo, que executa o procedimento de resolução do nome e encaminha a resposta final à máquina alvo.
O atacante vai mandar repetidamente a requisição ao “name server” intermediário mas com todas as respostas indo para a rede da vítima. Potencialmente, o adversário poderia consumir toda a largura de banda de uma linha T1* gerando alguns milhares de respostas.

*obs: uma linha T1 é uma conexão de transmissão dedicada entre um provedor de serviço e um cliente. Normalmente toda de fibra ótica e conseguindo carregar 60 vezes mais dados do que um modem residencial normal.

- **Amplificação**: usar escolhas (opções) da DNS query que fazem com que a resposta, que é um pacote UDP, seja muito maior do que o normal (muito maior do que as queries, por exemplo).

[1] *Fator de Amplificação*: (size of response)/(size of request);

Como amplificar?

[1] O atacante cria a requisição de modo que ela inclua um tipo específico de recurso DNS tal que o servidor DNS autoritativo gere respostas grandes. Isso pode ser feito de 2 maneiras:

1. Descobrir quais servidores DNS guardam RR's (Resource Records) que quando requisitadas criam respostas grandes.
2. Ou comprometendo um servidor DNS e deliberadamente incluir uma entrada - também conhecida como "entrada de amplificação" - que vai criar uma resposta grande. Um exemplo dessa técnica, explorando entradas .txt grandes, é introduzida no EDNS (é uma RFC). Depois disso, o atacante elabora uma lista de "name servers" recursivos abertos que irão, recursivamente, requerer e retornar a "entrada de amplificação" que o atacante criou.

Como mitigar esse tipo de ataque?

[1] O DNS usa UDP para transportar requisições e respostas. Deste modo, o usuário malicioso consegue fabricar as requisições DNS falsificadas facilmente. Assim, uma primeira medida de proteção seria introduzir detecção/prevenção de "spoofing" como aqueles propostos na RFC 4033 (DNSSEC). Outra medida interessante seria desabilitar recursão em "name servers" de fontes desconhecidas, pois isso reduziria o fator de amplificação.

Limitações das medidas de mitigação

[1] O principal problema relacionado às medidas previamente mencionadas é que poucos servidores DNS realmente as implementam. Decorre disso que os servidores desprotegidos ainda estão bastante propícios a ataques de amplificação. Adicionalmente, esses mecanismos de proteção não proporcionam segurança contra "insiders" maliciosos, que sabidamente são responsáveis por grande parcela dos incidentes de segurança.

Diferentes métodos por diferentes autores:

--[1] DAAD (DNS Amplification Attack Detector) (php) : vai perceber se uma response está associada a uma request que ele fez baseando-se em um "one-to-one mapping". Se não está associada a nenhuma request numa certa janela de tempo, adicione uma regra no firewall para bloquear o IP do provável atacante.
[3] comenta que o principal problema dessa implementação é o rápido aumento do tamanho do banco de dados nos casos em que há alta taxa de tráfego.

---[2] Stale Cache: Os resolvedores de nome não eliminam completamente entradas na cache cujos valores “time to live” (TTL) tenham expirado. Ao invés disso, tais entradas são expulsas da cache e guardadas em uma “stale cache” separada. No momento da resolução de queries, primeiro procura-se na cache, depois nos “name servers” apropriados e, caso estejam indisponíveis (por causa de um ataque), usa-se o valor da “stale cache”, caso essa tenha guardado a RR em questão.

---[3] Usar uma rede neural para classificar se o tráfico é malicioso ou não. O input da rede neural foi definido como três características do tráfico capturado no “name server” (numa janela de tempo de 20s):

1. Throughput das requisições DNS recebidas (número de bits recebidos pelo servidor;
2. Tamanho médio dos pacotes durante a janela de tempo;
3. Perda de pacotes, definido como o número de pacotes perdidos que não chegaram à sua destinação devido ao flooding.

O output da rede eram três unidades, tal que [0 0 0] indicava condições normais, [0 0 1] indicava um ataque DoS direto (sem amplificação), e [0 1 0] indicava um ataque de amplificação. O melhor tipo de rede classificadora foi a que usava BackPropagation, com uma acurácia de 99% e uma taxa de falsos positivos de 0.28%.

---[4] DNS-Guard: usar cookies para autenticar comunicação. É necessário incorporar (embed) cookies em mensagens DNS para tal fim. Para entender o DNS-Guard, vamos primeiro reportar aqui o “overview” que o próprio paper propõe da infraestrutura DNS:

Overview: a infraestrutura DNS é composta de três elementos: o “stub resolver”, que fica no host final, o LRS (Local Recursive Server) e o ANS (Authoritative Name Server). Quando uma aplicação pede a resolução de um nome, o stub resolver manda uma chamada recursiva para o servidor recursivo local. O LRS então faz chamadas iterativas ao ANS’s.

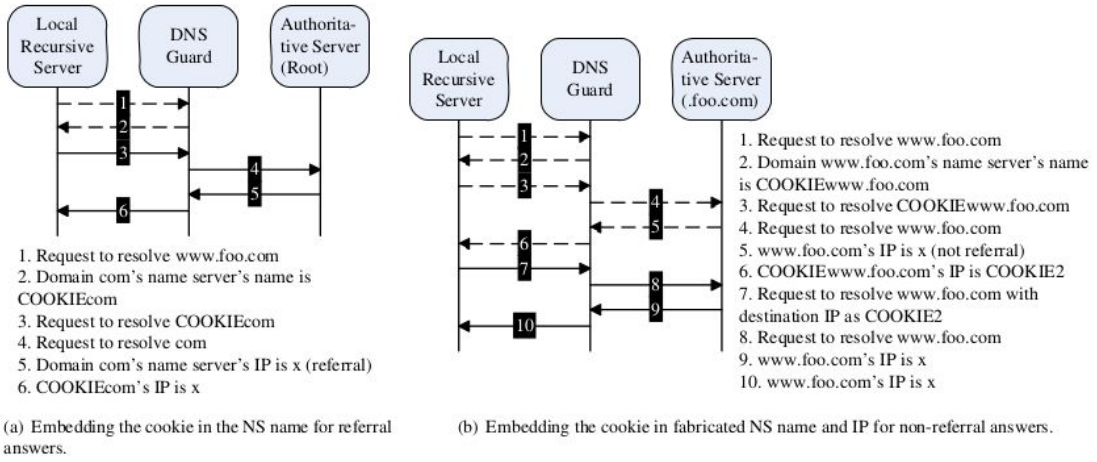
Dito isso, o DNS-Guard foi desenvolvido com o foco em detectar spoofing para proteger os ANS’s sem precisar modificar os LRS’s.

Como já foi dito, a solução encontrada foi incorporar as cookies em mensagens DNS. De maneira sucinta, DNS-Guard fica entre o LRS e o ANS, e age de duas maneiras distintas dependendo da resposta que ANS precisa retornar ao LRS:

- **Referral Answer:** neste caso, o cookie é incorporado ao “NS Name”, e o LRS terá que responder dizendo que precisa resolver o nome

Cookie(original_request). Quando DNS-Guard vê isso, sabe que a request é válida e a repassa ao ANS apenas com o “original request”. O processo continua assim até que se conclua a comunicação, com o DNS-Guard atuando como intermediário de maneira transparente para o ANS e o LRS.

- **Non-Referall Answer:** nesse caso a solução é parecida mas a cookie é incorporada tanto no NS Name quanto no IP;



(a) Embedding the cookie in the NS name for referral answers.

(b) Embedding the cookie in fabricated NS name and IP for non-referral answers.

Figure 2. Using NS name and IP to embed cookie in traditional DNS. Messages presented as dashed lines will be skipped after first access.

Imagem Original do paper [4] representando a atuação do DNS-Guard e como ele incorpora as cookies às mensagens DNS.

Bibliografia:

[1] A Fair Solution to DNS Amplification Attacks

- Georgios Kambourakis, Tassos Moschos, Dimitris Geneiatakis and Stefanos Gritzalis

[2] Mitigating DNS DoS Attacks

- Hitesh Ballani, Paul Francis.

[3] Detection of Denial of Service Attacks against Domain Name System Using Neural Networks

- Samaneh Rastegari, M. Iqbal Saripan and Mohd Fadlee A. Rasid

[4] Spoof Detection for Preventing DoS Attacks against DNS Servers

- Fanglu Guo Jiawu Chen Tzi-cker Chiueh