SR2I 203 - État de l'art, études initiaux

Matheus Augusto da Silva - Alessandro Montaldo

December 16, 2018

1 Introduction

Our project consists in the study and the implementation of a specific type of Denial of Service attack, a Low and Slow type attack, the slowloris attack.
The objective of this report is to introduce the studies and progresses done and explain and motivate which will be our next steps. In particular the have followed path:

- a study focused on the comprehension of the main functionallity of the HTTP protocol and Apache web server.
- a research focused on the slowloris attack.
- a realisation of a first implementation in Python using the software Scapy of the basic functionallity of the attack.

2 Étude du Protocole HTTP

Hypertext Transfer Protocol a été conçu avec le but de construire une architecture client-serveur tel qu'on peut implémenter un déploiement du WWW. C'est un protocole de la couche application et il permet l'implémentation d'un des outils les plus populaires d'Internet aujourd'hui: le navigateur Web.
Le principe c'est d'une architecture où un site web (serveur ou ensemble de serveurs) est associé à plusieurs ressources lesquelles un client peut demander via les méthodes HTTP. Un serveur web peut être identifié par son URL ou son adresse IP et généralement garde informations comme pages web, images et fichiers.

2.1 HTTP Requests and responses

Each request begins with a Request-Line that indicates specific method, the resource to which the method applies, and the version of http that the client can support. One or more message headers can follow and a message body separated with a blank line.

HTTP responses look a lot like HTTP requests. The only significant difference is that responses begin with a status line rather than a Request-Line.

```
GET / HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
    (compatible; MSIE 5.5; Windows NT 5.0)
Host: www.ft.com
Connection: Keep-Alive
```

Figure 1: GET request example

3 Étude du Serveur Web Apache

Apache est le serveur Web le plus populaire sur Internet. Plus que la majorité des sites Web sont basés sur elle. Ce qui rend Apache vulnérable à l'attaque qui sera analysée, c'est la manière dont il traite ses clients. En particulier, après avoir reçu une requête HTTP, un nouveau fil associé à cette conversation s'ouvrira.
Les serveurs Apache plus anciens utilisent un modèle de limitation de thread, généralement un serveur Apache normal peut ouvrir environ 200 threads en parallèle. Normalement, la limitation n'est pas un problème pour les serveurs de site de petite taille, dans la mesure où ils n'ont pas besoin de gérer leur trafic.

3.1 Nginx vs Apache

Nginx a été créé pour résoudre le problème de la gestion d'un grand nombre de connexions. Enfait, au lieu d'une architecture à base de threads, comme Apache, Nginx a une architecture pilotée par les événements qui ne crée pas de nouveau processus pour chaque requête. Dans notre projet, nous testerons l'attaque de slowloris sur différents serveurs Web et verrons si l'expérimentation

4 Étude du attaque Slowloris

L'idée de base de l'attaque de slowloris est d'ouvrir de nombreuses connexions en envoyant des requêtes HTTP partielles. Des fractions de la demande sont envoyées ultérieurement à intervalles réguliers pour empêcher les sockets de se fermer.
On peut diviser un exploit suivant l'algorithme suivant:

- On crée une requête GET destinée au serveur cible
- On envoie la requête byte par byte très lentement, dans l'ordre de secondes de delai
- On répète la même procédure avec un numero tres grand de messages, entre 100 et 200, dependant des configurations du serveur.
- Chaque GET répondu sera interprète dans le serveur comme un nouveau client; ça ouvre un nouveau thread dans le réservoir limité du serveur.
- Avec le temps, les requêtes lentes mais continues de la machine attaquante iront remplacer tous les connexions "légitimes".

4.1 Slowloris detectability

A desireble property of an attack is its detectability. After the slowloris attack starts any attached on the log file won't be done until the requests are completed. Once the request is completely sent or once the session gets shut down there will be a log trace of the connection, this means that several hundred 400 errors in the

web server logs will be presents, although it may be possible to turn them into 200 OK messages instead by completing a valid request.

5 Implémentation en Python

6 Prochain travail à faire

References

[1] Internet Engineering Task Force (IETF)
Request for Comment 1945 - Hypertext Transfer Protocol 1.0

[2] THOMAS, Stephen
HTTP Essentials: Protocols for Secure, Scaleable Web Sites

[3] The Apache Group
Vue d'ensemble des nouvelles fonctionnalités d'Apache 2.0

[4] Christian Folini, LWN.net group, archivé
Apache attacked by a slowloris

[5] Ha.ckers group, archivé
Slowloris HTTP DoS

[6] Repositoire Github
slowloris.pl, a perl implementation