

# Relatório Frameworks

João Fiuza de Alencastro 15/0131933

**Abstract**—Relatório destinado à matéria de Segurança de Redes do Departamento de engenharia Elétrica da Universidade de Brasília. Experimento realizado enfatizando a utilização de frameworks e aplicações prontas.

**Index Terms**—Segurança, redes, Framework, NMap, ports, TCP, UDP, vulnerabilidade.

## I. INTRODUCTION

**R**EALIZAR experimentos utilizando ferramentas específicas, melhor desenvolvidas e de fácil acesso. Frameworks oferecem a vantagem de juntar mais de uma ferramenta de descoberta de vulnerabilidades e de testes de penetração em uma só interface. Testes de penetração são realizados de forma correta quando seguem uma sequência de ações, construindo uma base de conhecimento no início e evoluindo à medida que são realizados testes, por este motivo aplicações específicas permitem ataques mais bem estruturados.

Serão utilizadas algumas das ferramentas já disponíveis na distribuição do Linux Kali. Essas ferramentas, são aplicações vastamente utilizadas e são um alicerce no arsenal de ataques de 'hackers' ou 'ethical hackers'. Dentre elas estão: nmap, nikto, hydra, cisco-torch e por fim, a mais completa das aplicações, o Sparta, um framework que contempla todas as ferramentas previamente citadas.

### A. Aplicativos

1) *nmap*: O conhecido aplicativo nmap deve ser realizado em etapas ou fases para ser utilizado de forma correta. Isso já é um forte indicativo que a ferramenta em questão vai muito além de um simples 'port scanner'.

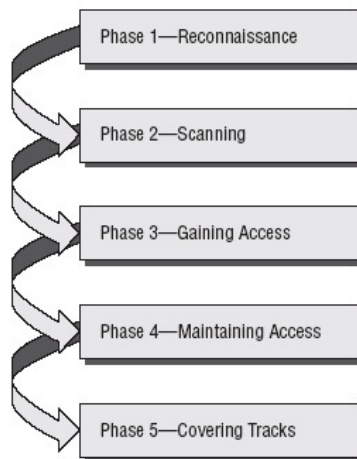


Fig. 1. Fases de um nmap scan

2) *nikto*: Baseando-se no resultado de um rápido scan no apache rodando no localhost, é visto que a ferramenta roda vários scripts de verificação de segurança do ambiente web. Apesar de o scan ser feito em um simples html no servidor local, o resultado é muito interessante, por mostrar o resultado de todas as verificações. A figura abaixo mostra o resultado obtido.

```

root@kali:~# nikto -host localhost
- Nikto v2.1.6
-----
+ Target IP: 127.0.0.1
+ Target Hostname: localhost
+ Target Port: 80
+ Start Time: 2018-10-16 09:57:12 (GMT-4)
+ Server: Apache/2.4.34 (Debian)
+ Server leaks inodes via ETags, header found with file /, fields: 0x29cd 0x572475c47b100
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: GET, POST, OPTIONS, HEAD
+ OSVDB-561: /server-status: This reveals Apache information. Comment out appropriate line in the Apache conf file or restrict access to allowed sources.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /server-status: Apache server-status interface found (pass protected)
+ 7372 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time: 2018-10-16 09:57:24 (GMT-4) (12 seconds)
+ 1 host(s) tested
  
```

Fig. 2. Resultados nikto

3) *hydra*: Com um resultado bem sucedido de um brute force, foi possível "quebrar" a senha de um banco de dados PostgreSQL - bastante utilizado em diversas aplicações - utilizando a ferramenta Hydra. No exemplo, foi passado como parâmetro só um login, "postgres", uma lista de possíveis senhas e um host vítima do ataque. A própria ferramenta já sabe a porta padrão utilizada pelo PostgreSQL, e caso a porta não fosse a padrão, a fase de pesquisa e information gathering é útil para este tipo de situação.

```

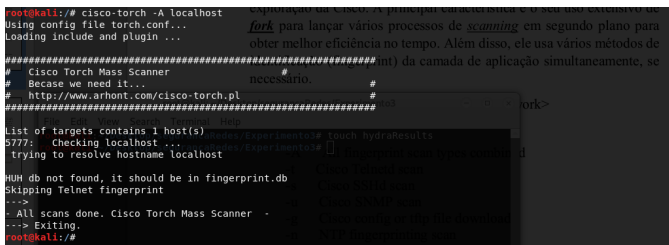
root@kali:~# hydra -l postgres -P /usr/share/wordlists/nmap.lst localhost postgres
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
Hydra (http://www.thc.org/thc-hydra) starting at 2018-10-16 10:10:21
[DATA] max 16 tasks per 1 server, overall 16 tasks, 5085 login tries (1:1/p:5085), ~310 tries per task
[5432][postgres] host: localhost login: postgres password: postgres
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-10-16 10:10:22
  
```

Fig. 3. Resultados hydra

4) *cisco-torch*: Este aplicativo também é um analisador de vulnerabilidades, porém ele difere em seus métodos de análise, além de que ele distribui o processamento computacional, agilizando os procedimentos. Além disso, ele utiliza vários métodos de *fingerprinting* em camada de aplicação simultaneamente. Abaixo é mostrado resultados obtidos no próprio hospedeiro local.

### B. Frameworks

#### 1) *Sparta*:



```

kali@kali:~$ cisco-torch -A localhost
Using config file torch.conf...
Loading include and plugin ...
=====
# Cisco Torch Mass Scanner
# Because we need it...
# https://www.arhont.com/cisco-torch.pl
=====
# List of targets contains 1 host(s)
# 2777: Checking localhost ...
# trying to resolve hostname localhost
#
# Httb db not found, it should be in fingerprint.db
# Skipping Telnet fingerprint
#
# All scans done. Cisco Torch Mass Scanner
# --> Exiting.
kali@kali:~$

```

Fig. 4. Resultados cisco-torch

## II. CONCLUSION

A essa altura, é sabido que testes de penetração são feitos em fases, assim como a descoberta de vulnerabilidades da ferramenta nmap. Por esse motivo a utilização de aplicações e frameworks é tão útil, já que a utilização das mesmas permitem uma análise cautelosa e segmentada para um relatório bem feito. Por exemplo, se é descoberto que a aplicação vítima de ataques é uma aplicação conhecida com certas vulnerabilidades, poderão ser exploradas facilmente pontos fracos específicos daquela versão daquela certa aplicação. Porém, as ferramentas apresentadas neste experimento são de cunho generalizado, permitindo um estudo mais básico que dará espaço para um estudo profundo posteriormente.

## REFERENCES

- [1] <https://tools.kali.org/information-gathering/cisco-torch>
- [2] <https://nmap.org/book/nmap-phases.html>
- [3] <http://amaliciousmind.blogspot.com/2013/08/nmap-step-by-step.html>