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

REALIZAR 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'*.

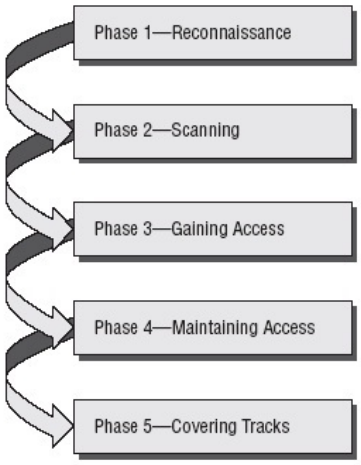

Fig. 1. Fases de um nmap scan

2) nikto: Baseando-se no resultado de um rápido scan no apache rodando no localhost, é visto que a ferramenta roda vários scripts de verificacão de seguranca 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 verificacões. A figura abaixo mostra o resultado obtido.

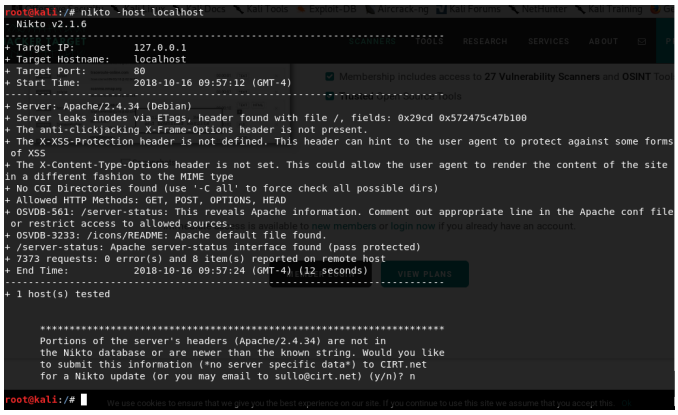

Fig. 2. Resultados nikto

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.

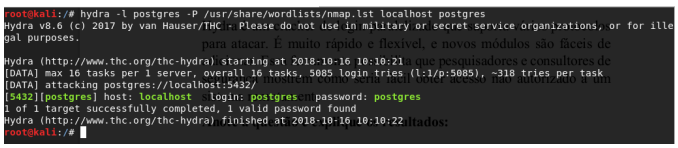

Fig. 3. Resultados hydra

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.

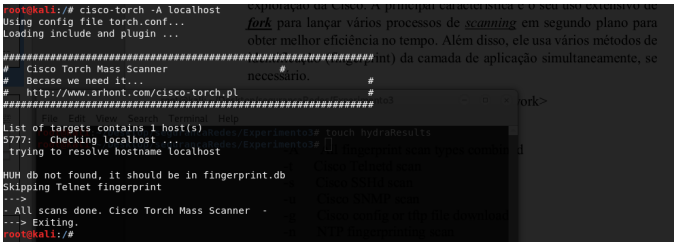

Fig. 4. Resultados cisco-torch

B. *Frameworks*

1) Sparta: Os frameworks podem ser considerados a parte mais importante de um teste de penetração. Pois, eles carregam dentro de sí uma grande diversidade de ferramentas pequenas, porém extremamente úteis. Como, por exemplo, o nikto, ou o hydra. São todas ferramentas que se completam e formam um ótimo alicerce de informações sobre a vítima.

Antes de começar a utilizar o Sparta, foram iniciados alguns serviços comuns na máquina local, abrindo certas portas, como mostra a figura abaixo.

```
root@kali:~# service postgresql start
root@kali:~# service vsftpd start
root@kali:~# service apache2 start
root@kali:~# nmap localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-04 14:06 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000050s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
5432/tcp open  postgresql

Nmap done: 1 IP address (1 host up) scanned in 0.50 seconds
root@kali:~#
```

Fig. 5. Serviços iniciados localmente

Uma vez que os serviços estão expostos na rede, e há uma conexão estabelecida nesta mesma rede, o próprio framework se encarrega de fazer um *port scan* e retorna para o usuário as portas abertas. Além disso, pode-se observar na aba de logs os scripts e tarefas que rodam no framework, como mostra a figura abaixo.

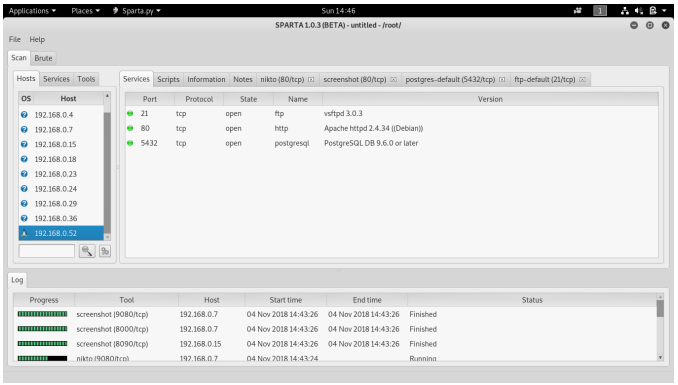

Fig. 6. Resultados nmap sparta

Depois de descobrir uma vulnerabilidade que pode ser explorada (neste caso será o postgresql), chega a hora de realizar ataques. Uma possibilidade que o Sparta oferece é o brute force da ferramenta Hydra, explicada anteriormente. Então, no próprio framework, utiliza-se o brute, apontando um nome de usuário e uma lista de possíveis senhas, como mostra a figura abaixo.

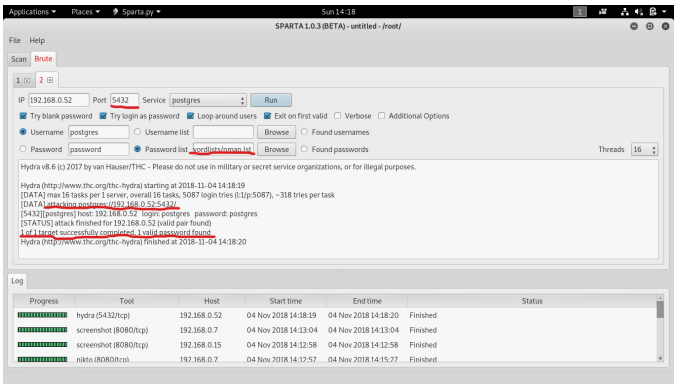

Fig. 7. Resultado do brute force na porta 5432

Foi apresentado para o usuário um ataque de força bruta bem sucedido.

Agora, analisando os pacotes trafegados durante todo o experimento, pode-se observar um grande número de tentativas do framework tentando autenticar o usuário na aplicação do postgresql. A figura 8 mostra uma pequena parte do tráfego que corre ”por trás dos panos”. E a figura 9 mostra como é um pacote da aplicação.

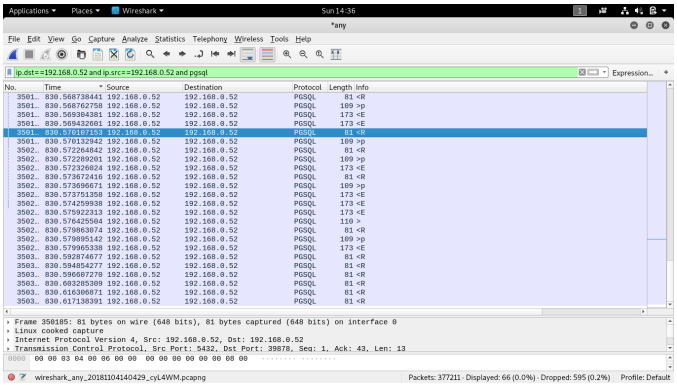

Fig. 8. Pacotes da aplicação PSQL recebidos

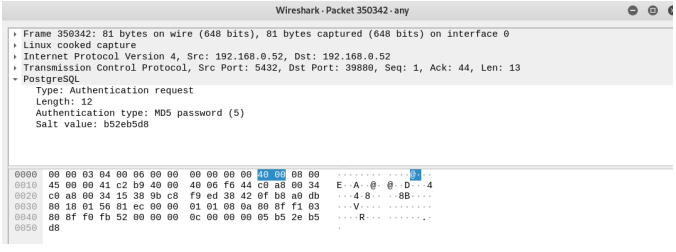

Fig. 9. Pacote a nível de aplicação

Para uma análise mais profunda e cautelosa, pode-se analisar pacote por pacote, e não somente a nível de aplicação.

Segundo [4] , é possível ver o formato das mensagens trocadas por este protocolo que tem sua documentação aberta para qualquer um ler e se informar melhor, mostrando quando o pacote recebido traz uma mensagem bem sucedida ou falha.

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 apliação. Porém, as ferramentas apresentadas neste experimento são de cunho generalizado, permitindo um estudo mais básico que dará espaço para um estudo profundo posteriormente.

Foi mostrado neste experimento como um ataque pode ser custoso para o atancante também, como, o brute force, por exemplo, que precisa enviar pacotes a nível de aplicação constantemente a fim de acertar uma autenticação de um usuário. Além disso, é concluído que um ataque como o brute force pode ser facilmente evitado pela vítima se a mesma criou uma senha forte para tal aplicação, aumentando exponencialmente a complexidade do sucesso da força bruta.

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
[4] https://www.postgresql.org/docs/9.3/static/protocol-message-formats.html