

Roteiro 2

Segurança em Sistemas Operacionais

Objetivo: Desenvolver as habilidades para a condução de um teste de vulnerabilidade de sistema adotando uma metodologia. Ainda são objetivos deste roteiro retomar conceitos fundamentais de administração de redes e sistemas operacionais.

Preâmbulo

Os desafios deste roteiro serão em vários momentos conduzidos por meio de imagens de máquinas virtuais que deverão ser executadas com o player gratuito VirtualBox (<https://www.virtualbox.org/>). As ferramentas e scripts para a execução dos testes poderão ser instaladas em seu sistema operacional pessoal ou até mesmo serem executadas por meio de outra máquina virtual que deverá ser executada com a distribuição Kali Linux (<https://www.kali.org/downloads/>).

Este roteiro é organizado em atividades individuais. Fiquem atentos com as datas dos entregáveis e bons estudos.

Disclaimer

A disciplina de Tecnologias Hackers proporciona aos alunos a experiência de testar e explorar ambientes computacionais por meio de ferramentas e scripts reais. O objetivo único é de capacitar os alunos para as práticas de testes e análises de segurança de redes, sistemas e aplicações por meio de simulações de exploração em ambiente educacional. A utilização destas técnicas não deverão ser realizadas em outros ambientes sem o consentimento do proprietário ou administrador da rede, sistema ou aplicação.

Getting Started

Ainda o termo Hacker é interpretado de diferentes formas em nossa sociedade. Na informática Hacker simboliza a pessoa que se dedica profundamente a analisar, questionar, modificar e testar os limites de arquitetura e segurança de dispositivos e softwares. É recorrente o termo Hacker estar associado à pessoa de atitude maliciosa, cuja a sua principal motivação é aplicar seus conhecimentos técnicos e de análise para atividades para benefício próprio, ou com a simples intenção de impactar um ambiente computacional por outros motivos. Este especialista é conhecido como Cracker. Para finalizar, Hacker é a pessoa com habilidades que além de burlar e comprometer sistemas, contribui para o desenvolvimento e evolução tecnológica, seja pela motivação em ampliar e compartilhar seus conhecimentos em segurança, ou simplesmente pelo desenvolvimento e melhoramento de softwares e sistemas informatizados.

Pentest é um processo de análise detalhada do nível de segurança de um sistema ou rede usando a perspectiva de um infrator, ou seja, deve ser tratado como o mais próximo possível de um ataque real. Se tratado desta forma, é possível ter o conhecimento total do que poderia acontecer caso um ataque realmente existisse, garantindo assim a possibilidade de uma estratégia de prevenção.

“envolvem a simulação de ataques reais para avaliar os riscos associados a potenciais brechas de segurança.” (Georgia Weidman)

A utilização de uma metodologia permite dividir um processo complexo em uma série de tarefas menores e mais administráveis. Possibilita conhecer o alvo:

- Onde ele está localizado?

- Qual o endereço IP?

- Que sistema operacional o alvo está executando?

- Quais serviços estão sendo executados?

- Quais versões de softwares estão sendo executados?

Dependendo da literatura a metodologia conterá entre quatro e sete passos. Entretanto, dependendo do autor ainda podem existir mais passos. No mercado são referências de metodologias e guias de boas práticas para a realização do PenTest **NIST e o OWASP**. Contudo não existe uma metodologia padrão para a realização do PenTest.

A seguir segue uma breve descrição das **metodologias e boas práticas adotadas** em testes de exploração de vulnerabilidade:

NIST SP 800-115 (National Institute of Standards and Technology)

Tem Como objetivo **orientar no planejamento tanto na aplicação quanto na análise dos testes**. Esta metodologia especifica **como as diferentes técnicas devem ser utilizadas para que os testes sejam efetuados com precisão**. É considerada um dos melhores documentos e é a mais adotada por profissionais e consultorias de segurança.

Essa metodologia é dividida em algumas etapas:

1. **Testes de segurança e visão geral dos exames**: focada em 3 métodos teste, exame e entrevista.
2. **Revisão das técnicas**: essa parte discute as técnicas utilizadas para descobrir as vulnerabilidades utilizando exames passivos.
3. **Identificação e técnicas de análise dos alvos**: essa parte tem como objetivo identificar serviços em atividades (e suas portas utilizadas) para verificar possíveis vulnerabilidades.

4. **Técnicas de validação das vulnerabilidades:** essa parte utiliza os dados obtidos na sessão anterior assim explorando a existência de possíveis vulnerabilidades
5. **Planejamento de avaliações de segurança:** essa parte aborda a melhor orientação para que possa ser criado as políticas de testes.
6. **Execução de avaliação de segurança:** nesse ponto são destacados pontos chaves na fase de execução onde são fornecidas recomendações referente a avaliação.
7. **Atividades pós testes:** nessa parte são fornecidas para a organização, maneiras de transformar as descobertas em formas de segurança assim fornecendo ações contra as vulnerabilidades encontradas.

Essa metodologia não é tão detalhada do ponto de vista técnico como as demais, mas fornece informações suficientes para a realização de um teste de penetração.

ISSAF (Information Systems Security Assessment Framework)

Essa metodologia é disponibilizada pelo OISSG (Open Information Systems Security Group), é a mais volumosa metodologia disponível. Basicamente consiste em três fases de estratégia:

1. **Planejamento e preparação:** fase onde são trocadas informações iniciais para planejamento e preparação dos testes para avaliação do sistema.
2. **Avaliação:** fase em que o teste coleta informações, mapeia a rede, identifica as vulnerabilidades no sistema, ou seja, o Pentest analisa todo o sistema que está sendo avaliado e corrige os problemas detectados.
3. **Relatórios e limpeza:** nesta fase, é apresentado o relatório de todos os testes executados, mas se algum erro ou vulnerabilidade forem encontrados durante os testes, devem ser avisados antes do término da avaliação do sistema e geração dos relatórios.

Sua abrangência cobre quatro áreas:

- A) Segurança de Rede
- B) Segurança de Host
- C) Segurança de Aplicação
- D) Segurança de Banco de Dados

OSSTMM (Open Source Security Testing Methodology Manual)

O OSSTMM (Open Source Security Testing Methodology Manual) é uma metodologia disponibilizada pela ISECOM (Institute for Security and Open Methodologies). Suas definições são constituídas a partir do escopo, que representa todo o ambiente de segurança operacional possível para qualquer interação com qualquer ativo. O principal objetivo dessa metodologia é **caracterizar a segurança operacional através dos exames e correlação dos resultados dos testes de uma maneira consistente.**

Canais de interação OSSTMM:

1. Humano
2. Físico
3. Wireless
4. Telecomunicações
5. Rede de dados

OWASP (Open Web Application Security Project)

Essa metodologia tem um **foco maior em testes de aplicações web.**

O OWASP segue alguns princípios para a execução dos testes: **não acreditar em milagres, pensar estrategicamente, testar cedo e com regularidade, entender o escopo da segurança, desenvolver a mentalidade correta, estender o objetivo, usar as ferramentas corretas, se atentar aos detalhes e documentar os resultados.**

O OWASP ainda disponibiliza diversos materiais para o desenvolvimento seguro de aplicações e orientações para mitigar vulnerabilidades.

O documento é organizado em 12 subcategorias para teste de penetração:

1. Introdução e objetivos;
2. Coleta de Informações;
3. Teste de gerenciamento de configurações;
4. Teste de gerenciamento de identidade;
5. Teste de autenticação;
6. Teste de autorização;
7. Teste de gerenciamento de sessão;
8. Teste de validação de entrada;
9. Manipulação de erros;
10. Criptografia;
11. Teste de lógica de negócios;
12. Teste do lado do cliente.

Estes testes podem ser realizados em diferentes maneiras:

Tem pleno conhecimento (caixa branca) - Onde o hacker conhece bem as características do ambiente (sistemas, equipamentos, protocolos...)

Tem conhecimento parcial (caixa cinza) - Onde se possui informações de parte do

ambiente a ser explorado.

Não tem conhecimento da meta a ser avaliada (caixa preta) - Onde não é fornecido nenhuma informação do ambiente a ser explorado.

Roteiro 2.1 – Reconhecimento do alvo

Objetivo: utilizar as principais técnicas para o reconhecimento de um alvo nas camadas de rede e sistema operacional em busca de vulnerabilidades expostas e conhecidas.

Máquina virtual para análise: Metasploitable2

Download: <https://sourceforge.net/projects/metasploitable/files/Metasploitable2/metasploitable-linux-2.0.0.zip/download>

Descrição:

Nesta fase deve-se colher o máximo de informações sobre o alvo. Crie ou utilize uma metodologia para o registros destas coletas. Sempre tenha em mãos um bom e valioso diário de bordo e anote tudo o que for percebido em relação ao alvo, como por exemplo testes realizados, respostas de comandos executados, entre outros resultados explorados (Existe um exemplo que poderá ser utilizado no blackboard).

Bibliografia de referência:

Capítulo 3 – Aprendendo Pentest com Python – Christopher Duffy

Exercício 1.1.a: Descubra qual ip do seu alvo. Depois de importar a máquina virtual para o seu sistema descubra o endereço que este host recebeu em sua rede. Você já pode utilizar neste momento outra instância de máquina virtual com o Kali Linux e a partir dele utilizar as ferramentas e scripts que permitiram você executar os

demais exercícios deste roteiro. Registre em seu diário de bordo, qual a técnica utilizada para resolver este exercício (print de tela com o comando, ferramenta ou script utilizado).

Exercício 1.1.b: reconhecendo serviços e portas abertas do alvo.

SEM utilizar uma ferramenta de escaneamento de portas e serviços descubra qual o nome e versão do processo que está executando na porta 21 do alvo. Evidencie o comando e sua saída no diário de bordo.

Exercício 1.1.c:

Fingerprint é o nome dado a fase dentro de um pentest com o objetivo de identificar o sistema operacional do host alvo. Neste exercício você deverá descobrir o maior número de informações sobre o Sistema Operacional do host alvo como versão, distribuição e arquitetura. Registre no diário de bordo os comandos, ferramentas e scripts utilizados. Obrigatoriamente esta informações devem ser acessadas a partir do host do atacante (Kali).

Exercício 1.1.d : Criação de Escaneamento de Portas com Python.

Descrição: Desenvolvimento de uma aplicação que realize o escaneamento de portas de comunicação de um destino por meio de bibliotecas de desenvolvimento da Linguagem de programação Python.

Você deverá realizar uma pesquisa dos módulos e bibliotecas que permitem o desenvolvimento de uma ferramenta para o escaneamento de portas TCP de acordo com as premissas a seguir:

- Ser em linguagem Python;
- Deverá possuir uma interface amigável e de fácil utilização (user-friendly interface);

- Permitir o escaneamento de um host ou uma rede;
- Permitir inserir o range (intervalo) de portas a serem escaneadas;
- Além da função de escaneamento, espera-se que seu código relacione as portas Well-Know Ports e seus serviços, e apresente em sua saída (imprimir) o número da porta e o nome do serviço associado.
- Existem diversos projetos e documentações relacionados com esta atividade. Aproveite para analisar os códigos já desenvolvidos para teu projeto.

Introdução

As ferramentas de escaneamento permitem a descoberta de vulnerabilidades em ambientes computacionais, entre outras funcionalidades. Os escaneadores estão disponíveis como ferramentas especializadas projetadas apenas para “escanear” vulnerabilidades em um host, como por exemplo determinar se suas portas de comunicação estão sendo ou não usadas. São extremamente úteis no processo de descoberta e reconhecimento do alvo em um PENTEST, bem como, para a administração de ambientes computacionais. Muitas portas estão associadas a serviços específicos de rede. Para isso, é fundamental o conhecimento sobre sockets e dos protocolos de transporte, bem como, suas características como cabeçalho e *flags*.

Existem basicamente três tipos de escaneamento:

- **Escaneamento de porta (*port scanner*):** Seu objetivo é verificar portas abertas e serviços disponíveis em um host.
- **Escaneamento de rede:** Permite identificar os hosts que estão ativos em uma rede.
- **Escaneamento de vulnerabilidades:** Busca por vulnerabilidades conhecidas em um host.

Neste roteiro vamos trabalhar com o *port scanner*.

Port scanner

É a técnica mais popular e usada por Hackers/Crackers para descobrir serviços vulneráveis em um sistema e o NMAP a mais popular das ferramentas.

NMAP

Pode ser considerada uma das ferramentas mais completas para realizar varredura em redes, pois disponibiliza um grande número de opções, possibilitando realizarmos diversas varreduras em busca de vulnerabilidades e características do alvo. Essa ferramenta possui, inclusive, opções que permitem burlar sistemas de proteção, como IDS/IPS e Firewall, cujas regras poderiam bloquear ou detectar varreduras não permitidas.

Ela localiza e identifica todas as portas TCP e UDP disponíveis em um host, tentando determinar qual o serviço que está “escutando” em cada porta e é capaz de identificar o tipo de sistema operacional em execução. O nmap é visto como uma ferramenta de segurança, usada para descobrir “brechas” em sistemas, ajudando na tarefa de monitoração e gerenciamento da rede e identificação de serviços rodando em servidores.

Sintaxe:

```
nmap [Scan Type(s)] [Options] {target specification}
```

```
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@exin-eth:/# nmap 192.168.0.1

Starting Nmap 6.00 ( http://nmap.org ) at 2015-09-30 10:20 BRT
Nmap scan report for 192.168.0.1
Host is up (0.0020s latency).
Not shown: 990 filtered ports
PORT      STATE SERVICE
23/tcp    closed telnet
80/tcp    open  http
1900/tcp  open  upnp
8080/tcp  open  http-proxy
MAC Address: CC:0D:EC:ED:3F:9D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 4.61 seconds
root@exin-eth:/#
```

No exemplo da Figura, ele foi executado de forma simples apenas indicando o IP do alvo. Como resposta é exibido as portas e serviços disponíveis no host.

Usando o modo “verbose” “-v” para exibir mais informações do alvo. Utilize “-vv” para ter uma saída de informações mais detalhadas.

```
hacker@exin-eth:~
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@exin-eth:/# nmap -v wikipedia.org

Starting Nmap 6.00 ( http://nmap.org ) at 2015-09-30 09:47 BRT
Initiating Ping Scan at 09:47
Scanning wikipedia.org (208.80.154.224) [4 ports]
Completed Ping Scan at 09:47, 0.15s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:47
Completed Parallel DNS resolution of 1 host. at 09:47, 0.01s elapsed
Initiating SYN Stealth Scan at 09:47
Scanning wikipedia.org (208.80.154.224) [1000 ports]
Discovered open port 80/tcp on 208.80.154.224
Discovered open port 443/tcp on 208.80.154.224
Completed SYN Stealth Scan at 09:47, 13.04s elapsed (1000 total ports)
Nmap scan report for wikipedia.org (208.80.154.224)
Host is up (0.15s latency).
rDNS record for 208.80.154.224: text-lb.eqiad.wikimedia.org
Not shown: 990 closed ports
PORT      STATE SERVICE
22/tcp    filtered ssh
25/tcp    filtered smtp
80/tcp    open  http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
179/tcp   filtered bgp
443/tcp   open  https
445/tcp   filtered microsoft-ds
1434/tcp  filtered ms-sql-m
5666/tcp  filtered nrpe

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 13.72 seconds
Raw packets sent: 1078 (47.408KB) | Rcvd: 1054 (42.196KB)
root@exin-eth:/#
```

Alguns exemplos de comandos nmap:

Reconhecendo o alvo com o nmap

Primeiramente descubra qual o número IP de sua máquina virtual Metasploitable. Lembre que a interface virtual deverá estar configurada em modo “Bridge”. Realize a autenticação na máquina virtual com o usuário **msfadmin** e a senha também **msfadmin**.

Para efeitos de exemplo, vou assumir que o IP da máquina virtual metasploitable seja **192.168.68.109**. Lembre de alterá-lo para o número de sua máquina quando for executar algum comando.

Exemplo 1: Descobrindo as portas abertas de um host

Vamos descobrir quais portas de comunicação TCP estão abertas no alvo.

`nmap -sT 192.168.68.109`

```
root@avelino-XPS-13-9350:/# nmap -sT 192.168.68.120
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-29 11:21 -03
Nmap scan report for 192.168.68.120
Host is up (0.00034s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:F1:A5:DE (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
```

A opção “-s” no script é o comando para o escaneamento. Já a opção “T”, indica o escaneamento de portas TCP. Caso for necessário escanear as portas UDP, é alterar o T pelo U.

A saída do comando apresentada na figura apresenta 3 colunas: o número da porta aberta, seu estado e o possível serviço que está sendo executado nesta porta.

Estado das portas

Aberta (open) - está ativamente aceitando conexões TCP ou pacotes UDP nesta porta;

Fechado (closed) - Uma porta fechada está acessível (ela recebe e responde a pacotes de sondagens do Nmap), mas não há nenhuma aplicação ouvindo nela.

Filtrado (filtered) - O Nmap não consegue determinar se a porta está aberta porque uma filtragem de pacotes impede que as sondagens alcancem a porta.

Exemplo 2: Descobrindo as versões dos serviços em execução

Comando:

`nmap -sV 192.168.68.109`

```
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-29 11:32 -03
Nmap scan report for 192.168.68.120
Host is up (0.00044s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rshd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:F1:A5:DE (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.56 seconds
```

Observe que na saída do comando é acrescentada uma quarta coluna, onde a versão do serviço em execução é apresentado.

Exemplo 3: Descobrindo o Sistema Operacional

```
nmap -O 192.168.68.109
```

```
MAC Address: 08:00:27:F1:A5:DE (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
```

A opção “-O” tenta descobrir qual a versão do sistema operacional do host alvo.

Exemplo 4: selecionando as portas a serem escaneadas

É possível você uma porta ou várias portas a serem escaneadas. Para isso usamos a opção “-p”. No primeiro exemplo vamos escanear apenas a porta 80. Já no segundo exemplo iremos escanear as portas 445 e 22.

```
nmap -sV -p 80 192.168.68.109
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-29 16:32 -03
Nmap scan report for 192.168.68.120
Host is up (0.00061s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
MAC Address: 08:00:27:F1:A5:DE (Oracle VirtualBox virtual NIC)
```

```
nmap -sV -p 445,22 192.168.68.109
```

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
445/tcp    open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 08:00:27:F1:A5:DE (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Nmap Scripting Engine (NSE)

Oferece um conjunto totalmente novo de recursos e confere uma nova dimensão para o Nmap. Permite que o Nmap conclua uma série de tarefas, incluindo scanning de vulnerabilidades, detecção de backdoors e em alguns casos a exploração de vulnerabilidades.

A seguir serão apresentados alguns exemplos e exercícios para a prática do nmap.

Para descoberta de vulnerabilidades

```
nmap -sV - --script vuln 192.168.68.109
```

Encontrar malware ou backdoor

```
nmap -v --script malware 192.168.68.109
```

Exercício 1.1.e – Listar as vulnerabilidades das portas 21 e 445

Exercício 1.1.f – Encontrar um exploit para uma vulnerabilidade nos serviços testados no exercício anterior.

Exercício 1.1.g – Encontrar uma CVE classificada como alta para os serviços das portas 3306 e 5432.

Nikto

Uma Ferramenta Open Source para Análise de Vulnerabilidades em Servidores Web, um scanner que executa teste completos contra servidores, incluindo mais de 6500 arquivos /CGIs perigosos, controles de versão não atualizados de mais de 1250 servidores. Foi desenvolvida em Perl, por Chris Solo e David Lodge, para validação de vulnerabilidade, verificando versões desatualizadas de servidores web, que procura mostra softwares e plugins desatualizados.

Permite gerar relatórios em diferentes formatos como: txt, html, csv, msf, xml

Sintaxe:

```
nikto -host [endereço] [opções]
```

Exemplo:

```
nikto -h 192.168.68.109
```

```
- Nikto v2.1.5
-----
+ Target IP:      192.168.68.109
+ Target Hostname: 192.168.68.109
+ Target Port:    80
+ Start Time:     2021-06-30 17:56:06 (GMT-3)
-----
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.2.22). Apache 1.3.42 (final release) and 2.0.64 are also current.
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3233: /phpinfo.php: Contains PHP configuration information
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /index.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that co
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ Cookie phpMyAdmin created without the httponly flag
+ OSVDB-3092: /phpMyAdmin/: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting...
+ OSVDB-3268: /icons/: Directory indexing found.
+ Server leaks inodes via ETags, header found with file /icons/README, inode: 412190, size: 5108, mtime: 0x438c0358aae80
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpMyAdmin/: phpMyAdmin directory found
+ 6544 items checked: 0 error(s) and 18 item(s) reported on remote host
+ End Time:      2021-06-30 17:56:19 (GMT-3) (13 seconds)
-----
+ 1 host(s) tested
```


Opções:

Display – mostra os controles de saída do nikto

- 1- Mostra os redirecionamentos
 - 2- Mostra cookies recebidos
 - 3- Mostra todas as respostas
 - 4- Mostramos URLs que exigem autenticação
- D- Saída de depuração
V- Saída detalhada

exemplo para visualizar redirecionamentos no host alvo:

nikto -h 192.168.68.109 -Display 1

```
- Nikto v2.1.5
-----
+ Target IP: 192.168.68.109
+ Target Hostname: 192.168.68.109
+ Target Port: 80
+ Start Time: 2021-06-30 17:52:58 (GMT-3)
-----
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.2.22). Apache 1.3.42 (final release) and 2.0.64 are also current.
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3233: /phpinfo.php: Contains PHP configuration information
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /index.php?PHPBB5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ Cookie phpMyAdmin created without the httponly flag
+ OSVDB-3092: /phpMyAdmin/: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting...
+ /test - Redirects (301) to http://192.168.68.109/test/ , Apache Tomcat default file found. All default files should be removed.
+ OSVDB-3268: /icons/: Directory indexing found.
+ Server leaks inodes via ETags, header found with file /icons/README, inode: 412190, size: 5108, mtime: 0x438c0358aae80
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpMyAdmin/setup - Redirects (301) to http://192.168.68.109/phpMyAdmin/setup/ , phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. The /setup/ directory may reveal details about the install application and databases.
+ /phpMyAdmin/: phpMyAdmin directory found
+ 6544 items checked: 0 error(s) and 18 item(s) reported on remote host
+ End Time: 2021-06-30 17:53:13 (GMT-3) (15 seconds)
-----
+ 1 host(s) tested
```



Tuning – as opções de teste que o Nikto usará contra um alvo.

- 0 - Upload de arquivo
- 1 - Arquivo Interessante / Visto nos logs
- 2 - Configuração incorreta / arquivo padrão
- 3 - Divulgação de Informações
- 4 - Injeção (XSS / Script / HTML)
- 5 - Recuperação Remota de Arquivos - Raiz Interna da Web
- 6 - Negação de Serviço
- 7 - Recuperação Remota de Arquivos - Server Wide
- 8 - Execução de Comando / Shell Remoto
- 9 - Injeção SQL
 - a - Autenticação Desvio
 - b - Identificação de Software
 - c - Inclusão remota de fontes

Exercício 1.1.h - Realize uma consulta ao nome www.ietf.org, e responda:

- a. Qual é o endereço IP associado?
- b. Quais são seus servidores DNS?
- c. Existe algum servidor de e-mail associado ao domínio ietf.org? Qual o seu nome e IP?

Exercício 1.1.i - Escolha um site na Internet e responda as seguintes perguntas:

- a) Quais servidores DNS são responsáveis por este domínio? (print a sua consulta)
- b) Existem outros domínios ou serviços hospedados no mesmo host (IP)? Quais são?
- c) Qual o Servidor WEB e Sistema Operacional que hospedam este site? Quais foram as últimas alterações?
- d) Quais tecnologias (jquery, utilizadas por este site)?
- e) Existe algum WAF protegendo este site? (Print a saída do comando)
- f) O Domínio possui um servidor de e-mail configurado? Qual (is) Ip (s)?

Indicação para pesquisa:

Capítulo 1 : O'CONNOR, T. J. Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers, 2012, ISBN-13: 978-1597499576

DUFFY, Christopher. Aprendendo Pentest com Python. Novatec, 2015, ISBN: 978-85-7522-505-9

Exercício 1.1.j

Utilizando sua ferramenta (port scan) descubra quais portas TCP e UDP estão abertas no alvo, bem como os serviços que estão associados nestes. Pesquise e anote em seu diário de bordo as vulnerabilidades comuns conhecidas (CVE) do processo que está gerenciando a porta 21.

Observação: é obrigatório o uso da ferramenta desenvolvida no exercício anterior para a conclusão deste exercício. Evidencie por meio de prints e insira no diário de bordo.

Prazo para entrega: 23/03/2023

Não será aceito a entrega dos roteiros após os prazos estabelecidos.