PABLOCYBERSEC

CYBER SECURITY 1.0

Do Zero ao Avançado

Fundamentos, Hacking Ético, Pentest
e Cibersegurança

TEORIA · PRÁTICA · LABORATÓRIO
EXERCÍCIO · DESAFIO

77 capítulos · 9 partes · 5 níveis · Projeto final de pentest

PABLOCYBERSEC

PabloCyberSec Cybersecurity 1.0

Do Zero ao Avançado: Fundamentos, Hacking Ético, Pentest e Cibersegurança

Primeira edição, versão 1.0

© PabloCyberSec. Todos os direitos reservados. Nenhuma parte desta obra pode ser reproduzida, distribuída ou revendida, no todo ou em parte, por qualquer meio, sem autorização prévia e por escrito do autor.

Este material tem finalidade exclusivamente educacional. Pratique somente em ambientes próprios ou com autorização expressa. Os nomes de produtos e marcas citados pertencem aos seus respectivos proprietários.

Sumário

Fundamentos 8

O que é Segurança da Informação (InfoSec) . . . 9
O que é Cybersecurity . . . 10
Ética, legislação e autorização . . . 11
Confidencialidade, Integridade e Disponibilidade (CIA) . . . 12
Ameaças, vulnerabilidades e riscos . . . 13
Malware, phishing, engenharia social e ataques . . . 14
Autenticação, autorização e controle de acesso . . . 16
Criptografia: conceitos fundamentais . . . 17

Redes 19

O que é uma rede . . . 20
Modelos OSI e TCP/IP . . . 21
IP, máscara, gateway e DNS . . . 22
TCP e UDP . . . 23
Portas e serviços . . . 24
HTTP e HTTPS . . . 25
DHCP, ARP e ICMP . . . 26
Sub-redes e conceitos essenciais . . . 27

Linux 29

Conhecendo o Linux . . . 30
Terminal e shell . . . 31
Navegação pelo sistema . . . 32
Arquivos e diretórios . . . 33
Permissões . . . 34
Usuários e grupos . . . 35
Processos e serviços . . . 36
Instalação de ferramentas . . . 37
Comandos fundamentais . . . 38

Programação para Cybersecurity 40

Lógica de programação . . . 41
Python para iniciantes . . . 42
Variáveis, condições e loops . . . 43
Funções . . . 44
Manipulação de arquivos . . . 45
Redes com Python . . . 46
Automação para segurança . . . 48

Hacking Ético 50

O que é Ethical Hacking . . . 51
Metodologia de Pentest . . . 52
Reconhecimento . . . 53
Enumeração . . . 54
Scanning . . . 55
Identificação de serviços . . . 56
Vulnerabilidades . . . 57
Exploração em laboratório . . . 58
Pós-exploração e documentação . . . 60
Relatório de Pentest . . . 61

Web Security 62

Como funciona uma aplicação Web . . . 63
HTTP na prática . . . 64
Cookies e sessões . . . 65
Autenticação . . . 66
OWASP Top 10 . . . 67
SQL Injection . . . 68
XSS (Cross-Site Scripting) . . . 69
Controle de acesso . . . 70
Vulnerabilidades de API . . . 71

Ferramentas 73

Nmap . . . 74
Wireshark . . . 75
Burp Suite . . . 76
Metasploit . . . 77
Gobuster . . . 78
Ferramentas de enumeração . . . 79
Como escolher a ferramenta correta . . . 80

Nível Avançado 82

Exploração de vulnerabilidades . . . 83
Privilege Escalation . . . 84
Buffer Overflow: fundamentos . . . 85
Engenharia Reversa: introdução . . . 86
Red Team: conceitos . . . 87
OPSEC . . . 88
Persistência: conceitos defensivos e ofensivos em laboratório . . . 90
Evasão e detecção . . . 91
Introdução à análise de malware . . . 92

Projeto Final 94

Montando um laboratório de Cibersegurança . . . 95
Criando um cenário de Pentest autorizado . . . 96
Projeto final: Reconhecimento . . . 97
Projeto final: Enumeração . . . 98
Projeto final: Identificação das vulnerabilidades . . . 99
Projeto final: Exploração controlada . . . 100
Projeto final: Análise dos resultados . . . 101
Projeto final: Correção das vulnerabilidades . . . 102
Projeto final: Relatório profissional . . . 103
Checklist do Pentester . . . 104

Apresentação

Bem-vindo ao PabloCyberSec Cybersecurity 1.0. Este livro foi pensado para quem quer entrar na área de segurança da informação começando do zero, sem exigir conhecimento prévio, e evoluir passo a passo até conceitos avançados.

A ideia central é simples: antes de digitar um comando, você precisa entender o motivo dele. Por isso, os primeiros capítulos têm mais teoria. Você aprende o que é uma porta, o que é TCP, o que é um serviço e o que é reconhecimento antes de ver o Nmap em ação. Assim, você aprende o raciocínio e não apenas copia comandos.

O livro percorre nove partes: fundamentos, redes, Linux, programação, hacking ético, web security, ferramentas, nível avançado e um projeto final completo. São 77 capítulos que se constroem uns sobre os outros.

Como usar este livro

Cada capítulo segue a mesma trilha de aprendizado: Aprenda (a explicação), Na prática (comandos e exemplos), Laboratório (atividade guiada), Exercícios (perguntas de verificação), Desafio PabloCyberSec (uma tarefa maior para resolver sozinho) e Correção (as respostas e o raciocínio esperado).

Leia na ordem. Faça os laboratórios no computador, e não apenas na cabeça. Tente responder os exercícios antes de olhar a correção. Use o Diário de Laboratório, no final do livro, para anotar comandos, dúvidas e descobertas. Quem escreve o que aprende, aprende duas vezes.

Ao longo do livro, os capítulos são marcados com um nível de 1 a 5. O Desafio Final reúne tudo em um projeto de pentest autorizado.

Os cinco níveis

Nível 1 · Iniciante. Fundamentos de InfoSec e Cybersecurity

Nível 2 · Fundamentos. Redes, Linux e comandos

Nível 3 · Praticante. Programação, ferramentas e laboratórios

Nível 4 · Pentester Júnior. Metodologia, Web Security e exploração controlada

Nível 5 · Avançado. Privilege Escalation, Engenharia Reversa, Red Team e análise de malware em laboratório

Aviso legal e ético

Todo o conteúdo deste livro tem finalidade educacional e de defesa. As técnicas apresentadas só devem ser praticadas em ambientes próprios, em laboratórios isolados ou em sistemas para os quais exista autorização expressa e por escrito. Acessar, testar ou interferir em sistemas de terceiros sem permissão é crime e pode gerar responsabilização civil e penal.

O autor e a PabloCyberSec não se responsabilizam por uso indevido das informações. Respeite as leis do seu país, os termos de uso das plataformas de treinamento e a privacidade das pessoas. Os endereços IP citados nos exemplos pertencem a um laboratório fictício e privado.

O que você precisa para começar

Um computador com pelo menos 8 GB de memória e 100 GB livres de disco, um programa de virtualização gratuito (VirtualBox ou VMware Player), uma conexão com a internet para baixar as imagens oficiais (Kali Linux, Ubuntu, Metasploitable) e, principalmente, curiosidade e disciplina.

O ambiente completo será montado passo a passo: o Capítulo 17 apresenta o Linux, o Capítulo 24 mostra como instalar ferramentas, e o Capítulo 68 consolida o laboratório do projeto final.

PARTE 1

Fundamentos

NÍVEL 1 · INICIANTE

A base de tudo: o que é segurança, quais são os conceitos que sustentam a área e por que a ética vem antes da técnica.

01 O que é Segurança da Informação (InfoSec)
02 O que é Cybersecurity
03 Ética, legislação e autorização
04 Confidencialidade, Integridade e Disponibilidade (CIA)
05 Ameaças, vulnerabilidades e riscos
06 Malware, phishing, engenharia social e ataques
07 Autenticação, autorização e controle de acesso
08 Criptografia: conceitos fundamentais

CAPÍTULO 1 · NÍVEL 1 · INICIANTE

O que é Segurança da Informação (InfoSec)

APRENDA

Segurança da Informação, ou InfoSec, é a prática de proteger informações contra acesso indevido, alteração, destruição ou perda, em qualquer formato: arquivos digitais, papéis impressos, conversas ou dados armazenados na nuvem. Note que a informação é o foco, e não apenas o computador.

Toda organização possui ativos de informação: cadastros de clientes, senhas, contratos, código-fonte, fotos, e-mails. Esses ativos têm valor, e quem tem valor atrai interesse de terceiros. A InfoSec organiza três frentes de trabalho: pessoas (treinamento e conscientização), processos (regras e procedimentos) e tecnologia (ferramentas e configurações). Quando uma delas falha, as outras ficam expostas.

InfoSec é um conceito mais amplo que Cybersecurity. Trancar um armário com documentos físicos é InfoSec, mas não é Cybersecurity. Você verá essa diferença com detalhes no próximo capítulo.

NA PRÁTICA

Pense em três ativos que você possui hoje, como seu e-mail, as fotos do celular e a conta bancária. Para cada um, responda: quem não deveria ter acesso? O que aconteceria se fosse apagado? O que aconteceria se fosse alterado sem você perceber?

LABORATÓRIO

Faça um inventário simples em uma planilha com cinco colunas: ativo, onde fica armazenado, quem tem acesso, impacto se vazar e impacto se for perdido. Liste pelo menos oito ativos pessoais ou da sua empresa.

EXERCÍCIOS

1. Cite dois exemplos de InfoSec que não envolvem computadores.

2. Quais são as três frentes de trabalho da InfoSec?

3. Por que a informação, e não o equipamento, é o centro da segurança?

DESAFIO PABLOCYBERSEC

Escolha um ativo da sua lista e proponha uma medida de proteção para cada frente: pessoas, processos e tecnologia.

Correção: *1) Trancar documentos em armário e triturar papéis. 2) Pessoas, processos e tecnologia. 3) O equipamento pode ser substituído, a informação nem sempre.*

Boa prática: *Segurança não é um produto que se compra, é um processo contínuo.*

CAPÍTULO 2 · NÍVEL 1 · INICIANTE

O que é Cybersecurity

APRENDA

Cybersecurity, ou segurança cibernética, é o ramo da InfoSec dedicado a proteger sistemas, redes, aplicações e dados que existem no ambiente digital. Ela lida com ameaças como invasões, ransomware, vazamentos de dados e fraudes online.

Os profissionais da área se dividem em grandes times. O time ofensivo (Red Team e pentesters) simula ataques para encontrar falhas antes dos criminosos. O time defensivo (Blue Team) monitora, detecta e responde a incidentes. Existe ainda a atuação em governança, risco e conformidade (GRC), que cuida de políticas, auditorias e leis.

Este livro percorre os dois lados. Para defender bem, é preciso entender como se ataca. Para atacar com ética, é preciso entender o impacto sobre quem defende. Em todos os casos, o trabalho acontece sempre com autorização.

NA PRÁTICA

Abra o gerenciador de tarefas do seu computador e identifique três processos que usam a rede. Pergunte-se: por que este programa se conecta à internet? O que ele envia? Esse é o raciocínio básico de quem trabalha com segurança.

LABORATÓRIO

Monte uma tabela comparando InfoSec e Cybersecurity com exemplos de cada uma. Em seguida, pesquise três vagas da área e anote quais pertencem ao lado ofensivo, ao defensivo e ao de governança.

EXERCÍCIOS

1. Qual a diferença entre Red Team e Blue Team?

2. Dê um exemplo de atividade de GRC.

3. Por que um defensor precisa conhecer técnicas de ataque?

DESAFIO PABLOCYBERSEC

Em um parágrafo, explique para um amigo leigo o que um profissional de cybersecurity faz no dia a dia.

Correção: *1) Red simula ataques, Blue detecta e responde. 2) Auditoria, política de senhas, análise de riscos. 3) Para antecipar os caminhos que um invasor usaria.*

CAPÍTULO 3 · NÍVEL 1 · INICIANTE

Ética, legislação e autorização

APRENDA

Tudo o que você aprender neste livro pode ajudar ou prejudicar pessoas. O que separa um profissional de um criminoso é a autorização. Testar um sistema sem permissão expressa é ilegal, mesmo que a intenção seja boa e mesmo que nada seja danificado.

No Brasil, a Lei 12.737/2012 (Lei Carolina Dieckmann) tipifica a invasão de dispositivos informáticos, a Lei 12.965/2014 (Marco Civil da Internet) estabelece direitos e deveres online, e a Lei 13.709/2018 (LGPD) regula o tratamento de dados pessoais. Outros países têm regras próprias, como o CFAA nos Estados Unidos. Consulte sempre um advogado antes de contratos de teste de segurança.

Um teste profissional começa por um documento: o escopo e as regras de engajamento. Ele define quais sistemas podem ser testados, em quais horários, quais técnicas são permitidas, quem é o contato em caso de problema e como os dados encontrados serão tratados. Sem esse documento, não há teste.

NA PRÁTICA

Modelo mínimo de autorização: nome da empresa contratante, nome do testador, lista de IPs e domínios no escopo, período de testes, técnicas proibidas (por exemplo, negação de serviço), contato de emergência e assinatura de ambas as partes.

LABORATÓRIO

Redija uma autorização fictícia para um teste em um laboratório seu, com todos os itens do modelo acima. Guarde-a: você a usará no projeto final.

EXERCÍCIOS

1. O que torna um teste de segurança legal?

2. O que é escopo?

3. Que lei brasileira trata de dados pessoais?

DESAFIO PABLOCYBERSEC

Leia um termo de uso de um programa de recompensa por vulnerabilidades (bug bounty) e liste o que é permitido e proibido.

Correção: *1) Autorização expressa e por escrito. 2) A lista do que pode ser testado. 3) A LGPD (Lei 13.709/2018).*

Boa prática: *Na dúvida, pergunte. Se não está escrito que pode, considere que não pode.*

CAPÍTULO 4 · NÍVEL 1 · INICIANTE

Confidencialidade, Integridade e Disponibilidade (CIA)

APRENDA

A tríade CIA é a base de toda a segurança. Confidencialidade significa que a informação só é acessada por quem tem direito. Integridade significa que ela não é alterada de forma indevida e permanece correta. Disponibilidade significa que ela está acessível quando é necessária.

Cada incidente costuma atingir um ou mais desses pilares. Um vazamento de banco de dados fere a confidencialidade. Um fraudador que altera o valor de um boleto fere a integridade. Um ataque de ransomware que cifra os arquivos de um hospital fere a disponibilidade, e muitas vezes também a confidencialidade, se os dados forem copiados antes.

Os pilares podem entrar em conflito. Trancar tudo com muitas camadas protege a confidencialidade, mas pode prejudicar a disponibilidade. O trabalho do profissional é encontrar o equilíbrio adequado ao valor da informação.

NA PRÁTICA

Exemplo de ferramentas por pilar: criptografia e controle de acesso (confidencialidade); hash e assinaturas digitais (integridade); backups, redundância e planos de recuperação (disponibilidade).

```
# Verificando integridade de um arquivo com hash
sha256sum documento.pdf
# Guarde o resultado. Se o arquivo mudar, o hash muda.
```

LABORATÓRIO

Crie um arquivo de texto, gere o seu hash SHA-256, altere uma única letra e gere o hash novamente. Compare os resultados.

EXERCÍCIOS

1. Qual pilar é atingido quando alguém altera uma nota fiscal?

2. Dê um exemplo de medida que protege a disponibilidade.

3. Por que os pilares podem entrar em conflito?

DESAFIO PABLOCYBERSEC

Para um sistema de prontuário médico, ordene os três pilares por prioridade e justifique.

Correção: *1) Integridade. 2) Backups, redundância de servidores, nobreak. 3) Reforçar um pilar pode dificultar outro, como mais barreiras de acesso reduzindo a facilidade de uso.*

CAPÍTULO 5 · NÍVEL 1 · INICIANTE

Ameaças, vulnerabilidades e riscos

APRENDA

Três palavras que parecem iguais, mas não são. Uma ameaça é algo ou alguém que pode causar dano: um criminoso, um funcionário descuidado, uma enchente. Uma vulnerabilidade é uma fraqueza que pode ser explorada: um software desatualizado, uma senha fraca, uma porta destrancada. Um risco é a combinação das duas, com a probabilidade e o impacto.

Uma forma comum de expressar isso: Risco = Ameaça × Vulnerabilidade × Impacto. Se não há vulnerabilidade, a ameaça não encontra caminho. Se o impacto é mínimo, o risco é baixo mesmo que a falha exista. Por isso nem toda vulnerabilidade merece a mesma urgência.

Diante de um risco, há quatro respostas clássicas: mitigar (reduzir), transferir (seguro, terceirização), aceitar (quando o custo de tratar é maior que o dano) e evitar (deixar de realizar a atividade).

NA PRÁTICA

Exemplo: um servidor de arquivos usa um sistema sem atualizações (vulnerabilidade). Criminosos varrem a internet atrás desse tipo de sistema (ameaça). Ele guarda dados de clientes (impacto alto). Resultado: risco alto, tratamento prioritário.

LABORATÓRIO

Monte uma matriz de riscos com 5 itens do seu inventário do Capítulo 1. Atribua probabilidade e impacto de 1 a 5, multiplique e ordene do maior para o menor.

EXERCÍCIOS

1. Qual a diferença entre ameaça e vulnerabilidade?

2. Cite as quatro respostas a um risco.

3. Por que nem toda vulnerabilidade é prioridade máxima?

DESAFIO PABLOCYBERSEC

Classifique este caso: uma loja online não faz backup e fica 3 dias fora do ar. Identifique ameaça, vulnerabilidade, impacto e a melhor resposta ao risco.

Correção: *1) Ameaça é a fonte do dano, vulnerabilidade é a fraqueza. 2) Mitigar, transferir, aceitar, evitar. 3) Depende da probabilidade e do impacto no negócio.*

CAPÍTULO 6 · NÍVEL 1 · INICIANTE

Malware, phishing, engenharia social e ataques

APRENDA

Malware é todo software criado para causar dano ou obter vantagem indevida. Os tipos mais comuns: vírus (anexam-se a arquivos), worms (se espalham sozinhos pela rede), trojans (fingem ser úteis), ransomware (cifra dados e exige pagamento), spyware (espiona), keyloggers (registram o que é digitado) e botnets (redes de máquinas controladas remotamente).

Engenharia social explora pessoas, e não máquinas. Phishing é o e-mail ou mensagem falsa que tenta levar a vítima a clicar, informar dados ou instalar algo. Variações: spear phishing (alvo específico), vishing (por voz), smishing

(por SMS) e pretexting (história inventada para ganhar confiança). A maior parte dos incidentes começa por uma pessoa enganada.

Outros ataques frequentes: negação de serviço (DoS/DDoS), que sobrecarrega um serviço; ataque de força bruta, que testa senhas em sequência; e man-in-the-middle, em que um terceiro intercepta a comunicação. O bom é que a maioria é evitável com hábitos simples.

NA PRÁTICA

Sinais de phishing: urgência exagerada, erros de português, remetente estranho, link cujo endereço real difere do texto exibido, pedido de senha ou pagamento fora do canal oficial.

LABORATÓRIO

Observe sem clicar: passe o mouse sobre links de e-mails suspeitos da sua caixa de spam e compare o texto com o endereço real exibido no rodapé do navegador. Anote três sinais de golpe.

EXERCÍCIOS

1. O que diferencia um worm de um vírus?

2. O que é spear phishing?

3. Cite três sinais de que uma mensagem pode ser phishing.

DESAFIO PABLOCYBERSEC

Escreva uma lista de cinco regras que você daria à sua família para evitar golpes por mensagem.

Correção: *1) O worm se espalha sozinho, o vírus precisa de um arquivo hospedeiro. 2) Phishing direcionado a uma pessoa ou grupo específico. 3) Urgência, link divergente, pedido de dados sensíveis.*

Boa prática: *Desconfie de qualquer mensagem que cria pressa. A urgência é a arma do golpista.*

CAPÍTULO 7 · NÍVEL 1 · INICIANTE

Autenticação, autorização e controle de acesso

APRENDA

Autenticação responde à pergunta: quem é você? Autorização responde: o que você pode fazer? Auditoria (ou accounting) registra: o que você fez? Juntas formam o modelo AAA. Confundir autenticação com autorização é fonte de muitas falhas, como veremos no capítulo de Web Security.

Os fatores de autenticação são três: algo que você sabe (senha), algo que você tem (celular, token) e algo que você é (biometria). A autenticação multifator (MFA) combina pelo menos dois fatores diferentes, e reduz drasticamente o sucesso de roubo de senhas.

Na autorização, vale o princípio do menor privilégio: cada pessoa ou programa recebe apenas o acesso necessário para sua função, e nada além. Modelos comuns de controle de acesso são o baseado em papéis (RBAC), em que permissões ficam ligadas a funções, e o baseado em atributos (ABAC), em que regras consideram contexto como horário e local.

NA PRÁTICA

Boas práticas de senha: frases longas em vez de palavras complexas curtas, uma senha diferente para cada serviço, uso de um gerenciador de senhas e MFA ativado em todas as contas importantes, de preferência com aplicativo autenticador em vez de SMS.

LABORATÓRIO

Ative a autenticação em dois fatores em seu e-mail principal e em uma rede social. Instale um gerenciador de senhas e troque a senha de três contas importantes por frases longas e únicas.

EXERCÍCIOS

1. Qual a diferença entre autenticação e autorização?

2. Dê um exemplo de cada fator de autenticação.

3. O que significa o princípio do menor privilégio?

DESAFIO PABLOCYBERSEC

Desenhe uma tabela de papéis para um pequeno sistema com três perfis (visitante, usuário, administrador) e as ações permitidas a cada um.

Correção: *1) Autenticar é provar identidade, autorizar é definir permissões. 2) Senha, token no celular, impressão digital. 3) Dar só o acesso necessário para a tarefa.*

CAPÍTULO 8 · NÍVEL 1 · INICIANTE

Criptografia: conceitos fundamentais

APRENDA

Criptografia é a ciência de transformar informação legível em ilegível, de modo que apenas quem possui a chave correta consiga reverter o processo. Ela protege a confidencialidade e, em certas formas, a integridade e a autenticidade.

Na criptografia simétrica, a mesma chave cifra e decifra. É rápida e usada para grandes volumes, como o algoritmo AES. O desafio é compartilhar a chave com segurança. Na criptografia assimétrica há um par de chaves: a pública, que pode ser divulgada, e a privada, que fica em segredo. O que uma cifra, só a outra decifra. É a base de certificados e assinaturas digitais (RSA, curvas elípticas).

Hash não é criptografia, é uma função de mão única: transforma qualquer dado em uma impressão digital de tamanho fixo, impossível de reverter na prática. Servem para verificar integridade e para armazenar senhas, sempre com sal (salt) e algoritmos próprios para isso, como bcrypt ou Argon2. Algoritmos antigos como MD5 e SHA-1 não devem ser usados para segurança.

O HTTPS combina tudo: o navegador usa criptografia assimétrica para negociar uma chave de sessão e, a partir daí, usa criptografia simétrica para a conversa.

NA PRÁTICA

Regra de ouro: nunca invente seu próprio algoritmo de criptografia. Use bibliotecas e padrões revisados por especialistas.

```
# Gerar hash e cifrar um arquivo com OpenSSL
echo "mensagem secreta" | sha256sum
openssl enc -aes-256-cbc -pbkdf2 -in segredo.txt -out segredo.enc
openssl enc -d -aes-256-cbc -pbkdf2 -in segredo.enc -out aberto.txt
```

LABORATÓRIO

Cifre um arquivo de texto com OpenSSL, confira que o conteúdo ficou ilegível, decifre e compare o hash do original com o do arquivo recuperado.

EXERCÍCIOS

1. Qual a diferença entre criptografia simétrica e assimétrica?

2. Por que hash não é criptografia?

3. Por que MD5 não é recomendado para senhas?

DESAFIO PABLOCYBERSEC

Explique, com suas palavras, o que acontece nos primeiros segundos quando você acessa um site com HTTPS.

Correção: *1) Simétrica usa uma chave, assimétrica usa um par. 2) Hash não tem volta, criptografia tem. 3) É rápido demais e possui falhas conhecidas, facilitando quebra.*

PARTE 2

Redes

NÍVEL 2 · FUNDAMENTOS

Como os computadores conversam. Sem entender redes, as ferramentas viram caixas-pretas.

09 O que é uma rede
10 Modelos OSI e TCP/IP
11 IP, máscara, gateway e DNS
12 TCP e UDP
13 Portas e serviços
14 HTTP e HTTPS
15 DHCP, ARP e ICMP
16 Sub-redes e conceitos essenciais

CAPÍTULO 9 · NÍVEL 2 · FUNDAMENTOS

O que é uma rede

APRENDA

Uma rede de computadores é um conjunto de dispositivos conectados para trocar dados. Pode ser pequena, como a do seu celular com o roteador (LAN), ou gigantesca, como a internet (WAN). Os dispositivos são chamados de hosts: computadores, celulares, câmeras, impressoras, servidores.

Para conectar, usamos equipamentos. O switch liga dispositivos dentro da mesma rede local. O roteador liga redes diferentes entre si e encaminha pacotes pelo melhor caminho. O access point oferece Wi-Fi. O firewall filtra o tráfego conforme regras.

Em geral, um computador atua como cliente (pede um serviço) e outro como servidor (entrega o serviço). Quando você abre um site, seu navegador é o cliente e o servidor web responde. Entender essa troca é a base para entender tanto o funcionamento quanto os ataques.

NA PRÁTICA

Topologia caseira típica: celular e notebook conectam ao roteador de casa por Wi-Fi, o roteador conecta ao provedor, e o provedor conecta à internet.

```
# Ver informações de rede no Linux
ip a
# Testar conectividade com outro host
ping -c 4 8.8.8.8
```

LABORATÓRIO

Desenhe em papel a rede da sua casa, incluindo todos os dispositivos conectados. Depois acesse o painel do seu roteador e compare com a lista de dispositivos.

EXERCÍCIOS

1. Qual a diferença entre switch e roteador?

2. O que são LAN e WAN?

3. O que é um host?

DESAFIO PABLOCYBERSEC

Identifique no desenho quais dispositivos funcionam como cliente e quais como servidor.

Correção: *1) Switch liga dispositivos da mesma rede, roteador liga redes diferentes. 2) Rede local e rede de longa distância. 3) Qualquer dispositivo conectado à rede.*

CAPÍTULO 10 · NÍVEL 2 · FUNDAMENTOS

Modelos OSI e TCP/IP

APRENDA

Modelos de camadas dividem a comunicação em etapas. Cada camada tem uma função e conversa com a camada vizinha, o que facilita entender e diagnosticar problemas. O modelo OSI tem sete camadas; o TCP/IP, que é o usado na prática, tem quatro.

As sete camadas do OSI, de cima para baixo: 7 Aplicação (HTTP, DNS), 6 Apresentação (formatos e criptografia), 5 Sessão (controle da conversa), 4 Transporte (TCP e UDP), 3 Rede (IP), 2 Enlace (Ethernet, MAC) e 1 Física (cabos, sinais de rádio). No TCP/IP, as camadas 5 a 7 viram Aplicação, a 4 é Transporte, a 3 é Internet e as 1 e 2 formam Acesso à rede.

Quando você envia dados, cada camada acrescenta seu cabeçalho (encapsulamento). No destino, o processo se inverte. Para a segurança, o modelo ajuda a localizar onde está o problema e onde um ataque age: um ataque de phishing atua na camada de aplicação, um ARP spoofing na de enlace, um DDoS volumétrico nas de rede e transporte.

NA PRÁTICA

Truque de memória em inglês, de baixo para cima: Please Do Not Throw Sausage Pizza Away (Physical, Data link, Network, Transport, Session, Presentation, Application).

LABORATÓRIO

Abra o Wireshark (ou use o comando tcpdump) enquanto acessa um site. Identifique em um pacote os cabeçalhos Ethernet, IP, TCP e HTTP, correspondendo a cada camada.

EXERCÍCIOS

1. Quantas camadas tem o modelo OSI?

2. Em qual camada atua o protocolo IP?

3. O que é encapsulamento?

DESAFIO PABLOCYBERSEC

Para os problemas a seguir, indique a camada provável: cabo desconectado, IP errado, site fora do ar, porta bloqueada.

Correção: *1) Sete. 2) Camada 3, Rede. 3) A adição de cabeçalhos em cada camada ao enviar dados. Desafio: física, rede, aplicação, transporte.*

CAPÍTULO 11 · NÍVEL 2 · FUNDAMENTOS

IP, máscara, gateway e DNS

APRENDA

O endereço IP identifica um host em uma rede. O IPv4 usa 32 bits em quatro números de 0 a 255 (como 192.168.1.10). O IPv6 usa 128 bits, criado para suprir a falta de endereços. Alguns blocos são privados e não roteáveis na internet: 10.0.0.0/8, 172.16.0.0/12 e 192.168.0.0/16.

A máscara de sub-rede define qual parte do IP é da rede e qual é do host. Já o gateway padrão é o roteador para onde o computador envia tudo o que não pertence à sua rede local. Sem gateway, você conversa com a vizinhança, mas não chega à internet.

O DNS (Domain Name System) traduz nomes em IPs: ao digitar um site, seu computador pergunta a um servidor DNS qual IP corresponde àquele nome. Por ser a porta de entrada de quase tudo, o DNS é alvo frequente de ataques e também fonte valiosa de informação para reconhecimento.

NA PRÁTICA

Seu IP público (visto pela internet) é diferente do privado (usado dentro da rede). O roteador faz a tradução com NAT.

```
ip a                # IP e máscara das interfaces
ip route            # tabela de rotas e gateway
nslookup exemplo.com # consulta DNS
dig exemplo.com A    # consulta detalhada
```

LABORATÓRIO

Anote seu IP privado, máscara, gateway e servidor DNS. Depois consulte o IP de três sites com dig e compare.

EXERCÍCIOS

1. Qual a função do gateway?

2. O que é um IP privado? Cite uma faixa.

3. O que faz um servidor DNS?

DESAFIO PABLOCYBERSEC

Explique por que você consegue abrir um site pelo IP, mas pode ficar sem acesso quando o DNS falha.

Correção: *1) Encaminhar o tráfego para fora da rede local. 2) Um IP não roteável na internet, como 192.168.0.0/16. 3) Traduz nomes em IPs.*

CAPÍTULO 12 · NÍVEL 2 · FUNDAMENTOS

TCP e UDP

APRENDA

Na camada de transporte, os dois protocolos principais têm filosofias diferentes. O TCP é confiável: estabelece uma conexão, confirma o recebimento, reenvia o que se perdeu e entrega na ordem. Usado em web, e-mail e transferência de arquivos. O UDP é rápido e simples: envia sem confirmar. Usado em streaming, chamadas de voz, jogos e DNS.

O TCP inicia a conexão com o three-way handshake: o cliente envia SYN, o servidor responde SYN-ACK e o cliente confirma com ACK. Só então os dados trafegam. Esse detalhe é essencial para entender varreduras de portas, pois a resposta do servidor ao SYN revela se há um serviço escutando.

Cada pacote TCP carrega flags (SYN, ACK, FIN, RST), números de sequência e portas de origem e destino. O fim da conexão ocorre com FIN, e o RST encerra de forma abrupta.

NA PRÁTICA

Se o TCP é uma ligação telefônica, em que se confirma que o outro lado atendeu, o UDP é uma carta jogada na caixa de correio: rápida, mas sem garantia de que foi lida.

LABORATÓRIO

Com o Wireshark, aplique o filtro tcp.flags.syn==1 e acesse um site. Localize os três pacotes do handshake e anote os números de sequência.

EXERCÍCIOS

1. Cite uma vantagem do UDP sobre o TCP.

2. Quais são os três passos do handshake?

3. Por que o DNS costuma usar UDP?

DESAFIO PABLOCYBERSEC

Explique por que uma chamada de vídeo tolera perda de pacotes melhor do que a transferência de um arquivo.

Correção: *1) Menor atraso e menos sobrecarga. 2) SYN, SYN-ACK, ACK. 3) Consultas pequenas e rápidas, sem necessidade de conexão.*

CAPÍTULO 13 · NÍVEL 2 · FUNDAMENTOS

Portas e serviços

APRENDA

Se o IP é o endereço de um prédio, a porta é o número do apartamento. Uma porta é um número de 0 a 65535 que identifica qual serviço, dentro de um host, deve receber os dados. Um serviço é um programa que escuta em uma porta, como um servidor web na 80.

Portas bem conhecidas: 20/21 FTP, 22 SSH, 23 Telnet, 25 SMTP, 53 DNS, 80 HTTP, 110 POP3, 143 IMAP, 443 HTTPS, 445 SMB, 3306 MySQL, 3389 RDP. Uma porta aberta significa que há um serviço aceitando conexões. Fechada indica que o host responde, mas sem serviço. Filtrada indica que um firewall bloqueia a resposta.

Cada porta aberta é uma superfície de ataque. A regra de ouro de defesa é reduzir essa superfície: desative serviços desnecessários e use firewall. Para um pentester, a lista de portas e serviços é o mapa do território.

NA PRÁTICA

Um serviço pode rodar em porta fora do padrão. Por isso a identificação de serviços, que veremos adiante, não deve confiar apenas no número da porta.

```
ss -tulpn          # serviços escutando neste computador
sudo ss -tulpn | grep :22
```

LABORATÓRIO

Execute ss -tulpn na sua máquina e liste cada serviço em escuta. Para cada um, descreva em uma frase para que serve e se é realmente necessário.

EXERCÍCIOS

1. O que significa porta aberta?

2. Qual a porta padrão do HTTPS?

3. Por que reduzir portas abertas melhora a segurança?

DESAFIO PABLOCYBERSEC

Pesquise o serviço de duas portas desconhecidas da lista e explique o risco de cada uma se exposta à internet.

Correção: *1) Há um serviço escutando e aceitando conexões. 2) 443. 3) Cada porta é um ponto de entrada em potencial.*

CAPÍTULO 14 · NÍVEL 2 · FUNDAMENTOS

HTTP e HTTPS

APRENDA

HTTP é o protocolo da web. O cliente envia uma requisição e o servidor devolve uma resposta. Uma requisição contém um método (GET para obter, POST para enviar dados, PUT, DELETE), um caminho, cabeçalhos e, às vezes, um corpo. A resposta traz um código de status, cabeçalhos e o conteúdo.

Códigos de status importantes: 200 (OK), 301 e 302 (redirecionamento), 400 (requisição inválida), 401 (não autenticado), 403 (proibido), 404 (não encontrado) e 500 (erro no servidor). Cabeçalhos como User-Agent, Cookie e Authorization carregam informações sobre o cliente e a sessão.

O HTTPS é o HTTP dentro de um túnel criptografado (TLS). Isso protege contra espionagem e alteração no caminho, e o certificado digital confirma a identidade do site. O cadeado do navegador mostra que a conexão é cifrada, mas não garante que o site seja confiável: sites de golpe também usam HTTPS.

NA PRÁTICA

Exemplo de requisição: GET /produtos HTTP/1.1, Host: exemplo.com. A resposta começa com HTTP/1.1 200 OK.

```
curl -I https://exemplo.com
curl -v https://exemplo.com 2>&1 | head -30
```

LABORATÓRIO

Use o curl com a opção -v em um site e identifique método, status, cabeçalhos de requisição e de resposta. Em seguida abra as ferramentas de desenvolvedor do navegador (aba Rede) e compare.

EXERCÍCIOS

1. O que significa o código 403?

2. Qual a diferença entre GET e POST?

3. O cadeado do navegador garante que o site é seguro? Por quê?

DESAFIO PABLOCYBERSEC

Descubra, usando curl -I, quais cabeçalhos de segurança três sites diferentes enviam e compare.

Correção: *1) O acesso é proibido ao usuário. 2) GET obtém dados, POST envia dados. 3) Não, só garante que a conexão é cifrada.*

CAPÍTULO 15 · NÍVEL 2 · FUNDAMENTOS

DHCP, ARP e ICMP

APRENDA

O DHCP entrega automaticamente IP, máscara, gateway e DNS aos dispositivos que entram na rede. Sem ele, cada configuração seria manual. O processo é conhecido como DORA: Discover, Offer, Request, Acknowledge.

O ARP resolve, dentro da rede local, qual endereço MAC (físico) corresponde a um IP. O computador pergunta 'quem tem o IP tal?' e o dono responde com seu MAC. Como o ARP não possui autenticação, é alvo do ARP spoofing, em que um dispositivo se passa por outro para interceptar o tráfego. Redes bem configuradas detectam esse comportamento.

O ICMP é um protocolo de controle e diagnóstico. O comando ping usa mensagens ICMP echo para testar se um host responde, e o traceroute usa o TTL para mostrar o caminho dos pacotes. Muitos firewalls bloqueiam ICMP,

então a ausência de resposta ao ping não prova que o host está desligado.

NA PRÁTICA

Todo dispositivo possui um MAC, identificador de 48 bits gravado na placa de rede, ex.: 3C:52:82:AA:10:7F.

```
ip neigh                  # tabela ARP
ping -c 4 192.168.0.1
traceroute 8.8.8.8
```

LABORATÓRIO

Rode ip neigh, identifique os MACs dos dispositivos da sua rede e compare com a lista do roteador. Depois execute um traceroute e conte quantos saltos até o destino.

EXERCÍCIOS

1. O que significa a sigla DORA?

2. Para que serve o ARP?

3. Por que um host pode estar ligado e não responder ao ping?

DESAFIO PABLOCYBERSEC

Explique por que a falta de autenticação no ARP permite ataques dentro da rede local.

Correção: *1) Discover, Offer, Request, Acknowledge. 2) Descobrir o MAC de um IP na rede local. 3) Um firewall pode bloquear o ICMP.*

CAPÍTULO 16 · NÍVEL 2 · FUNDAMENTOS

Sub-redes e conceitos essenciais

APRENDA

Sub-redes dividem uma rede maior em partes menores, melhorando organização, desempenho e segurança. A notação CIDR escreve o IP seguido de uma barra e o número de bits da rede: 192.168.1.0/24 significa que os 24 primeiros bits identificam a rede e os 8 restantes os hosts.

Fórmula prática: em uma rede com n bits de host, há 2^n endereços. Para /24, são 8 bits de host: 256 endereços, dos quais 254 utilizáveis (o primeiro é o endereço da rede, e o último o de broadcast). Um /25 tem 128 endereços, um /26 tem 64 e assim por diante, sempre pela metade.

Segmentar a rede limita o estrago: o setor financeiro não precisa estar na mesma rede dos visitantes. Outros conceitos importantes: NAT (traduz IPs privados para públicos), VLAN (separa redes logicamente no switch), VPN (túnel cifrado entre redes ou usuários), DMZ (zona intermediária para serviços públicos) e firewall (regras de filtragem).

NA PRÁTICA

Exemplo: 192.168.10.0/26. Máscara 255.255.255.192. Faixa utilizável: .1 até .62. Broadcast: .63. Total de hosts: 62.

LABORATÓRIO

Calcule por conta própria: quantos hosts utilizáveis existem em /27? E em /28? Depois divida 192.168.5.0/24 em quatro sub-redes iguais e escreva a faixa de cada uma.

EXERCÍCIOS

1. Quantos hosts utilizáveis há em /24?

2. O que representa o número após a barra na notação CIDR?

3. Por que segmentar a rede ajuda na segurança?

DESAFIO PABLOCYBERSEC

Projete uma rede para uma empresa com os setores: administrativo, financeiro, visitantes e servidores. Defina uma sub-rede para cada um.

Correção: *1) 254. 2) Quantos bits pertencem à parte da rede. 3) Limita o alcance de um invasor. Lab: /27 = 30 hosts, /28 = 14 hosts; as quatro sub-redes são /26: .0, .64, .128, .192.*

PARTE 3

Linux

NÍVEL 2 · FUNDAMENTOS

O sistema operacional da segurança: terminal, arquivos, permissões, processos e comandos essenciais.

17 Conhecendo o Linux
18 Terminal e shell
19 Navegação pelo sistema
20 Arquivos e diretórios
21 Permissões
22 Usuários e grupos
23 Processos e serviços
24 Instalação de ferramentas
25 Comandos fundamentais

CAPÍTULO 17 · NÍVEL 2 · FUNDAMENTOS

Conhecendo o Linux

APRENDA

Linux é um sistema operacional de código aberto, base de quase todos os servidores, dispositivos de rede, sistemas Android e infraestrutura em nuvem. Por isso, quem trabalha com segurança precisa dominá-lo. Tecnicamente, Linux é o kernel (núcleo); o sistema completo se chama distribuição, ou distro, e inclui programas, instalador e gerenciador de pacotes.

Distribuições comuns: Ubuntu e Debian (amigáveis e estáveis), Fedora, Arch (para quem quer controle total) e Kali Linux, que já vem com centenas de ferramentas de segurança. Para estudar, o ideal é usar uma máquina virtual, assim você pode errar, desfazer e recomeçar sem risco para o seu computador.

Um ponto a favor do Linux é a filosofia: tudo é um arquivo, programas pequenos fazem uma coisa bem feita e se combinam entre si. Essa lógica fica evidente quando começamos a usar o terminal.

NA PRÁTICA

Opções para começar: instalar o VirtualBox ou o VMware Player, baixar a imagem oficial do Kali Linux ou do Ubuntu e criar uma máquina virtual com pelo menos 4 GB de RAM e 40 GB de disco.

LABORATÓRIO

Instale uma máquina virtual com Kali Linux ou Ubuntu. Crie um snapshot logo após a instalação para poder voltar ao estado limpo sempre que quiser.

EXERCÍCIOS

1. Qual a diferença entre kernel e distribuição?

2. Por que usar máquina virtual para estudar?

3. Cite duas distribuições e uma característica de cada.

DESAFIO PABLOCYBERSEC

Abra o terminal da máquina virtual e descubra a versão do sistema e do kernel usando os comandos cat /etc/os-release e uname -r.

Correção: *1) Kernel é o núcleo; a distribuição é o sistema completo. 2) Permite errar e restaurar sem risco. 3) Ex.: Ubuntu (estável e amigável) e Kali (ferramentas de segurança).*

CAPÍTULO 18 · NÍVEL 2 · FUNDAMENTOS

Terminal e shell

APRENDA

O terminal é a janela onde você digita comandos. O shell é o programa que interpreta esses comandos; o mais usado é o Bash. A linha de comando parece intimidadora no início, mas permite fazer em segundos o que levaria minutos com o mouse, e permite automatizar tarefas.

A estrutura de um comando é: programa, opções e argumentos. Em ls -l /etc, ls é o programa, -l é a opção e /etc é o argumento. Muitos comandos oferecem a ajuda com --help ou o manual completo com man.

O shell possui atalhos que economizam tempo: a tecla Tab completa nomes, as setas para cima e para baixo percorrem o histórico, Ctrl+C interrompe um programa e Ctrl+L limpa a tela. O redirecionamento (>, >>) grava a saída em arquivo, e o pipe (|) liga a saída de um comando à entrada de outro.

NA PRÁTICA

O pipe é o recurso mais poderoso: ls /etc | grep ssh lista os arquivos e mostra apenas os que contêm 'ssh'.

```
whoami
pwd
ls -l
man ls
ls /etc | grep ssh
echo "teste" > nota.txt
```

LABORATÓRIO

Execute cada comando acima e explique, com suas palavras, o que cada um fez. Use o history para ver tudo que digitou.

EXERCÍCIOS

1. O que é o shell?

2. Para que serve o pipe?

3. Qual a diferença entre > e >>?

DESAFIO PABLOCYBERSEC

Combine dois comandos com pipe para listar o conteúdo de /etc e contar quantos itens existem (dica: wc -l).

Correção: *1) Programa que interpreta os comandos. 2) Ligar a saída de um comando à entrada de outro. 3) > sobrescreve, >> acrescenta ao final. Desafio: ls /etc | wc -l.*

CAPÍTULO 19 · NÍVEL 2 · FUNDAMENTOS

Navegação pelo sistema

APRENDA

O Linux organiza tudo a partir de uma raiz, representada por /. Diretórios importantes: /home (pastas dos usuários), /etc (configurações), /var (logs e dados variáveis), /tmp (arquivos temporários), /bin e /usr/bin (programas), /root (pasta do administrador) e /dev (dispositivos).

Um caminho absoluto começa na raiz, como /home/ana/documentos. Um caminho relativo parte do diretório atual. O ponto (.) representa o diretório atual, dois pontos (..) o diretório pai, e o til (~) a pasta do usuário.

Para um profissional de segurança, essa organização é um mapa: os logs de autenticação ficam em /var/log, as configurações de serviços em /etc, e arquivos temporários, em /tmp, são locais comuns de rastros deixados em incidentes.

NA PRÁTICA

Arquivos que começam com ponto são ocultos, como .bashrc. Use ls -a para vê-los.

```
pwd              # onde estou
ls -la           # lista com ocultos
cd /var/log      # entra em um diretório
cd ..            # sobe um nível
cd ~             # volta para a pasta pessoal
```

LABORATÓRIO

Navegue por /etc, /var/log, /tmp e /home. Em cada local, execute ls -la e anote o que encontrou de interessante.

EXERCÍCIOS

1. O que há dentro de /etc?

2. Qual a diferença entre caminho absoluto e relativo?

3. O que representa o símbolo ~?

DESAFIO PABLOCYBERSEC

Sem usar o mouse, vá do seu diretório pessoal até /var/log e liste os cinco arquivos mais recentes (dica: ls -lt | head -5).

Correção: *1) Arquivos de configuração do sistema e serviços. 2) O absoluto parte da raiz, o relativo parte do local atual. 3) A pasta pessoal do usuário.*

CAPÍTULO 20 · NÍVEL 2 · FUNDAMENTOS

Arquivos e diretórios

APRENDA

Dominar a manipulação de arquivos é o dia a dia de qualquer pessoa na linha de comando. Comandos essenciais: mkdir cria diretórios, touch cria arquivos vazios, cp copia, mv move ou renomeia, rm remove, cat exibe conteúdo, less permite rolar textos longos, head e tail mostram o início e o fim.

Cuidado com o rm: no Linux, apagar não envia para a lixeira. O comando rm -rf remove diretórios inteiros sem perguntar. Antes de executar, confira o caminho duas vezes.

Para localizar coisas, o comando find procura arquivos por nome, tipo, tamanho ou data, e o grep procura texto dentro dos arquivos. Os dois são companheiros constantes em análises, pois permitem encontrar rapidamente a agulha no palheiro.

NA PRÁTICA

O 2>/dev/null descarta mensagens de erro, como as de 'permissão negada', deixando a saída mais limpa.

```
mkdir estudo && cd estudo
touch notas.txt
cp notas.txt copia.txt
mv copia.txt backup.txt
find /etc -name "*.conf" 2>/dev/null | head
grep -r "root" /etc/passwd
tail -n 20 /var/log/syslog
```

LABORATÓRIO

Crie um diretório de estudos com três arquivos de texto. Use grep para encontrar uma palavra em todos e find para localizar os arquivos criados nas últimas 24 horas (-mtime -1).

EXERCÍCIOS

1. Qual a diferença entre cp e mv?

2. Para que serve o comando grep?

3. Por que o rm exige cuidado?

DESAFIO PABLOCYBERSEC

Descubra quantas linhas contendo a palavra 'error' existem em um arquivo de log do seu sistema.

Correção: *1) cp copia e mv move ou renomeia. 2) Procura texto dentro de arquivos. 3) Não há lixeira; a remoção é definitiva. Desafio: grep -c error /var/log/syslog.*

CAPÍTULO 21 · NÍVEL 2 · FUNDAMENTOS

Permissões

APRENDA

Cada arquivo no Linux tem um dono, um grupo e três conjuntos de permissões: para o dono (user), para o grupo (group) e para os demais (others). Cada conjunto possui três permissões: leitura (r), escrita (w) e execução (x). A saída do ls -l mostra algo como -rwxr-xr-- e é preciso saber lê-la.

Exemplo: -rwxr-xr-- significa que o dono pode ler, escrever e executar (rwx), o grupo pode ler e executar (r-x) e os outros apenas ler (r--). O primeiro caractere indica o tipo: - para arquivo e d para diretório.

As permissões também podem ser escritas em números: r vale 4, w vale 2 e x vale 1, somados em cada conjunto. Assim, 755 é rwxr-xr-x e 644 é rw-r--r--. O comando chmod altera as permissões e o chown altera dono e grupo.

Permissões frouxas são uma falha clássica. Um arquivo de senhas legível por todos ou um script gravável por qualquer um pode ser o ponto de partida para o controle total de uma máquina.

NA PRÁTICA

Por padrão, o arquivo /etc/shadow, que guarda os hashes das senhas, só pode ser lido pelo administrador.

```
ls -l
chmod 600 segredo.txt   # só o dono lê e escreve
chmod 755 script.sh      # dono total, outros leem/executam
chown ana:ana arquivo.txt
```

LABORATÓRIO

Crie um arquivo, defina suas permissões como 600, e depois 644. Observe a mudança com ls -l. Tente ler /etc/shadow como usuário comum e anote o resultado.

EXERCÍCIOS

1. O que significa a permissão r-x?

2. Quanto vale 640 em letras?

3. Por que o /etc/shadow não deve ser legível por todos?

DESAFIO PABLOCYBERSEC

Converta para número: rwxr-x---. Depois converta 700 para letras.

Correção: *1) Ler e executar, sem escrever. 2) rw-r-----. 3) Contém hashes de senha. Desafio: 750 e rwx------.*

Boa prática: *Prefira a permissão mínima necessária. Evite chmod 777, que libera tudo para todos.*

CAPÍTULO 22 · NÍVEL 2 · FUNDAMENTOS

Usuários e grupos

APRENDA

O Linux é multiusuário. Cada pessoa ou serviço roda como um usuário com identificador próprio (UID). O usuário root, UID 0, tem poder total. Para evitar usá-lo o tempo todo, adota-se o sudo, que concede privilégios administrativos pontuais e registra quem os usou.

Os grupos reúnem usuários para facilitar a gestão de permissões. Um usuário pode pertencer a vários grupos. Informações de contas ficam em /etc/passwd (nome, UID, shell), os hashes em /etc/shadow e os grupos em /etc/group.

Do ponto de vista da segurança, contas sem uso, contas com senhas fracas e usuários com sudo excessivo são pontos de atenção. Auditar periodicamente quem tem acesso é uma tarefa simples, mas de grande valor.

NA PRÁTICA

```
id                        # UID, GID e grupos
sudo adduser lab1         # cria usuário
sudo usermod -aG sudo lab1
sudo passwd -l lab1       # bloqueia a conta
cut -d: -f1 /etc/passwd   # lista os usuários
getent group sudo
```

LABORATÓRIO

Crie um usuário de teste sem privilégios, entre nele com su - e confirme que o sudo é negado. Depois adicione-o ao grupo sudo, repita o teste e, ao final, remova o usuário.

EXERCÍCIOS

1. O que é o root?

2. Onde ficam armazenados os usuários do sistema?

3. Por que usar sudo em vez de trabalhar sempre como root?

DESAFIO PABLOCYBERSEC

Liste quais usuários do seu sistema possuem um shell de login e quais são contas de serviço.

Correção: *1) Superusuário, com poder total. 2) /etc/passwd. 3) Reduz o risco de erros e deixa registro de quem executou cada ação.*

CAPÍTULO 23 · NÍVEL 2 · FUNDAMENTOS

Processos e serviços

APRENDA

Um processo é um programa em execução, identificado por um PID. Para observar o sistema, usamos ps (lista processos), top ou htop (monitor em tempo real) e kill (encerra processos). Entender o que está rodando é essencial, pois processos desconhecidos podem indicar um problema.

Serviços, ou daemons, são processos que rodam em segundo plano, como o servidor SSH ou um banco de dados. Nas distribuições modernas, são geridos pelo systemd, por meio do comando systemctl: start, stop, restart, enable (iniciar no boot) e status.

Os logs registram o que acontece. Com journalctl ou lendo arquivos em /var/log, como auth.log, é possível ver tentativas de login, erros e alterações. Quem investiga incidentes passa boa parte do tempo lendo logs.

NA PRÁTICA

```
ps aux | head
top
systemctl status ssh
sudo systemctl start ssh
sudo systemctl enable ssh
journalctl -u ssh --since today
sudo tail -f /var/log/auth.log
```

LABORATÓRIO

Inicie o serviço SSH na sua máquina virtual, tente uma conexão local com senha errada e procure essa tentativa no auth.log. Depois pare e desative o serviço.

EXERCÍCIOS

1. O que é um PID?

2. Para que serve o systemctl?

3. Onde você veria tentativas de login falhas?

DESAFIO PABLOCYBERSEC

Identifique os três processos que mais consomem memória no seu sistema e descreva para que servem.

Correção: *1) Identificador único do processo. 2) Gerenciar serviços. 3) Em /var/log/auth.log ou via journalctl.*

CAPÍTULO 24 · NÍVEL 2 · FUNDAMENTOS

Instalação de ferramentas

APRENDA

No Linux, os programas são instalados por gerenciadores de pacotes, que baixam de repositórios oficiais, resolvem dependências e verificam assinaturas. No Debian, Ubuntu e Kali, usamos o apt. No Fedora, o dnf. Instalar de fontes oficiais é bem mais seguro do que baixar executáveis aleatórios.

Antes de instalar, atualize a lista de pacotes. De vez em quando, atualize o sistema todo: é a medida de segurança mais eficaz e barata que existe. Para ferramentas escritas em Python, usa-se o pip, preferencialmente dentro de um ambiente virtual (venv) para não bagunçar o sistema. Ferramentas disponíveis só no GitHub exigem cuidado extra: leia o código e o histórico antes de

executar.

Em um ambiente de estudos, é comum instalar nmap, wireshark, gobuster, curl, git, python3 e um editor de texto. O Kali já traz boa parte deles.

NA PRÁTICA

```
sudo apt update
sudo apt upgrade -y
sudo apt install nmap curl git python3-pip -y
apt search wireshark
dpkg -l | grep nmap
python3 -m venv meu-ambiente
source meu-ambiente/bin/activate
```

LABORATÓRIO

Atualize sua máquina virtual, instale nmap e git, confirme as versões com nmap --version e git --version, e crie um ambiente virtual Python.

EXERCÍCIOS

1. Qual a vantagem do gerenciador de pacotes?

2. Para que serve um ambiente virtual?

3. Por que manter o sistema atualizado?

DESAFIO PABLOCYBERSEC

Pesquise uma ferramenta de segurança no repositório do seu sistema, instale-a e descreva em um parágrafo para que ela serve.

Correção: *1) Fontes confiáveis, dependências e assinaturas. 2) Isolar bibliotecas do projeto. 3) Atualizações corrigem vulnerabilidades conhecidas.*

CAPÍTULO 25 · NÍVEL 2 · FUNDAMENTOS

Comandos fundamentais

APRENDA

Este capítulo reúne os comandos que você usará com mais frequência, funcionando como uma folha de referência. Pratique-os até sentir que são naturais, pois todas as partes seguintes dependem deles.

Arquivos e texto: ls, cd, pwd, cp, mv, rm, mkdir, cat, less, head, tail, grep, find, wc, sort, uniq, cut, nano. Sistema: uname, df -h (espaço em disco), free -h (memória), uptime, ps, top, kill. Rede: ip a, ping, ss, curl, wget, dig, traceroute. Pacotes: apt update, apt install. Permissões: chmod, chown, sudo, id.

A força vem das combinações. Exemplo: para contar quantas vezes cada IP aparece em um log, extraímos a coluna, ordenamos e contamos repetições. Isso é análise de logs com ferramentas simples.

NA PRÁTICA

```
# Quais IPs mais aparecem em um log
awk '{print $1}' acesso.log | sort | uniq -c | sort -nr | head

# Espaço usado e memória
df -h
free -h
```

LABORATÓRIO

Crie um arquivo com 20 linhas contendo IPs repetidos e aplique a combinação acima. Depois adapte para contar quantas linhas contêm 'Failed password' em um auth.log.

EXERCÍCIOS

1. O que faz o comando sort | uniq -c?

2. Como ver o espaço livre em disco?

3. Qual comando mostra as últimas linhas de um arquivo?

DESAFIO PABLOCYBERSEC

Monte uma cola pessoal de 15 comandos, cada um com uma frase explicando quando usar.

Correção: *1) Ordena e conta ocorrências repetidas. 2) df -h. 3) tail.*

PARTE 4

Programação para Cybersecurity

NÍVEL 3 · PRATICANTE

Lógica e Python para automatizar tarefas, analisar dados e entender como o software funciona.

26 Lógica de programação
27 Python para iniciantes
28 Variáveis, condições e loops
29 Funções
30 Manipulação de arquivos
31 Redes com Python
32 Automação para segurança

CAPÍTULO 26 · NÍVEL 3 · PRATICANTE

Lógica de programação

APRENDA

Programar é dar instruções precisas a um computador. Antes de escrever código, é preciso pensar em passos lógicos. Um algoritmo é uma sequência finita de passos para resolver um problema, como a receita de um bolo. Quem entende lógica aprende qualquer linguagem com mais facilidade.

Todo programa se apoia em poucas ideias: entrada (dados recebidos), processamento (cálculos e decisões), saída (resultado), decisão (se isto, então aquilo) e repetição (fazer várias vezes). Um fluxograma ou um pseudocódigo ajuda a planejar sem se preocupar com a sintaxe.

Na segurança, a programação serve para automatizar tarefas repetitivas, analisar grandes volumes de logs, criar ferramentas simples de verificação e entender como as falhas surgem no código dos outros.

NA PRÁTICA

Pseudocódigo para verificar uma senha: leia a senha; se tiver menos de 8 caracteres, escreva 'fraca'; senão, escreva 'aceitável'.

LABORATÓRIO

Escreva em português, passo a passo, o algoritmo para decidir se um número é par ou ímpar e outro para somar todos os números de 1 a 10.

EXERCÍCIOS

1. O que é um algoritmo?

2. Quais são os blocos básicos de um programa?

3. Para que serve o pseudocódigo?

DESAFIO PABLOCYBERSEC

Escreva o algoritmo que recebe uma lista de senhas e informa quantas têm menos de 8 caracteres.

Correção: *1) Sequência finita de passos para resolver um problema. 2) Entrada, processamento, saída, decisão e repetição. 3) Planejar a lógica sem se prender à sintaxe.*

CAPÍTULO 27 · NÍVEL 3 · PRATICANTE

Python para iniciantes

APRENDA

Python é a linguagem mais usada em segurança: é legível, tem bibliotecas para tudo e permite criar ferramentas rapidamente. Os programas são arquivos de texto com a extensão .py e executados com o comando python3.

O primeiro programa imprime um texto na tela com a função print(). Comentários começam com # e são ignorados pelo computador. A indentação (espaços no início da linha) é obrigatória e define quais linhas pertencem a um bloco; é a principal fonte de erros de quem está começando.

Ao receber um erro, leia a última linha da mensagem: ela diz o tipo do problema, e as linhas acima indicam onde ocorreu. Ler mensagens de erro com calma é uma habilidade fundamental.

NA PRÁTICA

```
# ola.py
print("Olá, PabloCyberSec!")
nome = input("Seu nome: ")
print("Bem-vindo,", nome)

# executar:
# python3 ola.py
```

LABORATÓRIO

Instale o Python 3 (já vem no Linux), crie o arquivo ola.py, execute-o e altere a mensagem. Provoque de propósito um erro de indentação e leia a mensagem resultante.

EXERCÍCIOS

1. Como executar um programa Python no terminal?

2. Para que serve o input()?

3. Por que a indentação é importante em Python?

DESAFIO PABLOCYBERSEC

Crie um programa que pergunta seu nome e sua idade e mostra em quantos anos você fará 100.

Correção: *1) python3 arquivo.py. 2) Receber um texto digitado pelo usuário. 3) Define os blocos de código.*

CAPÍTULO 28 · NÍVEL 3 · PRATICANTE

Variáveis, condições e loops

APRENDA

Uma variável guarda um valor com um nome: idade = 25. Os tipos básicos são int (inteiro), float (decimal), str (texto) e bool (verdadeiro ou falso). Listas guardam vários valores em ordem, e dicionários guardam pares chave-valor.

As condições permitem que o programa decida: if, elif e else. Os operadores de comparação são ==, !=, <, >, <= e >=, e se combinam com and, or e not.

Os loops repetem ações. O for percorre uma lista ou um intervalo, e o while repete enquanto uma condição for verdadeira. Em segurança, esses recursos servem para ler logs linha por linha, testar uma lista de itens, contar ocorrências e filtrar resultados.

NA PRÁTICA

```
portas = [22, 80, 443, 3389]
for p in portas:
    if p == 3389:
        print(p, "RDP: atenção, exposto?")
    elif p in (80, 443):
        print(p, "Web")
    else:
        print(p, "Outro serviço")

contagem = {}
for ip in ["10.0.0.1", "10.0.0.2", "10.0.0.1"]:
    contagem[ip] = contagem.get(ip, 0) + 1
print(contagem)
```

LABORATÓRIO

Execute o código acima, depois modifique-o para incluir mais portas e classificar o serviço de cada uma.

EXERCÍCIOS

1. Qual a diferença entre lista e dicionário?

2. Quando usar for e quando usar while?

3. O que faz o operador ==?

DESAFIO PABLOCYBERSEC

Escreva um programa que recebe uma senha e classifica como fraca (menos de 8 caracteres), média ou forte (com 12 ou mais).

Correção: *1) Lista guarda valores em ordem, dicionário guarda pares chave-valor. 2) for para percorrer coleções, while para repetir até uma condição mudar. 3) Compara se dois valores são iguais.*

CAPÍTULO 29 · NÍVEL 3 · PRATICANTE

Funções

APRENDA

Uma função é um bloco de código com nome, que pode ser reutilizado. Ela recebe parâmetros e pode devolver um resultado com return. Funções evitam repetição, tornam o código organizado e facilitam testar partes isoladas.

Uma boa função faz uma coisa só e tem um nome que explica o que faz. Em vez de um programa gigante, você escreve várias funções pequenas e as combina. Isso é essencial em ferramentas de segurança, em que você normalmente precisa ler, filtrar, analisar e relatar.

O Python possui muitas funções prontas (len, sum, sorted) e bibliotecas inteiras que podem ser importadas com import, como os módulos os, sys, socket, hashlib e re.

NA PRÁTICA

```
import hashlib

def hash_arquivo(caminho):
    h = hashlib.sha256()
    with open(caminho, "rb") as f:
        for bloco in iter(lambda: f.read(4096), b""):
            h.update(bloco)
    return h.hexdigest()

print(hash_arquivo("nota.txt"))
```

LABORATÓRIO

Execute o programa acima em um arquivo seu e compare o resultado com sha256sum. Eles devem coincidir.

EXERCÍCIOS

1. Qual a vantagem de usar funções?

2. Para que serve o return?

3. O que faz a instrução import?

DESAFIO PABLOCYBERSEC

Crie uma função que recebe uma senha e devolve True se tiver ao menos 12 caracteres, um número e uma letra maiúscula.

Correção: *1) Reuso, organização e facilidade de teste. 2) Devolve um resultado ao ponto em que a função foi chamada. 3) Traz um módulo ou biblioteca para o programa.*

CAPÍTULO 30 · NÍVEL 3 · PRATICANTE

Manipulação de arquivos

APRENDA

Ler e escrever arquivos é uma tarefa comum: analisar logs, processar listas, salvar resultados. Em Python, usa-se open() com o modo desejado: r para leitura, w para escrita (apaga o conteúdo anterior) e a para acrescentar. O bloco with fecha o arquivo automaticamente, o que evita erros.

Ao ler linha por linha, o programa consegue processar arquivos enormes sem carregá-los inteiros na memória. Para encontrar padrões em textos, o módulo re (expressões regulares) é muito útil, por exemplo, para extrair endereços IP ou e-mails de um log.

Formatos comuns: texto simples, CSV (tabelas) e JSON (dados estruturados). Cada um tem um módulo próprio (csv e json) na biblioteca padrão.

NA PRÁTICA

```
import re

busca = re.compile(r"\d{1,3}(?:\.\d{1,3}){3}")
encontrados = set()
with open("auth.log") as f:
    for linha in f:
        if "Failed password" in linha:
            encontrados.update(busca.findall(linha))

with open("ips_suspeitos.txt", "w") as saida:
    for ip in sorted(encontrados):
        saida.write(ip + "\n")
print(len(encontrados), "IPs")
```

LABORATÓRIO

Gere um arquivo de log de testes com algumas linhas contendo 'Failed password' e IPs. Execute o programa acima e confira o arquivo de saída.

EXERCÍCIOS

1. Qual a diferença entre os modos w e a?

2. Por que usar with ao abrir arquivos?

3. Para que servem as expressões regulares?

DESAFIO PABLOCYBERSEC

Adapte o programa para contar quantas vezes cada IP aparece e mostrar os cinco que mais aparecem.

Correção: *1) w sobrescreve, a acrescenta. 2) Garante o fechamento do arquivo. 3) Encontrar padrões de texto.*

CAPÍTULO 31 · NÍVEL 3 · PRATICANTE

Redes com Python

APRENDA

O módulo socket permite criar conexões de rede em poucas linhas. Com ele, é possível construir clientes e servidores simples e entender como a comunicação TCP funciona por baixo dos panos. Um teste de porta, por exemplo, é só tentar abrir uma conexão e ver se foi aceita.

Esse exercício é didático: ele ensina o mecanismo que as ferramentas profissionais, como o Nmap, usam com muito mais sofisticação. Aplique-o apenas em máquinas suas ou de laboratório autorizado. Varrer sistemas de terceiros sem permissão é ilegal.

A biblioteca requests simplifica o trabalho com HTTP: você pode consultar páginas, ler cabeçalhos e verificar status. É a base de muitos scripts de verificação de aplicações web.

NA PRÁTICA

```
import socket

def porta_aberta(host, porta, timeout=1):
    s = socket.socket()
    s.settimeout(timeout)
    try:
        s.connect((host, porta))
        return True
    except OSError:
        return False
    finally:
        s.close()

HOST = "127.0.0.1"   # seu próprio computador
for p in (22, 80, 443, 8080):
    print(p, "aberta" if porta_aberta(HOST, p) else "fechada")
```

LABORATÓRIO

Rode o programa em 127.0.0.1. Em outro terminal, inicie um servidor web com python3 -m http.server 8080 e execute novamente: a porta 8080 deve aparecer como aberta.

EXERCÍCIOS

1. Que módulo do Python permite criar conexões de rede?

2. Por que só devemos testar máquinas autorizadas?

3. O que acontece quando a conexão é recusada?

DESAFIO PABLOCYBERSEC

Melhore o programa para receber o host e uma faixa de portas pela linha de comando (módulo sys.argv).

Correção: *1) socket. 2) Testar sem permissão pode ser crime. 3) A conexão falha e a porta é considerada fechada ou filtrada.*

CAPÍTULO 32 · NÍVEL 3 · PRATICANTE

Automação para segurança

APRENDA

Automação é fazer o computador repetir, sem cansaço, o que seria entediante e sujeito a erros. Na segurança ela é usada para verificar integridade de arquivos, analisar logs, conferir configurações, gerar relatórios e organizar resultados de ferramentas.

Um fluxo típico: executar uma ferramenta, capturar sua saída, filtrar o que interessa e salvar em um formato legível. O módulo subprocess permite chamar comandos do sistema a partir do Python, e o cron agenda tarefas periódicas no Linux.

Antes de automatizar, tenha certeza de que sabe fazer à mão e entenda cada passo. Um script que roda sem supervisão pode causar danos em larga escala se estiver errado. Teste sempre em ambiente de laboratório e registre o que o script faz.

NA PRÁTICA

Para agendar, edite o cron com crontab -e. Por exemplo: 0 2 * * * /usr/bin/python3 /home/lab/verifica.py (executa todo dia às 2h).

```
import subprocess, datetime

alvo = "192.168.56.101"   # máquina do laboratório
res = subprocess.run(["nmap", "-F", alvo], capture_output=True, text=Tru
e)
nome = f"scan_{datetime.date.today()}.txt"
with open(nome, "w") as f:
    f.write(res.stdout)
print("Resultado salvo em", nome)
```

LABORATÓRIO

Escreva um script que calcula o hash de todos os arquivos de uma pasta, salva em um arquivo e, em uma segunda execução, avisa quais mudaram.

EXERCÍCIOS

1. O que o módulo subprocess permite fazer?

2. Para que serve o cron?

3. Por que testar um script em laboratório antes de usá-lo?

DESAFIO PABLOCYBERSEC

Monte um pequeno monitor de integridade: dada uma pasta, ele registra os hashes e alerta sobre arquivos novos, removidos ou alterados.

Correção: *1) Executar comandos do sistema a partir do Python. 2) Agendar tarefas. 3) Um erro automatizado pode causar danos em escala.*

PARTE 5

Hacking Ético

NÍVEL 4 · PENTESTER JÚNIOR

Metodologia de pentest do planejamento ao relatório, sempre com autorização.

33 O que é Ethical Hacking
34 Metodologia de Pentest
35 Reconhecimento
36 Enumeração
37 Scanning
38 Identificação de serviços
39 Vulnerabilidades
40 Exploração em laboratório
41 Pós-exploração e documentação
42 Relatório de Pentest

CAPÍTULO 33 · NÍVEL 4 · PENTESTER JÚNIOR

O que é Ethical Hacking

APRENDA

Hacking ético é a prática de usar técnicas de ataque, com autorização, para encontrar falhas e corrigi-las antes que criminosos as explorem. O termo hacker, em sua origem, designa alguém curioso e habilidoso com tecnologia; o que define o caráter é o que se faz com esse conhecimento.

Por convenção, fala-se em chapéus: white hat (age com autorização, para proteger), black hat (age por ganho ilegal ou dano) e grey hat (age sem permissão, mas sem má intenção aparente, o que continua sendo ilegal). O profissional que você quer ser é o white hat.

Existem formas legais de praticar: laboratórios próprios, plataformas de treinamento como Hack The Box, TryHackMe e máquinas intencionalmente vulneráveis (Metasploitable, DVWA, OWASP Juice Shop), além de programas de bug bounty, em que empresas convidam pesquisadores a reportar falhas dentro de regras definidas.

NA PRÁTICA

Código de conduta do profissional: ter autorização escrita, respeitar o escopo, minimizar impactos, proteger os dados encontrados, comunicar as falhas de forma responsável e documentar tudo.

LABORATÓRIO

Crie contas em uma plataforma de treinamento legal (por exemplo, TryHackMe) e complete uma sala introdutória. Anote o que aprendeu.

EXERCÍCIOS

1. O que diferencia um hacker ético de um criminoso?

2. O que significa grey hat?

3. Cite duas formas legais de praticar.

DESAFIO PABLOCYBERSEC

Escreva seu próprio código de conduta com cinco regras que você se compromete a seguir.

Correção: *1) A autorização e a intenção de proteger. 2) Atua sem permissão, sem intenção maliciosa, mas ainda assim ilegalmente. 3) Laboratório próprio, plataformas de treino, bug bounty.*

CAPÍTULO 34 · NÍVEL 4 · PENTESTER JÚNIOR

Metodologia de Pentest

APRENDA

Pentest (teste de intrusão) é uma avaliação de segurança que simula um atacante real, de forma controlada e autorizada, para identificar vulnerabilidades e medir seu impacto. Não é uma varredura automática: exige análise humana, criatividade e método.

Um processo comum tem estas fases: 1) Planejamento e escopo; 2) Reconhecimento; 3) Enumeração e scanning; 4) Análise de vulnerabilidades; 5) Exploração controlada; 6) Pós-exploração; 7) Relatório e recomendações; 8) Reteste após correções. Padrões reconhecidos incluem PTES, OSSTMM, NIST SP 800-115 e, para web, o guia de testes da OWASP.

Quanto ao nível de informação, o teste pode ser black box (sem informações prévias), grey box (informações parciais, como uma conta de usuário) ou white box (acesso total a documentação e código). Cada um responde a perguntas diferentes e tem custos diferentes.

NA PRÁTICA

Um bom pentest termina com ações claras: o cliente deve saber o que está errado, qual o risco, como corrigir e em que ordem.

LABORATÓRIO

Desenhe um fluxograma das fases do pentest e, ao lado de cada fase, escreva um exemplo de atividade e o produto gerado (por exemplo, lista de hosts, lista de serviços, evidências).

EXERCÍCIOS

1. Qual a diferença entre pentest e varredura automática de vulnerabilidades?

2. O que é um teste black box?

3. Por que há uma fase de reteste?

DESAFIO PABLOCYBERSEC

Pesquise o padrão PTES e compare suas fases com as listadas neste capítulo.

Correção: *1) O pentest inclui análise humana e exploração controlada. 2) O testador não recebe informações prévias. 3) Para confirmar que as correções funcionaram.*

CAPÍTULO 35 · NÍVEL 4 · PENTESTER JÚNIOR

Reconhecimento

APRENDA

Reconhecimento (recon) é a fase de coletar informações sobre o alvo antes de qualquer interação mais intrusiva. Quanto melhor o mapeamento, mais eficiente e menos ruidoso será o restante do teste. Divide-se em passivo (sem tocar no alvo, usando fontes públicas) e ativo (interagindo diretamente com ele).

No recon passivo, consulta-se o WHOIS (registro do domínio), registros DNS, certificados públicos, páginas da empresa, perfis profissionais, repositórios de código e mecanismos de busca. A técnica de usar buscas avançadas para achar informações expostas chama-se Google Dorking. O conjunto de técnicas que usa fontes abertas é conhecido como OSINT.

No recon ativo, consultam-se servidores DNS, descobrem-se subdomínios e começa-se a observar o comportamento do alvo. Tudo precisa estar dentro do escopo autorizado.

NA PRÁTICA

```
whois exemplo.com
dig exemplo.com ANY +short
dig exemplo.com MX +short
nslookup -type=ns exemplo.com
site:exemplo.com filetype:pdf   # pesquisa avançada no buscador
```

LABORATÓRIO

Escolha um domínio seu (ou o de um programa de bug bounty que autorize essa atividade) e levante: servidores DNS, servidores de e-mail, tecnologias visíveis no site e perfis públicos relacionados. Organize em uma tabela.

EXERCÍCIOS

1. Qual a diferença entre recon passivo e ativo?

2. O que é OSINT?

3. Que informações o WHOIS fornece?

DESAFIO PABLOCYBERSEC

Liste cinco tipos de informação pública sobre uma empresa que poderiam ajudar um atacante e, para cada uma, como a empresa poderia reduzir a exposição.

Correção: *1) O passivo não interage com o alvo, o ativo interage. 2) Inteligência de fontes abertas. 3) Titularidade, datas e servidores de nomes do domínio.*

CAPÍTULO 36 · NÍVEL 4 · PENTESTER JÚNIOR

Enumeração

APRENDA

Enumeração é a etapa de extrair informações detalhadas dos serviços encontrados: nomes de usuários válidos, compartilhamentos de rede, versões de software, estruturas de diretórios, banners. É a ponte entre saber que algo existe e entender como ele funciona.

Cada serviço oferece pistas diferentes. Em um servidor web, enumeram-se diretórios e arquivos. Em compartilhamentos de arquivos (SMB), listam-se pastas acessíveis. Em DNS, verificam-se subdomínios. Em SSH ou FTP, observam-se os banners de versão. Muitas falhas estão em configurações padrão ou em informações que o administrador não percebeu que estavam expostas.

Anote tudo de forma organizada. Uma boa enumeração rende mais do que uma exploração apressada, e as anotações viram a base do relatório.

NA PRÁTICA

Dica: o banner de um serviço, como 'Apache 2.4.x', informa a versão. Com a versão é possível verificar se há vulnerabilidades públicas conhecidas.

```
# Exemplos em um ambiente de laboratório autorizado
nmap -sV -p 21,22,80 192.168.56.101
smbclient -L //192.168.56.101 -N
curl -I http://192.168.56.101/
```

LABORATÓRIO

No seu laboratório, enumere os serviços de uma máquina vulnerável de treino e monte uma tabela: porta, serviço, versão e observações.

EXERCÍCIOS

1. O que se busca na enumeração?

2. O que é um banner de serviço?

3. Por que documentar durante o processo?

DESAFIO PABLOCYBERSEC

Para um servidor com as portas 21, 22, 80 e 445 abertas, liste o que você tentaria enumerar em cada uma.

Correção: *1) Detalhes dos serviços, como usuários, versões e recursos acessíveis. 2) Texto com a identificação do serviço e da versão. 3) Para reconstruir o raciocínio e gerar o relatório.*

CAPÍTULO 37 · NÍVEL 4 · PENTESTER JÚNIOR

Scanning

APRENDA

Scanning é a varredura ativa de hosts e portas para descobrir o que está no ar e quais serviços respondem. Em geral, começa-se descobrindo hosts ativos na rede, depois se varrem suas portas e, por fim, identificam-se serviços e versões.

Tipos comuns de varredura de portas: TCP connect (completa o handshake, é confiável e deixa rastros), SYN scan (envia SYN e analisa a resposta sem completar a conexão, mais discreto) e UDP scan (mais lento, pois o UDP não responde de forma padronizada). Os estados observados são aberta, fechada e filtrada.

A varredura é detectável por sistemas de defesa. Em testes reais, ajusta-se a velocidade e os horários conforme o combinado no escopo, para não derrubar serviços frágeis.

NA PRÁTICA

```
# Sempre dentro do laboratório autorizado
nmap -sn 192.168.56.0/24           # descobrir hosts
nmap 192.168.56.101                # 1000 portas mais comuns
nmap -p- 192.168.56.101            # todas as portas TCP
nmap -sU --top-ports 20 192.168.56.101
```

LABORATÓRIO

Faça um scan de descoberta e depois uma varredura completa de uma máquina vulnerável de treino. Abra o Wireshark e observe os pacotes SYN, SYN-ACK e RST gerados.

EXERCÍCIOS

1. O que significa porta filtrada?

2. Qual a diferença entre TCP connect e SYN scan?

3. Por que o UDP scan costuma ser mais lento?

DESAFIO PABLOCYBERSEC

Compare o resultado de nmap sem opções com nmap -p-. Descreva o que cada um revelou.

Correção: *1) Um firewall bloqueia ou descarta a resposta. 2) O connect completa o handshake, o SYN não. 3) Muitas portas UDP não respondem, exigindo esperas.*

CAPÍTULO 38 · NÍVEL 4 · PENTESTER JÚNIOR

Identificação de serviços

APRENDA

Saber que a porta 80 está aberta ajuda pouco. É preciso descobrir qual programa e qual versão respondem ali. A identificação de serviços (service fingerprinting) envia sondas ao serviço e compara as respostas a uma base de assinaturas.

A versão é decisiva: uma versão desatualizada pode ter falhas conhecidas e catalogadas. As vulnerabilidades públicas recebem identificadores CVE (por exemplo, CVE-2021-44228) e uma pontuação de severidade CVSS, de 0 a 10. Bancos públicos como o NVD e o Exploit-DB ajudam a cruzar as informações.

Tome cuidado: um banner pode estar falso ou omitido, e uma versão que aparenta estar vulnerável pode ter recebido correção retroativa pelo fabricante da distribuição. Confirme antes de concluir.

NA PRÁTICA

Interpretação rápida do CVSS: 0,1–3,9 baixa; 4,0–6,9 média; 7,0–8,9 alta; 9,0–10,0 crítica.

```
nmap -sV 192.168.56.101
nmap -sV --version-intensity 5 -p 22,80 192.168.56.101
nmap -sC -sV -p 80 192.168.56.101   # scripts padrão
```

LABORATÓRIO

Execute nmap -sV no seu laboratório, escolha um serviço e pesquise a versão no NVD. Anote os CVEs encontrados e a severidade de cada um.

EXERCÍCIOS

1. O que é um CVE?

2. Para que serve o CVSS?

3. Por que um banner pode enganar?

DESAFIO PABLOCYBERSEC

Explique por que a identificação de versão é mais útil do que apenas conhecer o número da porta.

Correção: *1) Identificador público de uma vulnerabilidade. 2) Medir a severidade. 3) Pode ser alterado, omitido ou ter correção retroativa.*

CAPÍTULO 39 · NÍVEL 4 · PENTESTER JÚNIOR

Vulnerabilidades

APRENDA

Uma vulnerabilidade é uma falha que pode comprometer a CIA. Podem estar no software (erros de programação), na configuração (senhas padrão, serviços desnecessários), na arquitetura (ausência de segmentação) ou nas pessoas e processos.

Os scanners de vulnerabilidades, como Nessus, OpenVAS e Nikto, automatizam a verificação de problemas conhecidos. São úteis para ganhar escala, mas geram falsos positivos (alertas que não são reais) e falsos negativos (falhas que passam despercebidas). O papel do analista é validar cada achado.

Para priorizar, considere a severidade técnica, a facilidade de exploração, o valor do ativo afetado e se a falha está exposta à internet. Uma vulnerabilidade

média em um sistema crítico exposto pode ser mais urgente do que uma crítica em um sistema isolado.

NA PRÁTICA

Classificação útil para o relatório: crítica, alta, média, baixa e informativa, cada uma com evidência, impacto e recomendação.

```
# Exemplo em laboratório
nikto -h http://192.168.56.101
nmap --script vuln 192.168.56.101
```

LABORATÓRIO

Rode um scanner de vulnerabilidades no seu laboratório. Escolha três achados, tente confirmá-los manualmente e marque quais eram falsos positivos.

EXERCÍCIOS

1. O que é um falso positivo?

2. Cite duas categorias de vulnerabilidade.

3. O que considerar para priorizar uma correção?

DESAFIO PABLOCYBERSEC

Monte uma tabela com três vulnerabilidades fictícias e priorize-as, justificando a ordem.

Correção: *1) Alerta de falha que na verdade não existe. 2) Software e configuração, por exemplo. 3) Severidade, facilidade, valor do ativo e exposição.*

CAPÍTULO 40 · NÍVEL 4 · PENTESTER JÚNIOR

Exploração em laboratório

APRENDA

Explorar é usar uma vulnerabilidade confirmada para demonstrar o impacto real, como obter acesso, ler dados ou executar comandos. O objetivo profissional não é causar dano, mas provar que o risco existe, com o mínimo de interferência possível.

Antes de explorar, confirme a autorização e o escopo, avalie o risco de derrubar o serviço, tenha backup ou snapshot, e escolha a abordagem menos invasiva que comprove o ponto. Em ambientes de produção, combine janelas de teste com o cliente.

No laboratório, o processo educativo é: identificar o serviço e a versão, pesquisar a falha documentada, entender por que ela funciona (e não apenas executar um módulo), testar na máquina de treino, observar o resultado e registrar. Entender o mecanismo é o que diferencia o profissional de quem apenas aperta botões.

Este livro não fornece códigos de exploração prontos. O foco é o raciocínio, a metodologia e o uso responsável das ferramentas em laboratórios de treino.

NA PRÁTICA

Fluxo de raciocínio: o que está exposto? Qual é a falha? Como ela funciona? Qual o impacto? Como corrigir? Qual evidência comprova?

LABORATÓRIO

Em uma máquina vulnerável de treino (por exemplo, a Metasploitable em rede isolada), escolha uma falha documentada, descreva por escrito como ela funciona antes de testá-la e, depois, registre o resultado com capturas de tela.

EXERCÍCIOS

1. Qual o objetivo da exploração em um pentest profissional?

2. Que cuidados tomar antes de explorar?

3. Por que entender a falha é melhor que apenas executar uma ferramenta?

DESAFIO PABLOCYBERSEC

Escreva um parágrafo explicando uma vulnerabilidade antiga e famosa, indicando causa, impacto e como foi corrigida.

Correção: *1) Comprovar o risco com o mínimo de interferência. 2) Autorização, escopo, backup e avaliação de impacto. 3) Permite adaptar, diagnosticar e recomendar correções.*

CAPÍTULO 41 · NÍVEL 4 · PENTESTER JÚNIOR

Pós-exploração e documentação

APRENDA

Depois de comprovar um acesso, a pós-exploração avalia o que um atacante conseguiria fazer a partir dali: que dados estão acessíveis, que outros sistemas se alcançam, quais privilégios podem ser obtidos. Em um teste profissional, ela tem limites claros definidos no escopo, e o objetivo é medir o impacto, sem extrair dados reais além do estritamente necessário.

Nesse ponto surgem questões éticas: se você encontrar informações pessoais ou sigilosas, registre apenas o necessário como evidência, mascare dados sensíveis e comunique ao contato indicado. Ao final, todo o acesso criado durante o teste, como contas e arquivos, deve ser removido e informado ao cliente.

Documentar é tão importante quanto testar. Registre horário, comando ou ação, resultado e captura de tela. Sem evidências, o cliente não consegue reproduzir nem corrigir o problema, e você não consegue defender suas conclusões.

NA PRÁTICA

Modelo de registro: data/hora; sistema; ação realizada; resultado observado; evidência (arquivo ou captura); impacto; observações.

LABORATÓRIO

Crie um diário de teste em Markdown ou planilha com as colunas acima e use-o durante todo o laboratório do capítulo anterior.

EXERCÍCIOS

1. O que se avalia na pós-exploração?

2. O que fazer ao encontrar dados pessoais durante o teste?

3. Por que registrar horários e evidências?

DESAFIO PABLOCYBERSEC

Escreva um checklist de limpeza para o final de um teste: o que deve ser removido ou revertido?

Correção: *1) O impacto potencial a partir do acesso obtido. 2) Registrar o mínimo, mascarar e comunicar. 3) Para reprodutibilidade, correção e defesa das conclusões.*

CAPÍTULO 42 · NÍVEL 4 · PENTESTER JÚNIOR

Relatório de Pentest

APRENDA

O relatório é o produto final e a prova do valor do seu trabalho. Pode ter o melhor teste do mundo, mas, se o cliente não entender, nada será corrigido. Escreva para dois públicos: a direção (linguagem de negócio e riscos) e a equipe técnica (detalhes para reproduzir e corrigir).

Estrutura recomendada: 1) Capa e controle de versão; 2) Sumário executivo; 3) Escopo e metodologia; 4) Resumo dos achados por severidade; 5) Detalhamento de cada achado; 6) Recomendações priorizadas; 7) Conclusão; 8) Anexos (evidências e logs).

Cada achado deve trazer: título claro, severidade, descrição, onde ocorre, impacto, passos para reproduzir, evidência, recomendação de correção e referências (CVE, CWE, OWASP). Seja objetivo, imparcial e sem julgamento sobre pessoas. O tom é de parceria, e não de acusação.

NA PRÁTICA

Exemplo de título: 'Painel administrativo acessível sem autenticação'. Severidade: alta. Impacto: um visitante pode alterar dados dos clientes. Recomendação: exigir login e restringir o acesso por rede.

LABORATÓRIO

Pegue os achados do seu laboratório e escreva um relatório completo de 3 a 5 páginas, seguindo a estrutura acima. Peça a um colega que o leia e diga se entendeu.

EXERCÍCIOS

1. Quais os dois públicos de um relatório?

2. O que deve conter um achado?

3. Por que a linguagem deve ser objetiva e imparcial?

DESAFIO PABLOCYBERSEC

Escreva o sumário executivo (máximo de 150 palavras) de um pentest fictício que encontrou duas falhas altas e três médias.

Correção: *1) Direção e equipe técnica. 2) Título, severidade, descrição, impacto, reprodução, evidência e correção. 3) O foco é resolver o problema, não apontar culpados.*

PARTE 6

Web Security

NÍVEL 4 · PENTESTER JÚNIOR

Como as aplicações web funcionam e onde falham: OWASP Top 10, injeções, sessões, acesso e APIs.

43 Como funciona uma aplicação Web
44 HTTP na prática
45 Cookies e sessões
46 Autenticação
47 OWASP Top 10
48 SQL Injection
49 XSS (Cross-Site Scripting)
50 Controle de acesso
51 Vulnerabilidades de API

CAPÍTULO 43 · NÍVEL 4 · PENTESTER JÚNIOR

Como funciona uma aplicação Web

APRENDA

Uma aplicação web tem, em geral, três partes. O front-end é o que roda no navegador (HTML, CSS, JavaScript). O back-end é o código no servidor que processa requisições (PHP, Python, Java, Node.js). O banco de dados guarda as informações (MySQL, PostgreSQL, MongoDB). Entre elas, há APIs, servidores web e, muitas vezes, balanceadores e firewalls de aplicação (WAF).

O fluxo básico: o usuário acessa uma URL, o navegador envia a requisição, o servidor a processa, consulta o banco e devolve uma resposta. Toda informação que vem do navegador (formulários, URL, cabeçalhos, cookies) pode ser manipulada pelo usuário, por isso nunca deve ser considerada confiável.

Esse é o princípio central da segurança web: nunca confie na entrada do usuário. Quase todas as falhas que veremos nos próximos capítulos nascem de dados mal validados ou de verificações feitas apenas no lado do cliente.

NA PRÁTICA

Anatomia de uma URL: https://loja.exemplo.com:443/produtos?id=10#topo. Esquema (https), host (loja.exemplo.com), porta (443), caminho (/produtos), parâmetros (id=10) e fragmento (#topo).

LABORATÓRIO

Instale o OWASP Juice Shop ou o DVWA em uma máquina virtual isolada e navegue pela aplicação. Identifique front-end, formulários, parâmetros de URL e pontos de entrada de dados.

EXERCÍCIOS

1. Quais são as três partes de uma aplicação web?

2. Por que a entrada do usuário nunca é confiável?

3. O que é um parâmetro de URL?

DESAFIO PABLOCYBERSEC

Liste, em uma página de login, todos os pontos em que o usuário envia dados ao servidor.

Correção: *1) Front-end, back-end e banco de dados. 2) Porque pode ser alterada pelo usuário. 3) Dado enviado na URL, após o ponto de interrogação.*

CAPÍTULO 44 · NÍVEL 4 · PENTESTER JÚNIOR

HTTP na prática

APRENDA

Com as ferramentas certas, você pode observar e alterar o que o navegador envia. As Ferramentas de Desenvolvedor (tecla F12) mostram requisições, cabeçalhos, cookies, código-fonte e erros. O curl permite montar requisições manualmente, e os proxies de interceptação, como o Burp Suite, que veremos adiante, deixam você pausar e editar cada requisição antes de enviar.

Pontos a observar: métodos HTTP, parâmetros enviados por GET e POST, cabeçalhos de segurança (Content-Security-Policy, X-Frame-Options, Strict-Transport-Security), códigos de status e mensagens de erro. Mensagens de erro detalhadas demais podem revelar caminhos, versões e estruturas internas.

Também vale verificar os arquivos comuns, como robots.txt, que às vezes lista diretórios que o administrador preferia esconder, mas que acabam revelados por isso mesmo.

NA PRÁTICA

```
curl -i http://192.168.56.102/
curl -X POST -d "user=teste&senha=123" http://192.168.56.102/login
curl -s http://192.168.56.102/robots.txt
```

LABORATÓRIO

Em uma aplicação de treino, abra a aba Rede (F12), faça login e examine a requisição: método, URL, corpo, cabeçalhos e a resposta. Depois reproduza a mesma requisição com o curl.

EXERCÍCIOS

1. Qual a utilidade da aba Rede nas ferramentas de desenvolvedor?

2. Por que mensagens de erro detalhadas são arriscadas?

3. O que é o robots.txt?

DESAFIO PABLOCYBERSEC

Verifique quais cabeçalhos de segurança estão ausentes em uma aplicação de treino e explique a função de cada um.

Correção: *1) Observar requisições e respostas. 2) Revelam detalhes internos úteis a um atacante. 3) Arquivo que orienta mecanismos de busca sobre o que indexar.*

CAPÍTULO 45 · NÍVEL 4 · PENTESTER JÚNIOR

Cookies e sessões

APRENDA

O HTTP não guarda memória entre requisições. Para manter você logado, o servidor cria uma sessão e entrega um identificador de sessão, geralmente em um cookie, que o navegador devolve a cada requisição. Quem possui esse identificador é tratado como o usuário legítimo, e por isso ele precisa ser protegido.

Atributos de segurança dos cookies: HttpOnly (impede acesso por JavaScript), Secure (só trafega por HTTPS) e SameSite (limita o envio em requisições entre sites, reduzindo ataques de CSRF). Outras boas práticas: identificadores longos e aleatórios, expiração, renovação do ID após o login e encerramento completo da sessão no logout.

Problemas típicos: sessões previsíveis, tokens expostos na URL, sessões que não expiram e ausência de HTTPS. Sequestro de sessão (session hijacking) é a apropriação de um identificador válido de outra pessoa.

NA PRÁTICA

O cookie é uma informação guardada no navegador. Já a sessão é o estado mantido no servidor. O cookie costuma carregar apenas a chave que aponta para a sessão.

LABORATÓRIO

No navegador, abra F12, aba Aplicativo (ou Armazenamento) e observe os cookies de uma aplicação de treino: nome, valor, domínio e atributos. Verifique se HttpOnly, Secure e SameSite estão ativos.

EXERCÍCIOS

1. Para que serve um cookie de sessão?

2. O que faz o atributo HttpOnly?

3. O que é sequestro de sessão?

DESAFIO PABLOCYBERSEC

Descreva três medidas para reduzir o risco de roubo de sessão em uma aplicação.

Correção: *1) Identificar o usuário entre requisições. 2) Impede que scripts leiam o cookie. 3) Uso indevido de um identificador de sessão válido de outra pessoa.*

CAPÍTULO 46 · NÍVEL 4 · PENTESTER JÚNIOR

Autenticação

APRENDA

A autenticação web é um dos pontos mais visados. Falhas comuns: senhas fracas permitidas, ausência de limite de tentativas (permite força bruta), mensagens que revelam se o usuário existe ('usuário não encontrado' versus 'senha incorreta'), recuperação de senha insegura e armazenamento de senhas sem hash adequado.

Como defender: exigir senhas longas, armazenar apenas hashes com algoritmos próprios (bcrypt, scrypt, Argon2) com sal, limitar e monitorar tentativas, usar autenticação multifator, responder com mensagens genéricas e proteger o fluxo de redefinição com tokens aleatórios, de uso único e curta validade.

O credential stuffing é o uso de pares de usuário e senha vazados em outros sites, aproveitando a reutilização de senhas. Bloquear essa prática exige MFA, detecção de comportamento anômalo e verificação de senhas já vazadas.

NA PRÁTICA

Ao testar, em um laboratório, observe: o que acontece com 5, 10 e 50 tentativas erradas? A mensagem de erro é igual para usuário inexistente e senha errada?

LABORATÓRIO

Em uma aplicação de treino, analise a tela de login e a de recuperação de senha. Liste as falhas encontradas e proponha correções para cada uma.

EXERCÍCIOS

1. Por que a mensagem de erro deve ser genérica?

2. O que é credential stuffing?

3. Por que armazenar senhas com hash e sal?

DESAFIO PABLOCYBERSEC

Escreva uma política de senhas para uma empresa, com cinco regras objetivas.

Correção: *1) Para não revelar quais usuários existem. 2) Uso de credenciais vazadas de outros sites. 3) Evita exposição das senhas e dificulta a quebra em massa.*

CAPÍTULO 47 · NÍVEL 4 · PENTESTER JÚNIOR

OWASP Top 10

APRENDA

A OWASP é uma comunidade que publica materiais gratuitos de segurança. Seu Top 10 lista as categorias de risco mais críticas em aplicações web e é a referência mais citada do mercado. A lista é atualizada periodicamente; consulte a versão mais recente no site oficial.

Entre as categorias que aparecem de forma recorrente nas edições estão: controle de acesso quebrado, falhas criptográficas, injeção (SQL, comandos, entre outras), design inseguro, configuração incorreta de segurança, componentes vulneráveis e desatualizados, falhas de identificação e autenticação, falhas de integridade de software e dados, falhas de registro e monitoramento, e SSRF (requisições forjadas pelo servidor).

Para o estudante, o Top 10 é um roteiro: cada categoria deve ser entendida em três níveis: o que é, como se manifesta e como se previne. Os próximos capítulos aprofundam as mais importantes.

NA PRÁTICA

Recursos gratuitos da OWASP: Top 10, Cheat Sheet Series (guias práticos de defesa), Testing Guide (metodologia de testes) e projetos de treino como o Juice Shop.

LABORATÓRIO

Acesse o site oficial da OWASP, baixe o Top 10 mais recente e faça um resumo de uma página, com uma frase explicando cada categoria com suas palavras.

EXERCÍCIOS

1. O que é o OWASP Top 10?

2. Cite três categorias que aparecem nele.

3. Por que é preciso consultar a versão mais recente?

DESAFIO PABLOCYBERSEC

Escolha uma categoria e encontre, no Juice Shop, um desafio relacionado a ela.

Correção: *1) Lista das categorias de risco mais críticas em aplicações web. 2) Ex.: controle de acesso quebrado, injeção, configuração incorreta. 3) Porque ela evolui com as ameaças.*

CAPÍTULO 48 · NÍVEL 4 · PENTESTER JÚNIOR

SQL Injection

APRENDA

Muitas aplicações guardam dados em bancos relacionais e conversam com eles usando SQL. A injeção de SQL ocorre quando a aplicação monta a consulta concatenando texto vindo do usuário, sem tratamento. O usuário passa a alterar o significado do comando que o banco executa, e não apenas os dados.

Imagine uma consulta de login: SELECT * FROM usuarios WHERE nome='[entrada]' AND senha='[entrada]'. Se a aplicação insere a entrada diretamente, uma entrada com aspas e operadores lógicos pode mudar a condição para sempre verdadeira, contornando o login. Em casos mais graves, é possível ler dados de outras tabelas ou modificá-los. É uma das falhas mais antigas e ainda hoje presentes.

A defesa é conhecida e eficaz: consultas parametrizadas (prepared statements), em que o código da consulta e os dados viajam separados; validação de entradas; privilégios mínimos para a conta do banco; mensagens de erro genéricas; e uso de ORMs com cuidado. Um WAF ajuda, mas não substitui o código correto.

NA PRÁTICA

```
# Exemplo SEGURO em Python (consulta parametrizada)
cur.execute(
    "SELECT * FROM usuarios WHERE nome = %s AND senha_hash = %s",
    (nome, hash_senha)
)
```

LABORATÓRIO

No DVWA, com o nível de segurança baixo e em ambiente isolado, observe como uma aspa simples no campo de ID altera o comportamento da página. Depois mude para o nível alto e compare o código-fonte que o DVWA exibe.

EXERCÍCIOS

1. O que causa uma injeção de SQL?

2. Como funcionam as consultas parametrizadas?

3. Por que privilégios mínimos no banco reduzem o impacto?

DESAFIO PABLOCYBERSEC

Reescreva em pseudocódigo uma função de login de forma segura, usando parâmetros e hash de senha.

Correção: *1) Concatenar entrada do usuário à consulta sem tratamento. 2) Separam o código da consulta dos dados. 3) Limita o que o invasor consegue ler ou alterar.*

CAPÍTULO 49 · NÍVEL 4 · PENTESTER JÚNIOR

XSS (Cross-Site Scripting)

APRENDA

O XSS acontece quando uma aplicação exibe, em uma página, conteúdo enviado por usuários sem o tratamento adequado, permitindo que um script seja executado no navegador de outra pessoa. Como o script roda no contexto do site legítimo, pode ler informações da página, agir em nome do usuário ou redirecioná-lo.

Existem três tipos: refletido (a entrada volta imediatamente na resposta, geralmente por meio de um link), armazenado (o conteúdo fica salvo, como em um comentário, e atinge quem o visualizar) e baseado em DOM (a falha está no JavaScript do próprio navegador). O armazenado costuma ser o mais perigoso.

Defesas: codificar a saída conforme o contexto (HTML, atributo, JavaScript, URL), validar entradas, usar frameworks que escapam por padrão, cookies HttpOnly e uma política de segurança de conteúdo (CSP) que restrinja a origem dos scripts.

NA PRÁTICA

Teste inofensivo em laboratório: inserir um texto como teste em um campo de comentário. Se a página mostrar o texto em negrito, e não o código literal, a saída não está sendo tratada.

LABORATÓRIO

No DVWA ou no Juice Shop, em ambiente isolado, localize um campo que reflita sua entrada. Verifique se tags HTML são interpretadas e anote como o código corrigido deveria tratar a saída.

EXERCÍCIOS

1. Qual a diferença entre XSS refletido e armazenado?

2. O que é a codificação de saída?

3. Qual o papel da CSP?

DESAFIO PABLOCYBERSEC

Explique por que um XSS em uma página de administração tende a ser mais grave que em uma página pública.

Correção: *1) O refletido volta na resposta imediata, o armazenado fica salvo e atinge vários usuários. 2) Converter caracteres especiais para que sejam exibidos como texto. 3) Restringir de onde os scripts podem ser carregados.*

CAPÍTULO 50 · NÍVEL 4 · PENTESTER JÚNIOR

Controle de acesso

APRENDA

O controle de acesso quebrado lidera as listas de riscos web. Ele ocorre quando a aplicação não verifica corretamente o que cada usuário pode ver ou fazer. Lembre-se: autenticação diz quem você é, e a autorização, o que você pode. Muitas aplicações validam a primeira e se esquecem da segunda.

Falhas típicas: IDOR (Insecure Direct Object Reference), em que alterar um número na URL, como /pedido?id=123, mostra o pedido de outra pessoa; acesso a páginas administrativas apenas por conhecer o endereço; funções que só escondem o botão no front-end, mas aceitam a requisição no servidor; e escalada de privilégios horizontal (mesmo nível) ou vertical (nível superior).

Como prevenir: verificar a autorização no servidor em toda requisição, negar por padrão, usar identificadores não previsíveis como apoio (nunca como única proteção), centralizar as regras e registrar tentativas negadas.

NA PRÁTICA

Teste conceitual: logado com a conta A, tente abrir um recurso da conta B (em laboratório, com duas contas suas). O servidor deve responder 403 ou 404, e não entregar o conteúdo.

LABORATÓRIO

No Juice Shop, crie duas contas, crie um item em cada uma e verifique se uma consegue acessar o recurso da outra pela alteração de identificadores. Documente o que observou.

EXERCÍCIOS

1. O que é IDOR?

2. Por que esconder um botão não basta?

3. O que significa 'negar por padrão'?

DESAFIO PABLOCYBERSEC

Liste cinco recursos de uma loja virtual (como pedidos e perfis) e defina, para cada um, quem deve ter acesso.

Correção: *1) Acesso a objetos de outros por mudança de um identificador sem checagem de permissão. 2) A requisição ainda pode ser enviada diretamente. 3) Se não há regra explícita permitindo, o acesso é bloqueado.*

CAPÍTULO 51 · NÍVEL 4 · PENTESTER JÚNIOR

Vulnerabilidades de API

APRENDA

APIs permitem que sistemas conversem entre si, geralmente em REST com dados em JSON, e alimentam aplicativos móveis, sites e integrações. Como não possuem interface visual, são facilmente esquecidas nas revisões de segurança, e como expõem funções diretamente, concentram falhas graves.

A OWASP mantém um Top 10 específico para APIs. Entre as falhas recorrentes: autorização quebrada em nível de objeto (como o IDOR) e de função, autenticação fraca, exposição excessiva de dados (a API devolve campos que o cliente não precisa), ausência de limite de requisições, configuração incorreta e falta de inventário (APIs antigas esquecidas em produção).

Defesas: autenticação robusta com tokens de curta duração, autorização em cada endpoint, devolver apenas os campos necessários, validar o formato dos dados, aplicar limites de taxa (rate limiting), registrar acessos e manter documentação e inventário atualizados.

NA PRÁTICA

```
# Observando uma API de laboratório
curl -s http://192.168.56.102:3000/api/produtos | head -c 400
curl -i -H "Authorization: Bearer SEU_TOKEN" http://192.168.56.102:3000/
api/perfil
```

LABORATÓRIO

No Juice Shop, abra o navegador na aba Rede e identifique as chamadas à API. Verifique que campos são devolvidos e se há dados que o usuário não precisaria ver.

EXERCÍCIOS

1. Por que APIs são frequentemente esquecidas na segurança?

2. O que é exposição excessiva de dados?

3. Para que serve o rate limiting?

DESAFIO PABLOCYBERSEC

Escreva cinco itens de um checklist de revisão de segurança para uma API nova.

Correção: *1) Não têm interface visual e podem ficar fora das revisões. 2) A API devolve mais campos que o necessário. 3) Limitar requisições por tempo, reduzindo abusos como força bruta.*

PARTE 7

Ferramentas

NÍVEL 3 · PRATICANTE

As ferramentas centrais do profissional e, principalmente, como escolher a certa para cada pergunta.

52 Nmap
53 Wireshark
54 Burp Suite
55 Metasploit
56 Gobuster
57 Ferramentas de enumeração
58 Como escolher a ferramenta correta

CAPÍTULO 52 · NÍVEL 3 · PRATICANTE

Nmap

APRENDA

O Nmap (Network Mapper) é a ferramenta de referência para descobrir hosts, portas e serviços. Antes de usá-lo, você já aprendeu o que é uma porta, o que é TCP e o que significa um serviço aberto, e por isso cada opção agora fará sentido: o Nmap apenas automatiza o que você entende.

Opções frequentes: -sn (descoberta sem varrer portas), -p (define portas), -sS (SYN scan, requer privilégios), -sV (versão dos serviços), -O (sistema operacional), -sC (scripts padrão), -A (agressivo, combina várias), -T0 a -T5 (velocidade) e -oN/-oX (salvar resultados em arquivo). O NSE (Nmap Scripting Engine) amplia o poder com scripts para dezenas de verificações.

A saída mostra, para cada porta, o estado (open, closed, filtered), o serviço e a versão. Aprenda a ler isso como um mapa: cada linha é uma pergunta a ser investigada.

NA PRÁTICA

```
# Somente em laboratório autorizado
sudo nmap -sS -sV -O 192.168.56.101
nmap -p 1-1000 -T3 -oN resultado.txt 192.168.56.101
nmap --script http-title -p 80 192.168.56.101
nmap -sn 192.168.56.0/24
```

LABORATÓRIO

Faça três varreduras diferentes da mesma máquina do laboratório e compare tempo, quantidade de pacotes (no Wireshark) e informações obtidas. Salve os resultados em arquivos.

EXERCÍCIOS

1. O que faz a opção -sV?

2. Para que serve -oN?

3. Qual a diferença entre os estados closed e filtered?

DESAFIO PABLOCYBERSEC

Identifique quais serviços estão disponíveis no laboratório e explique o que cada um representa e qual o risco de estar exposto.

Correção: *1) Detecta o serviço e a versão em cada porta. 2) Salva o resultado em arquivo de texto. 3) Closed responde sem serviço; filtered não responde, normalmente por firewall.*

CAPÍTULO 53 · NÍVEL 3 · PRATICANTE

Wireshark

APRENDA

O Wireshark é um analisador de pacotes: captura o tráfego de uma interface de rede e permite examiná-lo em detalhe. É a melhor ferramenta de estudo de redes, pois mostra na prática o que os capítulos anteriores explicaram: o handshake TCP, as consultas DNS, as requisições HTTP.

A interface é dividida em três painéis: lista de pacotes, detalhes do pacote selecionado (em camadas) e bytes brutos. O poder está nos filtros: de captura (aplicados antes de gravar) e de exibição (aplicados sobre o que já foi capturado). Exemplos de filtros de exibição: ip.addr == 192.168.56.101, tcp.port == 80, dns, http, tcp.flags.syn == 1.

Tráfego sem criptografia, como HTTP simples, FTP ou Telnet, pode ser lido por quem o captura. Isso demonstra por que protocolos cifrados (HTTPS, SSH) são tão importantes. Capture somente tráfego da sua própria rede ou de laboratório: interceptar comunicações alheias é crime.

NA PRÁTICA

```
# Capturar via terminal (alternativa ao Wireshark)
sudo tcpdump -i eth0 -c 50 -w captura.pcap
# Abrir depois no Wireshark: wireshark captura.pcap
```

LABORATÓRIO

Capture o tráfego enquanto acessa um site HTTP e outro HTTPS de laboratório. Compare: o que você consegue ler em cada um? Use o filtro dns para ver as consultas de nomes.

EXERCÍCIOS

1. Qual a diferença entre filtro de captura e de exibição?

2. Por que o HTTP simples é inseguro?

3. Como localizar o handshake TCP?

DESAFIO PABLOCYBERSEC

Capture uma sessão de ping e identifique os pacotes ICMP de pedido e resposta. Anote tempos e tamanhos.

Correção: *1) O de captura limita o que é gravado, o de exibição filtra o que já foi gravado. 2) O conteúdo trafega legível. 3) Filtro tcp.flags.syn == 1.*

CAPÍTULO 54 · NÍVEL 3 · PRATICANTE

Burp Suite

APRENDA

O Burp Suite é um conjunto de ferramentas para testar aplicações web. Seu coração é o proxy de interceptação: o navegador envia o tráfego para o Burp, que o mostra e permite pausar, alterar e reenviar cada requisição. Existe a edição Community, gratuita, que serve muito bem para aprender.

Módulos principais: Proxy (intercepta e registra), Repeater (reenvia uma requisição várias vezes, com alterações manuais), Intruder (automatiza variações de uma requisição; é limitado na versão gratuita), Decoder (codifica e decodifica dados) e Comparer (compara respostas). O Scanner é exclusivo da versão paga.

Para usar com HTTPS, instala-se o certificado do Burp no navegador de teste. Use sempre um perfil de navegador separado para não misturar com o uso diário. E lembre-se: só se testam aplicações com autorização.

NA PRÁTICA

Fluxo básico: configurar o proxy do navegador em 127.0.0.1:8080; navegar na aplicação de treino; observar o histórico; enviar requisições interessantes ao Repeater; alterar um parâmetro e comparar as respostas.

LABORATÓRIO

Configure o Burp com o navegador, acesse o Juice Shop ou o DVWA e envie ao Repeater a requisição de login. Altere um parâmetro por vez e observe as diferenças nas respostas.

EXERCÍCIOS

1. Qual a função do proxy do Burp?

2. Para que serve o Repeater?

3. Por que usar um perfil de navegador separado?

DESAFIO PABLOCYBERSEC

Mapeie os endpoints de uma aplicação de treino a partir do histórico do Burp e organize em uma tabela com método, caminho e finalidade.

Correção: *1) Interceptar e permitir a edição de requisições e respostas. 2) Reenviar requisições com alterações manuais. 3) Evita misturar o tráfego de teste com o pessoal.*

CAPÍTULO 55 · NÍVEL 3 · PRATICANTE

Metasploit

APRENDA

O Metasploit Framework é uma plataforma que organiza módulos de teste de segurança: scanners, verificadores, módulos de exploração, cargas (payloads) e ferramentas de pós-exploração. Ele padroniza o uso e economiza tempo em testes autorizados, e também é muito usado em ambientes de treino.

O console (msfconsole) trabalha com módulos que possuem opções a configurar, como o alvo (RHOSTS) e as portas. O fluxo é: pesquisar um módulo relacionado ao serviço e à versão identificados, ler a descrição e as referências, configurar opções, verificar se o alvo parece afetado (quando o módulo oferece um check) e só então executar, no ambiente autorizado.

Cuidado: ferramentas automatizadas escondem o mecanismo e podem desestabilizar sistemas. Leia sempre a documentação do módulo. O objetivo é compreender a falha e demonstrar o risco, não colecionar acessos. Este livro não detalha módulos de ataque específicos; o foco está no fluxo de trabalho.

NA PRÁTICA

```
msfconsole
search type:auxiliary scanner smb   # módulos de verificação
info <módulo>                       # ler descrição e referências
use <módulo>
show options
set RHOSTS 192.168.56.101           # alvo do laboratório
run
```

LABORATÓRIO

No seu laboratório, use um módulo auxiliar de verificação (scanner) contra a máquina vulnerável de treino, leia a documentação (info) e registre o que ele informou.

EXERCÍCIOS

1. O que é um módulo no Metasploit?

2. Para que serve o comando show options?

3. Por que ler a descrição do módulo antes de executar?

DESAFIO PABLOCYBERSEC

Descreva em um parágrafo a diferença entre um módulo auxiliar (scanner) e um módulo de exploração, e quando cada um é apropriado.

Correção: *1) Componente que realiza uma tarefa, como verificar ou testar uma falha. 2) Mostrar as opções configuráveis. 3) Para entender o que ele faz e o risco que oferece ao alvo.*

CAPÍTULO 56 · NÍVEL 3 · PRATICANTE

Gobuster

APRENDA

O Gobuster é uma ferramenta de força bruta de nomes: ele testa uma lista de palavras (wordlist) contra um alvo para descobrir diretórios e arquivos em servidores web, subdomínios e hosts virtuais. Muitas aplicações têm páginas que não aparecem nos menus, mas continuam acessíveis a quem sabe o endereço.

O modo dir procura caminhos em um servidor web; dns busca subdomínios; vhost procura hosts virtuais. O resultado traz os códigos de status: 200 (existe), 301/302 (redirecionamento), 403 (existe, mas proibido), 404 (não existe). Os 403 e os 200 são os que mais merecem atenção.

A qualidade da wordlist influencia o resultado. O Kali traz listas em /usr/share/wordlists e o projeto SecLists é uma coleção popular. Atenção: a ferramenta gera muitas requisições. Combine o ritmo com o cliente e use-a apenas em alvos autorizados.

NA PRÁTICA

```
gobuster dir -u http://192.168.56.102 -w /usr/share/wordlists/dirb/commo
n.txt
gobuster dir -u http://192.168.56.102 -w lista.txt -x php,txt -t 10
```

LABORATÓRIO

Execute o Gobuster contra uma aplicação de treino. Classifique os caminhos encontrados em: públicos, administrativos, de backup e de teste. Em seguida, aponte o que um administrador deveria remover.

EXERCÍCIOS

1. Para que serve o modo dir?

2. O que significa o status 403 para a enumeração?

3. Por que o tamanho da wordlist importa?

DESAFIO PABLOCYBERSEC

Explique por que esconder uma página apenas não a divulgando não é uma medida de segurança.

Correção: *1) Descobrir diretórios e arquivos em servidores web. 2) O recurso existe, mas o acesso é proibido. 3) Listas maiores encontram mais, mas geram mais tráfego.*

CAPÍTULO 57 · NÍVEL 3 · PRATICANTE

Ferramentas de enumeração

APRENDA

Além das ferramentas principais, cada serviço tem suas companheiras de enumeração. O curl e o whatweb ajudam a identificar tecnologias web. O nikto verifica configurações e arquivos conhecidos em servidores web. O enum4linux e o smbclient exploram compartilhamentos SMB. O dnsrecon e o dig exploram o DNS. O whois e o theHarvester coletam dados públicos.

Para recursos mais avançados, o Wappalyzer (extensão de navegador) identifica tecnologias, o Shodan indexa dispositivos expostos à internet e o Censys faz algo semelhante com certificados e serviços. Servem para entender a superfície de ataque de uma organização sem tocar diretamente em seus sistemas.

Nenhuma ferramenta substitui o entendimento. O ideal é conhecer o suficiente do protocolo para saber o que cada uma está fazendo e interpretar o resultado com espírito crítico.

NA PRÁTICA

```
whatweb http://192.168.56.102
nikto -h http://192.168.56.102
smbclient -L //192.168.56.101 -N
dig axfr exemplo.com @ns1.exemplo.com   # só em domínio seu
```

LABORATÓRIO

Escolha uma máquina do seu laboratório e use pelo menos três ferramentas diferentes de enumeração. Compare os resultados e anote quais informações apareceram em mais de uma.

EXERCÍCIOS

1. Para que serve o Shodan?

2. Qual ferramenta pode listar compartilhamentos SMB?

3. Por que cruzar resultados de ferramentas diferentes?

DESAFIO PABLOCYBERSEC

Monte uma tabela de ferramentas: nome, para que serve, protocolo alvo e um exemplo de comando.

Correção: *1) Indexar dispositivos e serviços expostos na internet. 2) smbclient ou enum4linux. 3) Confirma achados e reduz falsos positivos.*

CAPÍTULO 58 · NÍVEL 3 · PRATICANTE

Como escolher a ferramenta correta

APRENDA

Um iniciante tende a decorar ferramentas. Um profissional parte da pergunta: o que preciso descobrir? Só então escolhe a ferramenta. Comece sempre pelo objetivo e pelo protocolo envolvido.

Guia rápido: descobrir hosts, usar Nmap -sn; descobrir portas e serviços, Nmap; analisar tráfego, Wireshark ou tcpdump; interagir com aplicações web, navegador, curl e Burp Suite; descobrir diretórios, Gobuster; verificar falhas conhecidas, scanners como Nikto ou OpenVAS; analisar hashes e integridade, sha256sum e OpenSSL; automatizar algo sob medida, Python.

Critérios práticos: a ferramenta é adequada ao que quero provar? Quanto ruído gera? Tenho autorização para esse tipo de ação? Entendo a saída? Existe uma forma menos invasiva de chegar à mesma resposta? Se você sabe fazer o essencial à mão com curl ou netcat, você entende o que a ferramenta automatiza.

NA PRÁTICA

Regra de ouro: automatize só o que você entende. Ferramentas diferentes podem dar resultados diferentes, e o analista é quem decide o que é real.

LABORATÓRIO

Dadas as perguntas a seguir, escolha a ferramenta e justifique: (a) Quais máquinas estão ligadas na rede? (b) Esse site expõe um painel administrativo? (c) O que passa no cabo quando faço login em um site HTTP? (d) Esse arquivo foi alterado desde ontem?

EXERCÍCIOS

1. Por que partir do objetivo antes de escolher a ferramenta?

2. Cite dois critérios para escolher uma ferramenta.

3. Por que saber fazer à mão ajuda?

DESAFIO PABLOCYBERSEC

Para um alvo fictício (um servidor web com SSH e banco de dados), planeje a sequência de ferramentas que usaria e o motivo de cada uma.

Correção: *1) Porque a ferramenta é um meio, não o fim. 2) Adequação ao objetivo e autorização, por exemplo. 3) Garante que você entende o que a ferramenta automatiza. Lab: (a) Nmap -sn, (b) Gobuster ou navegador, (c) Wireshark, (d) sha256sum.*

PARTE 8

Nível Avançado

NÍVEL 5 · AVANÇADO

Conceitos avançados com foco em compreensão, detecção e defesa em ambiente de laboratório.

59 Exploração de vulnerabilidades
60 Privilege Escalation
61 Buffer Overflow: fundamentos
62 Engenharia Reversa: introdução
63 Red Team: conceitos
64 OPSEC
65 Persistência: conceitos defensivos e ofensivos em laboratório
66 Evasão e detecção
67 Introdução à análise de malware

CAPÍTULO 59 · NÍVEL 5 · AVANÇADO

Exploração de vulnerabilidades

APRENDA

Neste nível, deixamos de apenas usar ferramentas prontas e passamos a entender por que as falhas existem. Quase toda vulnerabilidade nasce de uma suposição errada do programador: assumir que a entrada terá certo tamanho, que o usuário seguirá o fluxo previsto ou que dois eventos nunca acontecerão ao mesmo tempo.

Classes comuns de falha: validação insuficiente de entradas (injeções), erros de gerenciamento de memória (estouros de buffer), lógica de negócio falha (pular etapas, manipular preços), condições de corrida, desserialização insegura, configurações padrão e falhas de autenticação e autorização. Cada uma tem causa, sintoma e correção características.

O raciocínio de um analista avançado é: qual suposição o sistema faz? Como eu a quebro de forma segura em laboratório? Qual o impacto? Como o código deveria ser escrito? A resposta final é sempre a correção. Quem entende as causas escreve software mais seguro e produz relatórios mais valiosos.

NA PRÁTICA

Ciclo de estudo recomendado: ler a descrição de um CVE antigo e bem documentado, entender o código vulnerável e o código corrigido, e reproduzir em uma máquina de laboratório, somente para compreender o mecanismo.

LABORATÓRIO

Escolha um CVE antigo e famoso. Escreva uma página com: componente afetado, causa raiz, impacto, versão corrigida e a diferença conceitual entre o código antes e depois da correção.

EXERCÍCIOS

1. De onde nascem a maioria das vulnerabilidades?

2. O que é uma falha de lógica de negócio?

3. Por que a correção é o produto final?

DESAFIO PABLOCYBERSEC

Descreva uma falha de lógica de negócio possível em um carrinho de compras online e como preveni-la.

Correção: *1) De suposições erradas feitas pelo desenvolvedor. 2) Uma falha no fluxo de regras, e não no código técnico, como alterar o preço enviado. 3) O objetivo é reduzir o risco, e não apenas demonstrá-lo.*

CAPÍTULO 60 · NÍVEL 5 · AVANÇADO

Privilege Escalation

APRENDA

Escalada de privilégios é a passagem de um nível de acesso menor para um maior, como de usuário comum para administrador. Ela é horizontal quando se alcança outra conta do mesmo nível, e vertical quando se alcança um nível superior. Em testes e em incidentes reais, é uma etapa central, porque o acesso inicial costuma ser limitado.

Do ponto de vista conceitual, as causas mais comuns são: configurações de permissões frouxas, serviços rodando com mais privilégio do que precisam, regras de sudo excessivas, programas com bit SUID mal controlados, tarefas agendadas executando arquivos graváveis por qualquer usuário, credenciais esquecidas em arquivos e softwares desatualizados no sistema.

A abordagem profissional é a auditoria de configuração: em laboratório, um analista examina o que a conta atual pode fazer e o que o sistema oferece de forma indevida. A defesa é bem objetiva: princípio do menor privilégio, atualizações, revisão de sudo e de permissões, segredos fora de arquivos em texto simples e monitoramento de comportamento anômalo.

NA PRÁTICA

```
# Auditoria de configuração no SEU laboratório
id
sudo -l                              # o que o usuário pode executar com
 sudo
find / -perm -4000 -type f 2>/dev/null   # programas com bit SUID
ls -l /etc/cron.d /etc/crontab
```

LABORATÓRIO

Na sua máquina virtual, execute os comandos acima como usuário comum, liste os resultados e classifique cada item como normal do sistema ou digno de revisão. Pesquise o motivo de cada SUID padrão.

EXERCÍCIOS

1. Qual a diferença entre escalada horizontal e vertical?

2. Cite duas causas comuns de escalada.

3. Qual a principal defesa contra isso?

DESAFIO PABLOCYBERSEC

Escreva um checklist de endurecimento (hardening) de 8 itens que reduz a chance de escalada de privilégios em um servidor Linux.

Correção: *1) Horizontal mantém o nível; vertical aumenta o nível de acesso. 2) Permissões frouxas e sudo excessivo, por exemplo. 3) Princípio do menor privilégio com atualizações e revisões regulares.*

CAPÍTULO 61 · NÍVEL 5 · AVANÇADO

Buffer Overflow: fundamentos

APRENDA

Um buffer é uma área de memória reservada para guardar dados. Um estouro de buffer (buffer overflow) ocorre quando um programa grava mais dados do que a área comporta, sobrescrevendo regiões vizinhas da memória. Em linguagens como C e C++, que não verificam limites automaticamente, o erro é possível; em linguagens como Python e Java, o risco é muito menor.

Para entender, imagine a pilha (stack) de um programa como uma sequência de caixas: variáveis locais, dados de controle e o endereço de retorno da função. Se o programa escreve além do tamanho da caixa, pode alterar o que está ao lado. Em casos graves, isso permite alterar o fluxo do programa. Por isso a falha já causou alguns dos incidentes mais famosos da história.

As defesas modernas tornam o problema bem mais difícil: ASLR (aleatoriza endereços), DEP/NX (impede execução em áreas de dados), canários de pilha (detectam sobrescrita), compiladores com verificações e o uso de linguagens com segurança de memória, como Rust e Go. A melhor prevenção é escrever código que valide tamanhos.

NA PRÁTICA

```
// Exemplo de padrão INSEGURO (apenas para análise de código)
char destino[16];
strcpy(destino, entrada);        // não verifica tamanho

// Padrão mais seguro
strncpy(destino, entrada, sizeof(destino) - 1);
destino[sizeof(destino) - 1] = '\0';
```

LABORATÓRIO

Leia o código acima e explique, por escrito, por que a primeira versão é perigosa. Pesquise quais funções da linguagem C são consideradas arriscadas (gets, strcpy, sprintf) e quais as alternativas.

EXERCÍCIOS

1. O que é um buffer overflow?

2. Por que linguagens como Python são menos suscetíveis?

3. Cite duas proteções modernas contra esse tipo de falha.

DESAFIO PABLOCYBERSEC

Explique, com um desenho da pilha, como a escrita além do limite de uma variável pode afetar dados vizinhos.

Correção: *1) Gravar mais dados do que a área de memória comporta. 2) Verificam limites e gerenciam a memória automaticamente. 3) ASLR, DEP/NX e canários de pilha.*

CAPÍTULO 62 · NÍVEL 5 · AVANÇADO

Engenharia Reversa: introdução

APRENDA

Engenharia reversa é analisar um programa já compilado para entender como ele funciona, sem acesso ao código-fonte. Serve para análise de malware, auditoria de software, pesquisa de vulnerabilidades e compatibilidade. É uma habilidade de longo prazo, que combina conhecimento de sistemas, arquitetura de computadores e paciência.

A análise se divide em estática (examinar o arquivo sem executá-lo) e dinâmica (observar a execução em ambiente controlado). Na estática, começa-se por informações simples: tipo de arquivo, textos legíveis dentro do binário, bibliotecas importadas e metadados. Depois, usam-se desassembladores e decompiladores, como o Ghidra (gratuito, da NSA) e o radare2, para ler o código em assembly ou em pseudocódigo.

A legalidade importa: a engenharia reversa de software de terceiros pode ser restringida por licenças e leis de direitos autorais. Em estudos, use programas próprios, desafios educativos (crackmes) e exemplos de livre distribuição.

NA PRÁTICA

```
file programa
strings programa | head -30
sha256sum programa
ldd programa              # bibliotecas usadas
objdump -d programa | head -40
```

LABORATÓRIO

Compile um programa simples em C que peça uma senha e compare com um texto fixo. Use strings e o Ghidra para descobrir o texto sem olhar o código-fonte. Reflita: o que isso ensina sobre guardar segredos dentro de programas?

EXERCÍCIOS

1. Qual a diferença entre análise estática e dinâmica?

2. Para que serve o comando strings?

3. Cite uma ferramenta de decompilação gratuita.

DESAFIO PABLOCYBERSEC

Pesquise o que é um crackme e resolva um de nível iniciante, registrando seu raciocínio.

Correção: *1) A estática examina sem executar, a dinâmica observa a execução. 2) Mostra textos legíveis dentro de um arquivo binário. 3) Ghidra.*

CAPÍTULO 63 · NÍVEL 5 · AVANÇADO

Red Team: conceitos

APRENDA

O Red Team simula um adversário real para testar a capacidade de detecção e resposta de uma organização, e não apenas listar falhas. Enquanto um pentest costuma procurar o maior número de vulnerabilidades em um escopo definido, um exercício de Red Team persegue objetivos específicos, como alcançar um dado crítico, de forma realista e discreta, medindo se a equipe de defesa percebe.

Elementos usados: inteligência sobre ameaças, mapeamento de táticas e técnicas reais de grupos atacantes (o framework MITRE ATT&CK é a referência), engenharia social autorizada, testes físicos e técnicos, e regras de engajamento rígidas, com um grupo de controle (white team) que acompanha o exercício e garante a segurança.

O Purple Team une as duas pontas: ofensivo e defensivo trabalham juntos, testando técnicas específicas e ajustando a detecção em tempo real. O resultado mais valioso não é a invasão, mas o aprendizado que melhora as defesas.

NA PRÁTICA

MITRE ATT&CK organiza o comportamento do adversário em táticas (o objetivo, como acesso inicial ou movimento lateral) e técnicas (o método). É um idioma comum entre atacantes simulados e defensores.

LABORATÓRIO

Acesse o site do MITRE ATT&CK, escolha uma tática e leia três técnicas. Para cada uma, escreva como uma equipe defensiva poderia detectá-la.

EXERCÍCIOS

1. Qual a diferença entre um pentest e um exercício de Red Team?

2. O que é o MITRE ATT&CK?

3. Qual o papel do Purple Team?

DESAFIO PABLOCYBERSEC

Desenhe um cenário fictício de Red Team (objetivo, regras de engajamento e como o sucesso seria medido).

Correção: *1) O Red Team mede detecção e resposta com objetivos realistas; o pentest lista falhas. 2) Base de conhecimento de táticas e técnicas de adversários. 3) Integrar ataque e defesa para melhorar a detecção.*

CAPÍTULO 64 · NÍVEL 5 · AVANÇADO

OPSEC

APRENDA

OPSEC (segurança operacional) é o conjunto de práticas para evitar que informações sensíveis vazem pelas próprias atividades. Surgiu no meio militar e hoje é aplicada em empresas, em investigações, em pesquisa de segurança e na vida pessoal. A pergunta central é: que informações um adversário poderia reunir sobre mim e como as usaria?

O processo clássico tem cinco passos: identificar as informações críticas, analisar as ameaças, analisar as vulnerabilidades, avaliar os riscos e aplicar contramedidas. Em uma equipe de segurança, isso significa proteger documentos de clientes, credenciais de teste, evidências e comunicações do

projeto.

Práticas para o profissional: separar identidades e dispositivos de trabalho e pessoais, usar gerenciador de senhas e MFA, cifrar discos e backups, não reutilizar credenciais, compartilhar o mínimo em redes sociais, guardar evidências em local seguro e destruí-las ao final do prazo contratado, conforme o contrato.

NA PRÁTICA

Pergunta útil: se alguém reunisse tudo o que está publicado sobre você e sua empresa, que conclusões poderia tirar?

LABORATÓRIO

Faça uma auditoria de OPSEC pessoal: pesquise seu nome e seus apelidos online, revise as configurações de privacidade das redes sociais, e liste três informações que você decidiu reduzir ou remover.

EXERCÍCIOS

1. O que é OPSEC?

2. Quais são as cinco etapas do processo?

3. Cite duas práticas de OPSEC para quem trabalha com segurança.

DESAFIO PABLOCYBERSEC

Elabore uma política de manuseio de evidências para seus testes: onde são armazenadas, quem acessa e quando são destruídas.

Correção: *1) Prática de evitar que informações sensíveis vazem pelas atividades. 2) Identificar informações críticas, ameaças, vulnerabilidades, riscos e contramedidas. 3) Cifrar discos e separar identidades de trabalho e pessoais, por exemplo.*

CAPÍTULO 65 · NÍVEL 5 · AVANÇADO

Persistência: conceitos defensivos e ofensivos em laboratório

APRENDA

Persistência é a capacidade de um acesso sobreviver a reinicializações, trocas de senha ou interrupções. Em incidentes reais, quem invade costuma deixar mecanismos para voltar depois, e por isso a equipe de resposta precisa saber procurá-los. Neste livro, o tema é abordado pelo lado da detecção e da auditoria.

Locais clássicos que a defesa precisa revisar: serviços e tarefas agendadas (cron, systemd, Agendador de Tarefas), programas de inicialização, chaves de inicialização do registro no Windows, contas de usuário novas ou alteradas, chaves SSH autorizadas e arquivos de perfil do shell. Qualquer item inesperado nesses locais é um indício a investigar.

Boas práticas defensivas: manter uma linha de base do que é normal, monitorar mudanças nesses pontos com ferramentas de integridade, alertar sobre novas contas e serviços, restringir quem pode criar tarefas agendadas e registrar tudo em um local centralizado. Em um pentest, relata-se o risco e remove-se tudo o que foi criado ao final.

NA PRÁTICA

```
# Revisão defensiva em Linux (laboratório)
crontab -l
ls -l /etc/cron.d
systemctl list-unit-files --state=enabled | head -20
cat ~/.ssh/authorized_keys
awk -F: '$3 >= 1000 {print $1}' /etc/passwd
```

LABORATÓRIO

Na sua máquina virtual, execute a revisão acima e salve a saída como linha de base. Crie um novo usuário e uma tarefa agendada de teste e depois verifique como a comparação com a linha de base revela as mudanças. Remova-os ao final.

EXERCÍCIOS

1. O que significa persistência em segurança?

2. Cite dois locais que a defesa deve revisar.

3. O que é uma linha de base?

DESAFIO PABLOCYBERSEC

Monte uma lista de verificação de persistência para um servidor Linux, com um item para cada local de revisão.

Correção: *1) A capacidade de manter acesso após reinicializações ou mudanças. 2) Tarefas agendadas e chaves SSH autorizadas, por exemplo. 3) O registro do estado normal do sistema, usado para comparação.*

CAPÍTULO 66 · NÍVEL 5 · AVANÇADO

Evasão e detecção

APRENDA

Os defensores usam antivírus, EDRs, firewalls, sistemas de detecção de intrusão (IDS/IPS) e plataformas de análise de logs (SIEM). Os atacantes tentam passar despercebidos. Entender essa corrida é importante, mas aqui o foco é a defesa: como detectar o que tenta se esconder.

Os mecanismos de detecção usam assinaturas (padrões conhecidos), heurística (comportamentos suspeitos), análise de anomalias e inteligência de ameaças. Nenhum é perfeito: assinaturas não pegam o desconhecido, e a análise de comportamento gera falsos positivos. Por isso a defesa em camadas, com várias fontes de dados, é indispensável.

Bons sinais para os defensores: processos filhos incomuns (como um editor de documentos abrindo um terminal), conexões para destinos raros, picos de tráfego fora do horário, uso de ferramentas administrativas por contas comuns, apagamento de logs e criação de serviços novos. Boas detecções combinam vários sinais fracos em um alerta forte.

NA PRÁTICA

Pense como detetive: um único dado raramente prova algo; a correlação entre eventos conta a história.

LABORATÓRIO

Escolha um comportamento suspeito (por exemplo, muitas falhas de login seguidas de um sucesso). Descreva em português uma regra de detecção: quais eventos, em qual janela de tempo, com qual limite e qual alerta gerar.

EXERCÍCIOS

1. Qual a diferença entre detecção por assinatura e por comportamento?

2. O que é um SIEM?

3. Por que a defesa em camadas é necessária?

DESAFIO PABLOCYBERSEC

Escreva três regras de detecção em linguagem simples para um servidor web: uma para autenticação, uma para tráfego e uma para alterações de arquivos.

Correção: *1) A assinatura procura padrões conhecidos, o comportamento observa ações suspeitas. 2) Plataforma que reúne e correlaciona logs de várias fontes. 3) Nenhum mecanismo isolado detecta tudo.*

CAPÍTULO 67 · NÍVEL 5 · AVANÇADO

Introdução à análise de malware

APRENDA

Analisar malware é entender o que um programa malicioso faz, como se espalha e como detectá-lo, para proteger sistemas e responder a incidentes. Como o objetivo é compreender e defender, o trabalho exige um ambiente rigorosamente isolado, pois executar uma amostra em um computador comum pode causar dano real.

O laboratório seguro: máquina virtual sem acesso à rede real (ou em rede fechada), snapshots antes de qualquer teste, nenhuma pasta compartilhada com o computador principal e nenhum dado pessoal. Use amostras de fontes legítimas para estudo e sempre dentro dessa estrutura.

A análise estática inicial é segura e muito informativa: calcular o hash e consultá-lo em bases públicas como o VirusTotal, identificar o tipo do arquivo, listar textos e bibliotecas importadas, e verificar propriedades. A análise dinâmica, que observa o comportamento (arquivos criados, conexões, alterações), exige ainda mais cuidado e experiência. O resultado final costuma ser uma lista de indicadores de comprometimento (IoCs) para detecção.

NA PRÁTICA

IoCs comuns: hashes de arquivos, endereços IP e domínios contatados, nomes de arquivos criados, chaves de registro e mutexes.

```
sha256sum amostra.bin              # identificador da amostra
file amostra.bin
strings -n 8 amostra.bin | head -40
```

LABORATÓRIO

Sem executar nada suspeito, pegue um arquivo inofensivo qualquer, calcule seu hash, identifique o tipo e liste os textos legíveis. Depois descreva em uma página como você montaria um ambiente isolado para analisar uma amostra de verdade.

EXERCÍCIOS

1. Por que é essencial isolar o ambiente de análise?

2. Qual a vantagem de começar pela análise estática?

3. O que são IoCs?

DESAFIO PABLOCYBERSEC

Escreva um passo a passo de segurança para a análise de malware: antes, durante e depois da análise.

Correção: *1) Para evitar que a amostra cause dano real ou se espalhe. 2) É segura e já fornece informações úteis. 3) Indicadores usados para detectar a ameaça, como hashes e domínios.*

PARTE 9

Projeto Final

DESAFIO FINAL PABLOCYBERSEC

Um pentest autorizado completo, do laboratório ao relatório profissional.

68 Montando um laboratório de Cibersegurança
69 Criando um cenário de Pentest autorizado
70 Projeto final: Reconhecimento
71 Projeto final: Enumeração
72 Projeto final: Identificação das vulnerabilidades
73 Projeto final: Exploração controlada
74 Projeto final: Análise dos resultados
75 Projeto final: Correção das vulnerabilidades
76 Projeto final: Relatório profissional
77 Checklist do Pentester

CAPÍTULO 68 · DESAFIO FINAL PABLOCYBERSEC

Montando um laboratório de Cibersegurança

APRENDA

O laboratório é o seu campo de treino: um ambiente isolado, onde você pode errar, quebrar e reiniciar sem prejudicar ninguém. Ele reúne ao menos duas máquinas virtuais: a máquina de ataque (Kali Linux) e uma ou mais máquinas-alvo intencionalmente vulneráveis (Metasploitable, DVWA, OWASP Juice Shop).

Requisitos mínimos: 8 GB de RAM (ideal 16), 100 GB livres, um virtualizador (VirtualBox ou VMware) e as imagens obtidas nos sites oficiais. Configure uma rede interna ou somente-hospedeiro (host-only), para que as máquinas vulneráveis nunca fiquem expostas à internet ou à sua rede doméstica. Máquinas vulneráveis em rede aberta são um risco real.

Boas práticas: um snapshot limpo de cada máquina, nomes organizados, um diário de laboratório, e um endereçamento fixo e documentado. Exemplo: Kali em 192.168.56.10, Metasploitable em 192.168.56.101 e Juice Shop em 192.168.56.102.

NA PRÁTICA

Esquema: [Kali 192.168.56.10] <-- rede host-only --> [Metasploitable .101] e [Juice Shop .102]. Nenhuma das máquinas-alvo deve ter acesso à internet.

```
# Verificações iniciais no Kali
ip a
ping -c 2 192.168.56.101
nmap -sn 192.168.56.0/24
```

LABORATÓRIO

Monte o laboratório completo, confirme a comunicação entre as máquinas, tire os snapshots e documente em uma tabela: nome, sistema, IP, função e senha de acesso do laboratório.

EXERCÍCIOS

1. Por que isolar as máquinas vulneráveis da rede real?

2. O que é um snapshot e por que usá-lo?

3. Que máquinas compõem o laboratório básico?

DESAFIO PABLOCYBERSEC

Faça um diagrama do seu laboratório com IPs e rede, e redija regras de uso (por exemplo: nunca ligar as máquinas-alvo em modo bridge).

Correção: *1) Elas contêm falhas conhecidas e poderiam ser invadidas por terceiros. 2) Cópia do estado da máquina para restaurar depois. 3) Uma máquina de ataque e uma ou mais máquinas-alvo.*

CAPÍTULO 69 · DESAFIO FINAL PABLOCYBERSEC

Criando um cenário de Pentest autorizado

APRENDA

Agora você vai reunir tudo em um projeto completo, como um pentest de verdade, do documento de autorização ao relatório final. O cenário: uma empresa fictícia, a Loja Exemplo, contratou você para testar seu ambiente de teste, composto por um servidor Linux com serviços variados e uma aplicação web.

Primeiro, formalize o planejamento: escreva a autorização (modelo do Capítulo 3), defina o escopo (IPs 192.168.56.101 e 192.168.56.102), o período, as regras de engajamento (sem negação de serviço, sem acesso à internet), os contatos e o formato do relatório. Defina também o nível do teste: caixa preta, partindo apenas dos IPs.

Crie uma pasta de projeto com a estrutura: 00-autorizacao, 01-recon, 02-enumeracao, 03-vulnerabilidades, 04-exploracao, 05-evidencias e 06-relatorio. Essa disciplina de organização é um sinal de profissionalismo e vai poupar horas no final.

NA PRÁTICA

```
mkdir -p projeto-loja/{00-autorizacao,01-recon,02-enumeracao,03-vulnerab
ilidades,04-exploracao,05-evidencias,06-relatorio}
ls projeto-loja
```

LABORATÓRIO

Escreva o documento de autorização e escopo do projeto final (uma página) e crie a estrutura de pastas. Defina o cronograma de cada fase.

EXERCÍCIOS

1. O que deve constar no escopo?

2. Para que servem as regras de engajamento?

3. Por que organizar pastas por fase?

DESAFIO PABLOCYBERSEC

Crie um cronograma de 5 dias para o pentest do cenário, com as atividades de cada dia.

Correção: *1) Os alvos permitidos, o período e o que pode ou não ser feito. 2) Definir limites e evitar danos ou mal-entendidos. 3) Facilita a documentação, a rastreabilidade e o relatório.*

CAPÍTULO 70 · DESAFIO FINAL PABLOCYBERSEC

Projeto final: Reconhecimento

APRENDA

Com o escopo aprovado, começa a fase de reconhecimento. Como o cenário é um laboratório fechado, o recon passivo é limitado, mas você deve registrar o que seria feito em um caso real: consulta de domínio, DNS, tecnologias, e-mails públicos, repositórios e perfis. Anote tudo, até mesmo o que não foi possível fazer e por quê.

No recon ativo, valide a conectividade com os alvos, descubra os hosts ativos na rede do laboratório e confirme que os IPs encontrados estão dentro do escopo. Nunca teste um endereço que não está na lista autorizada.

Produza como entrega um documento de reconhecimento com: lista de hosts ativos, sistema operacional estimado, informações públicas relevantes e observações. Esse documento será a base da enumeração.

NA PRÁTICA

```
nmap -sn 192.168.56.0/24 -oN projeto-loja/01-recon/hosts.txt
ping -c 2 192.168.56.101
ping -c 2 192.168.56.102
```

LABORATÓRIO

Execute o reconhecimento do projeto, grave os resultados na pasta 01-recon e redija um resumo de meia página com os hosts encontrados e eventuais pontos de atenção.

EXERCÍCIOS

1. Por que registrar o que não foi possível fazer?

2. O que fazer se um IP fora do escopo aparecer?

3. Qual a entrega da fase de reconhecimento?

DESAFIO PABLOCYBERSEC

Adicione uma tabela de hosts com IP, nome, sistema estimado e função provável.

Correção: *1) Para documentar limitações e dar transparência ao cliente. 2) Não testar, apenas registrar e comunicar. 3) Documento com hosts ativos e observações.*

CAPÍTULO 71 · DESAFIO FINAL PABLOCYBERSEC

Projeto final: Enumeração

APRENDA

Com os hosts mapeados, chega a hora de descobrir quais serviços cada um oferece. Faça uma varredura completa de portas TCP, depois identifique serviços e versões. Em seguida, aprofunde por serviço: banners, compartilhamentos, diretórios web, métodos HTTP e cabeçalhos.

Trabalhe de forma sistemática: um host por vez, uma porta por vez, anotando tudo na pasta 02-enumeracao. Registre o comando, o horário e a saída. Quando algo chamar a atenção, marque para investigar na fase de vulnerabilidades.

Entrega: uma tabela completa de serviços (host, porta, protocolo, serviço, versão e observações) e um mapa de superfície de ataque da aplicação web (páginas, formulários, parâmetros, APIs).

NA PRÁTICA

```
nmap -p- -T3 -oN projeto-loja/02-enumeracao/portas.txt 192.168.56.101
nmap -sC -sV -oN projeto-loja/02-enumeracao/servicos.txt 192.168.56.101
gobuster dir -u http://192.168.56.102 -w /usr/share/wordlists/dirb/commo
n.txt -o projeto-loja/02-enumeracao/dirs.txt
```

LABORATÓRIO

Execute a enumeração dos dois alvos, salve as saídas e monte a tabela de serviços na pasta do projeto.

EXERCÍCIOS

1. Qual a diferença entre varredura de portas e identificação de serviços?

2. Por que salvar todas as saídas?

3. O que é o mapa de superfície de ataque?

DESAFIO PABLOCYBERSEC

Priorize os três serviços mais promissores para investigação e explique o porquê de cada escolha.

Correção: *1) A varredura acha portas abertas; a identificação diz qual programa e versão respondem. 2) Para evidência, reprodução e relatório. 3) O conjunto de pontos de entrada que um atacante poderia explorar.*

CAPÍTULO 72 · DESAFIO FINAL PABLOCYBERSEC

Projeto final: Identificação das vulnerabilidades

APRENDA

Agora analise o que foi enumerado. Cruze as versões dos serviços com bases públicas (NVD, Exploit-DB, avisos dos fabricantes), execute scanners de vulnerabilidades e revise manualmente as configurações e as funcionalidades da aplicação web, seguindo o OWASP Top 10.

Para cada candidato a vulnerabilidade, anote: onde está, como foi identificada, evidência, severidade preliminar (usando CVSS ou escala interna) e a confiança no achado. Descarte os falsos positivos explicando como verificou. Itens que dependem de teste adicional ficam na lista de pendências para a próxima etapa.

Entrega: um registro de vulnerabilidades com identificador próprio (VULN-01, VULN-02, ...), título, localização, descrição, severidade e status (confirmada, a confirmar, descartada).

NA PRÁTICA

```
nikto -h http://192.168.56.102 -o projeto-loja/03-vulnerabilidades/nikto
.txt
nmap --script vuln -oN projeto-loja/03-vulnerabilidades/nmap-vuln.txt 19
2.168.56.101
```

LABORATÓRIO

Monte o registro de vulnerabilidades do projeto com no mínimo seis itens, em uma planilha com as colunas indicadas.

EXERCÍCIOS

1. Por que descartar falsos positivos com justificativa?

2. O que é severidade preliminar?

3. Como o OWASP Top 10 ajuda nessa fase?

DESAFIO PABLOCYBERSEC

Ordene as vulnerabilidades por prioridade de teste e de correção, justificando cada decisão.

Correção: *1) Para provar que a análise foi criteriosa. 2) Uma classificação inicial que pode mudar após os testes. 3) Serve como roteiro de categorias a verificar na aplicação.*

CAPÍTULO 73 · DESAFIO FINAL PABLOCYBERSEC

Projeto final: Exploração controlada

APRENDA

Com as vulnerabilidades priorizadas, execute a validação controlada, sempre dentro do escopo e das regras. O objetivo é confirmar que a falha é real e demonstrar o impacto, com o mínimo necessário. Em um cenário de laboratório, você tem liberdade para aprofundar; em uma contratação real, siga estritamente o combinado.

Para cada vulnerabilidade, siga este roteiro: confirme a causa, planeje o teste menos invasivo, execute, registre o resultado e as evidências (comando, captura de tela, horário) e reverta qualquer alteração. Antes de testes de maior risco, tire um snapshot do alvo.

Reflita sobre a cadeia de eventos: uma falha isolada pode ser de baixo impacto, mas combinada com outra pode ficar grave. Registre essas combinações, pois elas ajudam o cliente a priorizar. Resista à tentação de ir além do necessário: provar o risco é o suficiente.

NA PRÁTICA

Modelo de evidência: ID da vulnerabilidade; data e hora; ação executada; resultado; arquivo de captura; observações; ação de reversão feita.

LABORATÓRIO

Valide ao menos três vulnerabilidades do seu registro, documentando cada uma conforme o modelo. Restaure os snapshots ao final.

EXERCÍCIOS

1. Qual o objetivo da exploração controlada?

2. Por que tirar snapshot antes de testes arriscados?

3. O que são falhas encadeadas?

DESAFIO PABLOCYBERSEC

Explique, em um parágrafo, a cadeia de uma falha de baixo impacto que, combinada com outra, se torna crítica.

Correção: *1) Confirmar a falha e demonstrar o impacto com o mínimo de interferência. 2) Para poder restaurar o estado original. 3) Falhas que, combinadas, aumentam o impacto.*

CAPÍTULO 74 · DESAFIO FINAL PABLOCYBERSEC

Projeto final: Análise dos resultados

APRENDA

Os testes terminaram. Agora é hora de transformar dados em conhecimento. Reúna todas as evidências, confirme cada achado e classifique em definitivo: crítica, alta, média, baixa ou informativa. Considere probabilidade, impacto e contexto, e não apenas a pontuação técnica.

Busque os padrões. Muitas vulnerabilidades têm causas em comum: ausência de atualizações, configuração padrão, falta de validação de entradas, controles de acesso mal implementados. Agrupar por causa raiz permite recomendações mais eficientes: corrigir uma raiz resolve vários sintomas.

Também registre o que estava bem: controles que funcionaram e tentativas que foram bloqueadas. Um relatório que mostra só os problemas passa uma visão incompleta, e o equilíbrio fortalece a credibilidade.

NA PRÁTICA

Matriz sugerida: ID | Título | Severidade | Causa raiz | Impacto | Esforço de correção | Prioridade.

LABORATÓRIO

Complete a matriz com todos os achados do projeto e agrupe-os por causa raiz. Escreva três conclusões gerais sobre a postura de segurança do ambiente.

EXERCÍCIOS

1. Por que agrupar vulnerabilidades por causa raiz?

2. Por que registrar também os pontos positivos?

3. Que fatores definem a severidade final?

DESAFIO PABLOCYBERSEC

Redija um parágrafo de conclusão técnica do projeto, resumindo os principais problemas e sua origem.

Correção: *1) Porque corrigir a raiz resolve vários problemas ao mesmo tempo. 2) Dá equilíbrio e credibilidade ao relatório. 3) Probabilidade, impacto, contexto e facilidade de exploração.*

CAPÍTULO 75 · DESAFIO FINAL PABLOCYBERSEC

Projeto final: Correção das vulnerabilidades

APRENDA

O verdadeiro valor do pentest está na correção. Para cada vulnerabilidade, escreva uma recomendação clara, específica e viável: o que mudar, onde e como verificar. Recomendações vagas, como 'melhorar a segurança', não ajudam ninguém.

Exemplos de recomendações: atualizar o serviço X para a versão corrigida; desativar um serviço desnecessário e fechar a porta no firewall; trocar credenciais padrão e exigir MFA; usar consultas parametrizadas na função de busca; incluir verificação de autorização no servidor em cada endpoint; aplicar cabeçalhos de segurança; habilitar logs e alertas.

Em laboratório, aplique as correções no alvo e execute o reteste: confirme que a falha deixou de existir e que nada novo foi quebrado. Registre o antes e o depois. Esse ciclo (testar, corrigir, retestar) é a essência do trabalho de segurança.

NA PRÁTICA

Modelo de recomendação: Problema; Onde; Correção recomendada; Como verificar; Prazo sugerido; Responsável sugerido.

LABORATÓRIO

Corrija pelo menos três vulnerabilidades do seu laboratório (por exemplo, desativar um serviço, alterar uma configuração, atualizar um pacote), reteste e documente antes e depois.

EXERCÍCIOS

1. O que torna uma recomendação boa?

2. Por que o reteste é obrigatório?

3. Cite duas correções comuns.

DESAFIO PABLOCYBERSEC

Redija recomendações no modelo para três vulnerabilidades do seu registro.

Correção: *1) Ser específica, viável e verificável. 2) Confirma que a correção funcionou e não gerou novos problemas. 3) Atualização de software e desativação de serviços desnecessários, por exemplo.*

CAPÍTULO 76 · DESAFIO FINAL PABLOCYBERSEC

Projeto final: Relatório profissional

APRENDA

Chegou o momento de entregar o produto final. Use a estrutura do Capítulo 42: capa e controle de versão, sumário executivo, escopo e metodologia, resumo dos achados, detalhamento de cada achado, recomendações priorizadas, conclusão e anexos.

O sumário executivo é a parte mais lida. Em no máximo uma página, em linguagem de negócio, diga o que foi testado, o resultado geral, os riscos mais importantes e as ações prioritárias. Evite jargões e siglas sem explicação.

Revise com cuidado: gramática, consistência de nomes e números, evidências correspondentes a cada achado, ausência de dados sensíveis desnecessários e tom respeitoso. Peça a outra pessoa para ler como se fosse o cliente. Um relatório claro e organizado é o seu cartão de visitas.

NA PRÁTICA

Lista de revisão: todos os achados têm evidência? Todas as recomendações são acionáveis? A severidade é coerente? O resumo executivo cabe em uma página? Há dados sensíveis que precisam ser mascarados?

LABORATÓRIO

Redija o relatório completo do projeto final. Entregue a um colega para revisão e ajuste o que ele não entendeu.

EXERCÍCIOS

1. Qual parte do relatório é lida pela direção?

2. Por que mascarar dados sensíveis?

3. Cite dois itens de revisão final.

DESAFIO PABLOCYBERSEC

Escreva o resumo executivo do relatório do seu projeto em até 200 palavras.

Correção: *1) O sumário executivo. 2) Para proteger as informações do cliente. 3) Evidências para cada achado e coerência da severidade, por exemplo.*

CAPÍTULO 77 · DESAFIO FINAL PABLOCYBERSEC

Checklist do Pentester

APRENDA

Este checklist final resume o ciclo completo do trabalho. Use-o como lista de verificação em cada projeto, em laboratório ou em contratos reais. Marque cada item e, ao final, revise o que pode ser melhorado na próxima vez.

Antes: autorização assinada; escopo e regras definidos; contatos de emergência; janela de testes; backups e snapshots; ambiente e ferramentas atualizados; local seguro para evidências. Durante: respeitar o escopo; registrar cada ação; usar o mínimo impacto; interromper e avisar em caso de instabilidade; proteger dados encontrados.

Depois: reverter alterações e remover contas e arquivos criados; consolidar evidências; classificar achados; escrever recomendações práticas; entregar o relatório em canal seguro; apresentar os resultados; reteste após correções; destruir ou arquivar os dados conforme contrato.

Parabéns por chegar até aqui. A cibersegurança é uma jornada contínua: tecnologias mudam, ameaças evoluem e o aprendizado nunca termina. Mantenha o laboratório ativo, a ética em primeiro lugar e a curiosidade acesa.

NA PRÁTICA

Mantra do profissional: autorização, escopo, mínimo impacto, evidência, clareza e correção.

LABORATÓRIO

Monte a sua versão pessoal do checklist em uma página, adaptada ao seu estilo de trabalho, e use-a em seu próximo laboratório.

EXERCÍCIOS

1. Quais itens devem ser verificados antes do teste?

2. O que fazer se o alvo ficar instável durante o teste?

3. O que se faz com os dados ao final?

DESAFIO PABLOCYBERSEC

Planeje seus próximos 90 dias de estudo: metas semanais, laboratórios e certificações que deseja conquistar.

Correção: *1) Autorização, escopo, contatos, backups e ambiente. 2) Interromper e avisar o contato responsável. 3) Destruir ou arquivar conforme o contrato.*

Glossário

Ameaça	Algo ou alguém com potencial de causar dano a um ativo.
API	Interface que permite a programas conversarem entre si.
Ativo	Qualquer coisa de valor para a organização: dados, sistemas, pessoas, equipamentos.
Banner	Texto de identificação que um serviço exibe ao receber uma conexão.
Blue Team	Equipe de defesa: monitora, detecta e responde a incidentes.
Botnet	Rede de computadores infectados e controlados remotamente.
CIA	Confidencialidade, Integridade e Disponibilidade, a tríade da segurança.
CIDR	Notação que indica a parte de rede de um IP, como /24.
CVE	Identificador público de uma vulnerabilidade conhecida.
CVSS	Pontuação de 0 a 10 da severidade de uma vulnerabilidade.
DNS	Sistema que traduz nomes de domínio em endereços IP.
DoS / DDoS	Ataque que torna um serviço indisponível por sobrecarga.
Enumeração	Coleta detalhada de informações sobre serviços e recursos de um alvo.
Exploit	Técnica ou código que aproveita uma vulnerabilidade.
Firewall	Sistema que filtra o tráfego de rede conforme regras.
Hash	Resumo de tamanho fixo gerado a partir de um dado, sem volta.
IDOR	Acesso a objetos de outros usuários por troca de um identificador.
IoC	Indicador de comprometimento, como hash, IP ou domínio malicioso.
LGPD	Lei Geral de Proteção de Dados Pessoais (Lei 13.709/2018).
MFA	Autenticação com dois ou mais fatores diferentes.

Malware	Software criado para causar dano ou obter vantagem indevida.
NAT	Tradução de endereços IP privados em públicos.
OSINT	Inteligência obtida de fontes abertas e públicas.
OWASP	Comunidade que publica materiais gratuitos de segurança de aplicações.
Payload	Carga útil: a parte de um ataque que executa a ação desejada.
Pentest	Teste de intrusão autorizado para encontrar e comprovar falhas.
Phishing	Mensagem falsa que tenta enganar a vítima para obter dados ou ações.
Porta	Número que identifica um serviço dentro de um host.
Ransomware	Malware que cifra dados e exige pagamento para liberá-los.
Red Team	Equipe que simula adversários reais para testar as defesas.
Risco	Combinação de probabilidade e impacto de uma ameaça explorar uma vulnerabilidade.
Root	Usuário administrador com poder total em sistemas Linux.
Sal (salt)	Valor aleatório adicionado à senha antes do hash.
Scanning	Varredura ativa de hosts e portas.
SIEM	Plataforma que reúne e correlaciona logs de segurança.
SQL Injection	Falha em que a entrada do usuário altera a consulta ao banco.
TCP	Protocolo de transporte confiável, orientado à conexão.
TLS	Protocolo que cifra a comunicação, base do HTTPS.
UDP	Protocolo de transporte rápido e sem confirmação.
Vulnerabilidade	Fraqueza que pode ser explorada para comprometer a segurança.
WAF	Firewall especializado em proteger aplicações web.
XSS	Falha que permite executar scripts no navegador de outras pessoas.

Recursos para continuar

Laboratórios e treino legal

TryHackMe, Hack The Box, OWASP Juice Shop, DVWA, Metasploitable, VulnHub, PortSwigger Web Security Academy (gratuita).

Referências

OWASP (Top 10, Cheat Sheets, Testing Guide), MITRE ATT&CK, NIST, NVD (banco de vulnerabilidades), Exploit-DB, documentação oficial do Nmap, do Wireshark e do Kali Linux.

Para seguir estudando

Certificações de entrada como CompTIA Security+ e eJPT; depois CEH, PNPT, OSCP e trilhas defensivas como CySA+ e BTL1. Escolha conforme o objetivo de carreira.

Hábitos que fazem a diferença

Estudar toda semana, manter o diário de laboratório, participar de comunidades, praticar em plataformas legais e escrever relatórios de tudo o que fizer.

Palavras finais

Você chegou ao fim da versão 1.0. Revise os capítulos, refaça os laboratórios e transforme o projeto final em um item do seu portfólio. A versão 2.0 expandirá este conteúdo sem abandonar a estrutura que você já conhece. Continue estudando, sempre com ética.

Sobre o autor

[Inclua aqui a biografia do autor: formação, experiência e a história da PabloCyberSec.]

Diário de Laboratório

Data: ____/____/_____ Laboratório / Capítulo: _____