INSTITUTO SUPERIOR DE TECNOLOGIAS AVANÇADAS

Exploração de vulnerabilidade do tipo RCE

José Luís Almeida

Dissertação para obtenção do Grau de Mestre em

INFORMÁTICA

Júri

Presidente: Professor Doutor Paulo André Reis Duarte Branco

Arguente: Professora Doutora Carla Sofia Rocha da Silva

Orientador: Professor Doutor Pedro Ramos dos Santos Brandão

Outubro, 2025

ISTEC

Instituto Superior de Tecnologias Avançadas

Campus Académico do Lumiar, Lisboa

Dissertação

Mestrado em Informática

Por José Luís Almeida

Dissertação de Mestrado apresentada para cumprimento dos requisitos necessários à obtenção do grau de mestre em Informática, realizada sob a orientação científica da Professor Doutor Pedro Ramos dos Santos Brandão.

Lisboa, 2025

Agradecimentos

Em primeiro lugar, expresso a minha sincera gratidão ao meu orientador, Professor Doutor Pedro Brandão pela sua paciência, persistência, pelo seu rigor e pela sua total disponibilidade. A sua sabedoria, as suas críticas construtivas e o seu incentivo foram fundamentais para concluir esta dissertação. Um mentor que levarei para o resto da vida.

Um especial agradecimento à minha esposa, Vanessa Lindeza, pelo apoio incondicional que eternamente reconhecerei, pelas suas sábias e assertivas palavras, pela sua compressão, pelo suporte familiar nos momentos mais desafiadores desta jornada.

Por fim agradeço ao ISTEC pela excelência do seu ensino.

RESUMO

A identificação de vulnerabilidades é um processo sistemático que pode combinar um conjunto de fontes públicas, ferramentas técnicas especializadas e metodologias de análise de resultados.

A evolução das vulnerabilidades, impulsionada pela heterogeneidade e complexidade das arquiteturas tecnológicas, tem-se revelado crescente e dinâmica, aliada a uma sofisticação das técnicas de exploração e ao aumento das superfícies de ataque.

Este trabalho tem como objetivo analisar técnicas de identificação e exploração de vulnerabilidades de *Remote Code Execution* (RCE), um dos riscos mais críticos em cibersegurança. A exploração de vulnerabilidades assume-se como um pilar fundamental na investigação em cibersegurança. O estudo combina análise teórica com experimentação prática em ambiente laboratorial controlado, seguindo princípios éticos de segurança ofensiva.

Palavras-chave: análise de vulnerabilidades, vetores de ataque, exploração, segurança ofensiva.

ABSTRACT

Vulnerability identification is a systematic process that integrates public sources, specialized technical tools, and analytical methodologies.

The evolution of vulnerabilities, driven by the heterogeneity and complexity of modern technological architectures, has become increasingly dynamic, marked by sophisticated exploitation techniques and expanding attack surfaces.

This study aims to analyze techniques for identifying and exploiting Remote Code Execution (RCE) vulnerabilities, one of the most critical risks in cybersecurity. Vulnerability exploitation serves as a foundational pillar in cybersecurity research. The work combines theoretical analysis with controlled lab-based experimentation, adhering to ethical principles of offensive security.

Keywords: vulnerability analysis, attack vectors, exploitation, offensive security.

LISTA DE ABREVIATURAS E ACRÓNIMOS

API - Application Programming Interface
CERT - Computer Emergency Response Team
CISA – Cybersecurity and Infrastructure Security Agency
CMS - Content Management Systems
CVSS - Common Vulnerability Scoring System
DBIR - Data Breach Investigations Report
DHCP – Dynamic Host Configuration Protocol
EAAS - Exploitation As a Service
EDR – Endpoint Detection and Response
GDPR – General Data Protection Regulation
GET/POST - Método HTTP
HTML – HyperText Markup Language
HTTP – Hypertext Transfer Protocol
HTTPS – Hypertext Transfer Protocol Secure
LLM - Large Language Model
MAC – Media Access Control
MEC - Multi-access Edge Computing
MTTC – Mean Time To Contain
MTTD – Mean Time To Detect
NVD - National Vulnerability Database
IAC – Infrastructure as Code
IDS – Intrusion Detection System
IEC - International Electrotechnical Commission
IMAP – Internet Message Access Protocol
IMAPS – Internet Message Access Protocol Secure
IOT – Internet Of Things
IP – Internet Protocol
IPS – Intrusion Prevention System
ISC – Internet Systems Consortium
ISO – International Organization for Standardization
OAUTH – Open Authorization
OSINT - Open-Source Intelligence
OWASP - Open Web Application Security Project

PTES - Penetration Testing Execution Standard

RCE - Remote Code Execution

RDP - Remote Desktop Protocol

REST – Representational State Transfer

RPC - Remote Procedure Call

SAAS – Software as a Service

SIEM - Security Information and Event Management

SMB - Server Message Block

SMTP – Simple Mail Transfer Protocol

SOAP – Simple Object Access Protocol

TCP – Transmission Control Protocol

TTP – Tactics, Techniques, and Procedures

UDP – User Datagram Protocol

URL – Uniform Resource Locator

WAF - Web Application Firewall

WAN – Wide Area Network

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

ÍNDICE GERAL

1 INTRODUÇÃO

Num contexto de acelerada transformação digital e crescente interdependência sistémica, a cibersegurança afirma-se como âncora vital na proteção de ecossistemas tecnológicos, ativos informacionais e infraestruturas críticas [1]. A progressiva complexificação, heterogeneização das arquiteturas computacionais, aliadas à exponencial proliferação de serviços digitalmente mediados, têm catalisado o surgimento de vetores de ameaça de sofisticação inédita [2], capazes de explorar vulnerabilidades sistémicas com impactos potencialmente catastróficos. Entre estas vulnerabilidades, destaca-se a de execução remota de código (Remote Code Execution - RCE) [3], particularmente crítica por permitir que agentes maliciosos executem código arbitrário em sistemas-alvo de forma remota e não autorizada.

Este fenómeno tem ganho crescente relevo tanto na literatura académica especializada como nos relatórios técnicos das principais organizações internacionais de cibersegurança [5], consolidando-se como vetor recorrente em incidentes de segurança de grande magnitude. A sua capacidade de comprometer servidores à distância, manipular conjuntos de dados críticos, exfiltrar informação sensível ou estabelecer acessos persistentes sem deteção imediata [6] transforma a exploração desta classe de vulnerabilidades num risco sistémico para organizações dos setores público e privado, com implicações operacionais, financeiras e reputacionais potencialmente devastadoras [7].

As aplicações web emergiram como principal vetor de ataques digitais, representando um dos desafios mais prementes para organizações de todos os setores [8]. Esta predominância resulta de uma conjugação de fatores críticos: a inevitável exposição pública inerente à natureza da Internet [9], a crescente complexidade arquitetural das soluções web modernas [10] - que integram desde interfaces de entrada até sistemas de suporte de gestão de dados, passando por múltiplas bibliotecas externas [11] - e as vulnerabilidades humanas e de configuração que persistem em ambientes digitais [12]. A sofisticação dos ataques tem evoluído em paralelo com as próprias tecnologias [13],

com agentes maliciosos a recorrerem a ferramentas altamente especializadas como o SQLmap para exploração de vulnerabilidades [14], Shodan ou a Burp Suite para análise abrangente de aplicações [15], o que lhes permite identificar fragilidades com maior eficiência e conduzir explorações direcionadas. Esta situação agrava-se devido à frequente deteção de configurações básicas comprometidas, como credenciais por defeito ou mecanismos de segurança mal implementados [16], que funcionam como portas de entrada para acessos não autorizados.

No âmbito deste panorama, as vulnerabilidades de *Remote Code Execution* (RCE) destacam-se pela sua gravidade excecional [17]. O perigo do RCE reside na sua capacidade única de permitir a execução remota de comandos arbitrários no servidor-alvo, frequentemente com privilégios elevados [18]. As consequências desta exploração são invariavelmente severas, incluindo o comprometimento total do sistema atacado, exfiltração massiva de dados sensíveis, movimentação lateral dentro de redes corporativas e instalação persistente de *backdoors* ou *ransomware* [19].

Perante este paradigma de ameaças, a presente dissertação propõe um exame sistemático dos requisitos técnicos necessários para a exploração de vulnerabilidades RCE [20], combinando perspetivas teóricas e empíricas. A abordagem metodológica assenta em dois pilares fundamentais [21]: uma análise conceptual multidisciplinar que abrange dimensões técnicas, teóricas e normativas [22], e uma validação experimental através da construção de um ambiente controlado de testes [23] que incorpora aplicações com vulnerabilidades conhecidas. Esta investigação visa duplamente [24] aprofundar a compreensão dos mecanismos subjacentes à exploração de vulnerabilidades [25] e fomentar uma consciência crítica acerca da necessidade imperativa de adoção de boas práticas no desenvolvimento, configuração e monitorização de sistemas informáticos [26]. O estudo seguirá metodologias consagradas no âmbito da segurança ofensiva [27], observando escrupulosamente os princípios éticos e enquadramentos legais aplicáveis [28], e recorrerá a ferramentas padronizadas no domínio dos testes de penetração [29], com especial destaque para o Metasploit Framework [30], entre outras soluções tecnologicamente validadas pela comunidade técnica.

A questão central que esta dissertação pretende responder é: será possível detetar ameaças do tipo RCE através da metodologia PTES (Penetration Testing Execution Standard)? [31]

2 CONCEITO DE REMOTE CODE EXECUTION (RCE)

As vulnerabilidades de execução remota de código, denominadas por *Remote Code Execution* (RCE), constituem uma das ameaças mais significativas no atual panorama da cibersegurança [31]. Como demonstrado por Brandão [32], estas vulnerabilidades representam um risco sistémico para infraestruturas críticas, sendo a sua gravidade amplificada pela possibilidade de um atacante, na maioria dos casos de forma remota e não autenticada, executar comandos arbitrários em sistemas-alvo através da exploração de falhas em aplicações ou serviços expostos na Internet. Este tipo de exploração compromete não apenas a integridade do sistema afetado, como pode também permitir movimentação lateral na rede, acesso a outros componentes da infraestrutura, exfiltração de dados sensíveis ou instalação de *malware* persistente [33].

O crescente interesse na temática das vulnerabilidades RCE é evidenciado pelo número de incidentes reportados por entidades como o CERT e pela relevância deste vetor de ataque em relatórios de referência, como o Verizon Data Breach Investigations Report (DBIR), que indica que 25% dos ataques a aplicações web registados em 2023 envolveram execução remota de código [34]. Estes dados sublinham a importância de compreender os mecanismos técnicos subjacentes a estes ataques, bem como a necessidade de implementar políticas de proteção robustas.

Tipicamente, uma vulnerabilidade do tipo RCE permite a um atacante manipular a execução normal de uma aplicação, injetando e executando comandos através de entradas não validadas ou com validações inadequadas [35]. Estas vulnerabilidades podem manifestar-se em diversos contextos, nomeadamente:

- Falhas em *frameworks* web [36]
- Bibliotecas de parsing com proteção insuficiente [37]

- Serviços expostos com validações superficiais [38]
- Configurações incorretas em aplicações do lado do servidor [39]

Este capítulo tem como objetivo apresentar o conhecimento teórico fundamental sobre a natureza das vulnerabilidades RCE, os seus principais vetores de exploração, ferramentas utilizadas pelos atacantes, bem como as medidas de mitigação e proteção adequadas. Aborda-se ainda o enquadramento ético e técnico das práticas de teste ofensivo (*ethical hacking*). O desenvolvimento aprofundado destes aspetos proporciona o suporte conceptual essencial para a análise prática que será desenvolvida no capítulo seguinte.

2.1 CLASSIFICAÇÃO DE VULNERABILIDADES DO TIPO RCE

As vulnerabilidades de execução remota de código (RCE) não se limitam a um único tipo de falha, podendo ser causadas por diversos fatores técnicos e lógicos. De acordo com a OWASP (*Open Web Application Security Project*), estas vulnerabilidades ocorrem tipicamente devido a falhas na validação de entradas, execução dinâmica de código e gestão inadequada de dependências externas [40]. Nas aplicações web, os casos mais comuns incluem a utilização de funções como eval(), exec(), popen() ou chamadas de sistema que executam diretamente comandos com base em dados fornecidos pelo utilizador sem a devida validação [41].

Entre os cenários mais frequentes destacam-se as vulnerabilidades em *Content Management Systems* (CMS) como WordPress, Joomla e Drupal, onde extensões ou plugins desatualizados contêm frequentemente falhas exploráveis através de scripts de execução remota [42]. A utilização de bibliotecas sem as atualizações de segurança mais recentes ou a configuração inadequada de serviços podem igualmente proporcionar vetores de ataque para exploração RCE, conforme documentado em diversas publicações técnicas especializadas [43]-[45].

Estas vulnerabilidades podem ainda ser exploradas através de falhas em protocolos como SMB, RPC ou RDP, bem como devido à ausência de mecanismos eficazes de

sandboxing e isolamento de processos em arquiteturas mais antigas [46]. A diversidade de vetores de ataque possíveis exige uma abordagem abrangente para a sua mitigação, que deve incluir desde a revisão rigorosa do código até à implementação de segmentação de rede e políticas estritas de privilégio mínimo [47].

2.2 VETORES DE ATAQUE E FASES DE EXPLORAÇÃO NUM CONTEXTO DO TIPO RCE

O ciclo de exploração de uma vulnerabilidade de execução remota de código (RCE) pode ser sistematicamente descrito em quatro fases fundamentais: reconhecimento, enumeração, exploração e pós-exploração. Este modelo encontra-se alinhado com as metodologias de teste de penetração utilizadas em segurança ofensiva, conforme estabelecido por Anderson [48] e consagrado em *frameworks* amplamente reconhecidos como o *Penetration Testing Execution Standard* (PTES) [49].

Esta abordagem faseada reflete as melhores práticas da segurança ofensiva, permitindo não só uma compreensão aprofundada dos mecanismos de ataque, mas também o desenvolvimento de estratégias de mitigação mais eficazes. Como destacam os estudos recentes, a adoção de metodologias estruturadas como o PTES [50] tem-se revelado fundamental para uma avaliação abrangente dos riscos associados a vulnerabilidades RCE em ambientes empresariais e de infraestruturas críticas [51].

2.3 RECONHECIMENTO

A etapa de reconhecimento é considerada crítica e determinante para o sucesso das etapas seguintes. Consiste na recolha de informações passivas e ativas sobre o alvo, sem que exista uma interação direta com os seus serviços. Como demonstrado em diversos estudos [52], esta etapa preliminar é fundamental para estabelecer o perfil de segurança do sistema em análise. Ferramentas especializadas como o Shodan, Censys ou

DNSdumpster permitem identificar sistemas expostos na Internet, aplicações em utilização e potenciais vetores de ataque [53], [54].

A metodologia de reconhecimento envolve tipicamente:

1. Análise de infraestruturas expostas através de motores de busca especializados
2. Mapeamento de serviços e portas abertas
3. Identificação de tecnologias e versões em utilização
4. Recolha de meta dados e informações técnicas

Esta fase assume particular importância na medida em que permite restringir o âmbito de atuação e selecionar alvos com maior probabilidade de apresentarem vulnerabilidades [55]. Para além das ferramentas automatizadas, a análise de registos públicos, o reconhecimento de domínios e a investigação em redes sociais corporativas (como o LinkedIn) podem fornecer dados valiosos sobre:

- A arquitetura da infraestrutura alvo
- Fornecedores e tecnologias adotadas
- Perfis de utilizadores e estruturas organizacionais
- Relações entre diferentes sistemas e entidades [56]

Como destacam os estudos mais recentes [57], a fase de reconhecimento bem executada pode representar até 70% do sucesso de uma avaliação de segurança, permitindo identificar pontos fracos sem alertar os sistemas de deteção. A abordagem deve ser sistemática e abrangente, cobrindo tanto as componentes técnicas como as humanas da organização alvo.

2.3.1 EXEMPLO DE RECONHECIMENTO EM CÉNARIO REAL

Uma vez estabelecida a motivação do ataque - seja para resgate, espionagem, roubo de informação ou interrupção de serviços - inicia-se um processo meticuloso de recolha de

dados através de técnicas não intrusivas. Esta abordagem permite evitar a deteção pelos sistemas de proteção do alvo, mantendo a descrição da operação.

A investigação assenta na exploração estratégica de fontes públicas de informação disponíveis na Internet, com particular foco na análise de:

1. **Estrutura de domínios e subdomínios**
2. **Registos DNS públicos**
3. **Registos históricos de resolução de nomes**
4. **Ficheiros de configuração expostos**

A análise sistemática da infraestrutura de domínios revela frequentemente vetores de ataque alternativos e menos protegidos. Por exemplo, em vez de direcionar esforços para o domínio principal (www.dominio.pt), a investigação pode identificar:

- Interfaces de administração de serviços (mail.dominio.pt)
- Subdomínios de desenvolvimento (dev.dominio.pt)
- Portais de parceiros (extranet.dominio.pt)
- APIs expostas (api.dominio.pt)

Esta abordagem permite descobrir componentes menos visíveis da infraestrutura digital da organização, que frequentemente apresentam medidas de segurança menos robustas do que os sistemas principais. A identificação destes elementos secundários amplia significativamente a superfície de ataque potencial, oferecendo vetores alternativos para tentativas de intrusão.

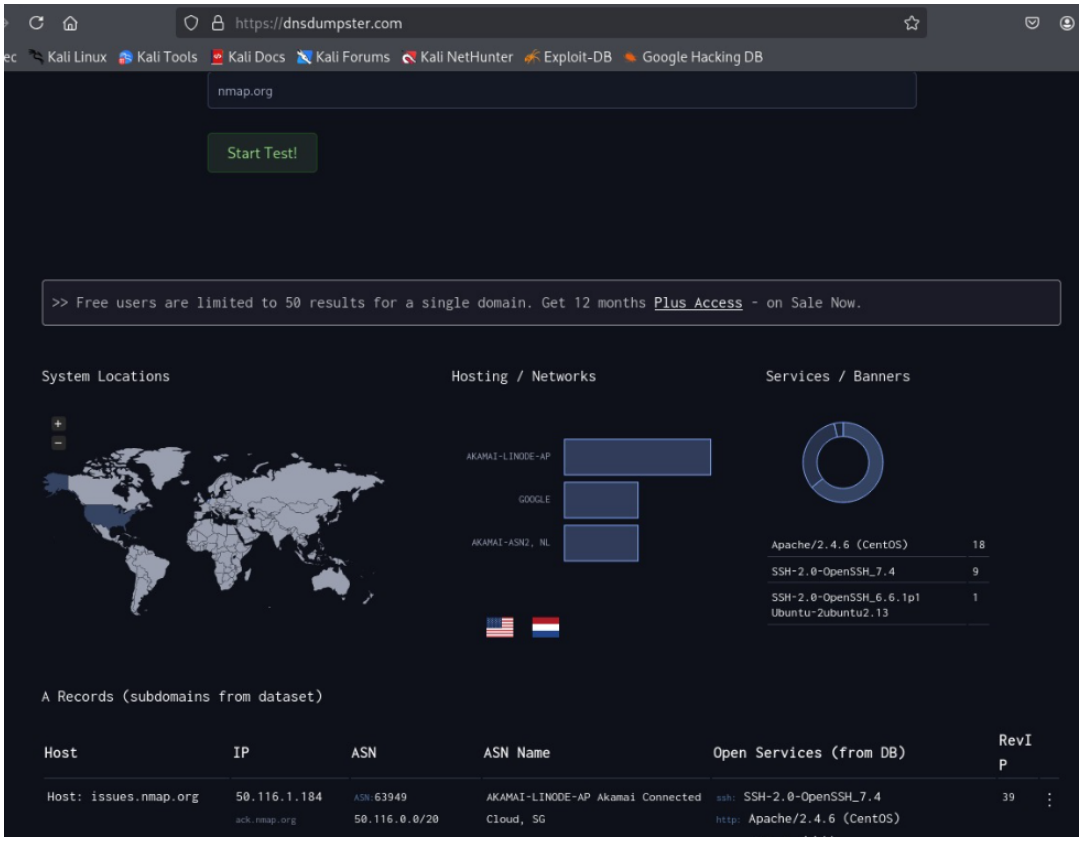

Figura 1 - Investigação por domínio na plataforma dnsdumpster

A plataforma DNSDumpster.com (Figura 1) revela-se um instrumento valioso para o reconhecimento de infraestruturas digitais, como demonstrado na análise efetuada para o domínio nmap.org. Esta ferramenta proporciona uma caracterização abrangente da arquitetura de rede, fornecendo:

1. **Mapeamento completo da estrutura de domínios**:

    - Domínio principal (nmap.org)
    - Subdomínios associados
    - Registos DNS históricos

2. **Informação técnica detalhada**:

    - Geolocalização física dos servidores

- Portas de comunicação expostas
- Versões de serviços de rede em execução
- Conjuntos de endereços IP vinculados ao domínio

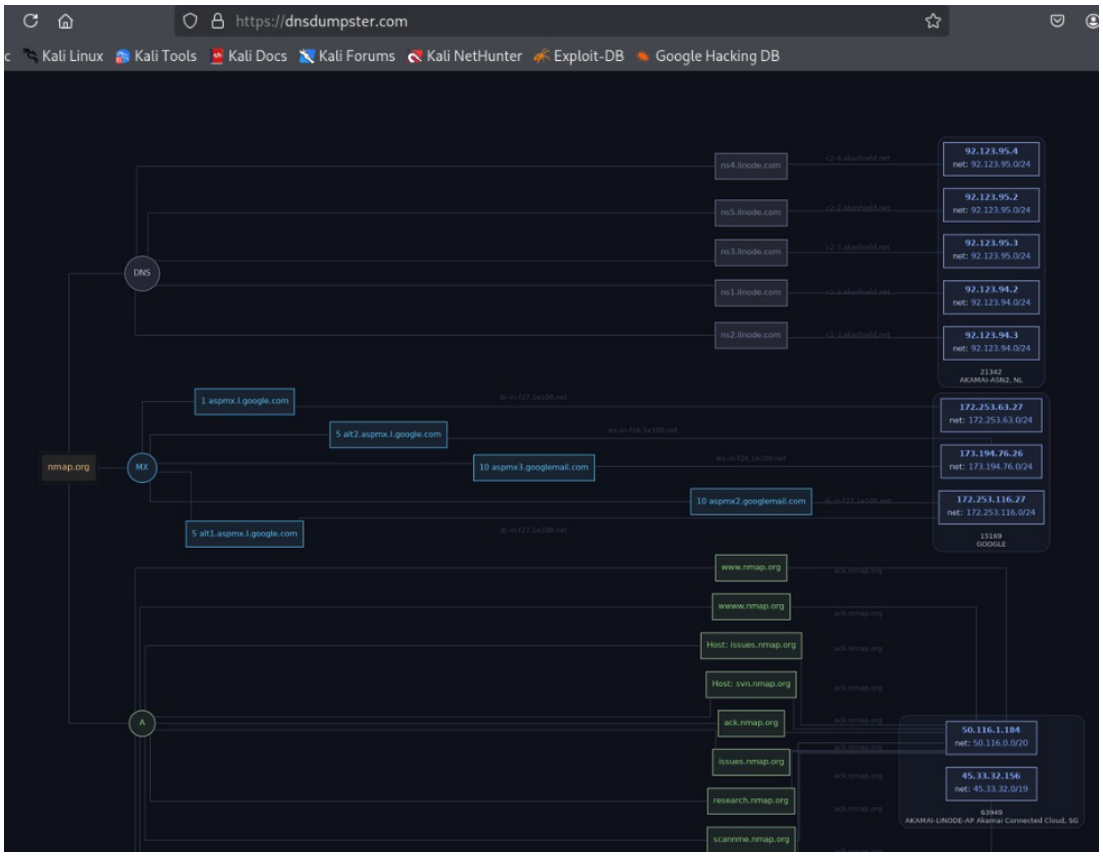

Figura 2 - Diagrama de rede gerado pelo dnsdumpster

Como ilustrado na Figura 2, a ferramenta gera automaticamente um diagrama de rede que estabelece visualmente:

- Relações hierárquicas entre nomes de domínio e tipos de registo DNS
- Fluxos de comunicação entre diferentes serviços
- Mapeamento entre nomes de domínio e endereços IP específicos
- Interligações entre diversos elementos da infraestrutura

Esta representação gráfica permite uma compreensão imediata da topologia da rede, identificando potenciais pontos de entrada vulneráveis e relações críticas entre componentes. A análise destes diagramas revela-se particularmente útil para:

1. Identificar ativos não documentados
2. Detetar configurações potencialmente perigosas
3. Mapear a arquitetura global dos sistemas
4. Planear vetores de ataque específicos

A riqueza informativa proporcionada por esta ferramenta sublinha a importância do reconhecimento passivo na fase inicial de avaliação de segurança, demonstrando como informação publicamente acessível pode ser utilizada para construir um perfil detalhado da infraestrutura alvo.

As redes sociais profissionais, em particular o LinkedIn, constituem também uma fonte valiosa de informação para potenciais cibe criminosos, permitindo-lhes recolher dados sensíveis através da análise de perfis públicos de colaboradores e consultores externos. Esta exposição de informação facilita a identificação de tecnologias específicas, componentes de infraestrutura e até mesmo parcerias estratégicas da organização-alvo. Muitos utilizadores partilham, de forma inadvertida, estudos de caso detalhados que incluem referências explícitas a clientes, projetos implementados e soluções tecnológicas adotadas. Estas publicações, aparentemente inócuas, podem revelar dados críticos sobre a arquitetura de sistemas internos, práticas de desenvolvimento ou ferramentas utilizadas pela organização. Quando analisadas de forma sistemática, estas informações permitem aos cibe criminosos:

1. **Mapear o ecossistema tecnológico** da organização, identificando aplicações, versões e *frameworks* em uso;
2. **Reconhecer parceiros e fornecedores**, o que pode revelar vulnerabilidades na cadeia de abastecimento;
3. **Identificar colaboradores-chave** com acesso a sistemas críticos, potenciais alvos de engenharia social;
4. **Detetar projetos recentes** que possam ter exposto novas infraestruturas ou interfaces de administração.

Esta recolha passiva de inteligência, sem qualquer interação direta com os sistemas da organização, demonstra como a partilha excessiva de informação em redes sociais pode ampliar significativamente a superfície de ataque. A natureza pública e acessível destes dados transforma plataformas como o LinkedIn em ferramentas eficazes para o reconhecimento pré-ataque, sublinhando a necessidade de políticas rigorosas de sensibilização e controlo da informação partilhada pelos colaboradores.

A exposição involuntária de detalhes técnicos ou organizacionais nestas plataformas pode, assim, facilitar a fase de reconhecimento de um ciberataque, permitindo que agentes maliciosos desenvolvam estratégias de intrusão altamente direcionadas e com maior probabilidade de sucesso.

Mesmo sem um alvo específico definido, atacantes podem utilizar plataformas agregadoras de informação como Shodan para:

- Identificar sistemas vulneráveis em escala global
- Filtrar por tecnologias específicas (ex: Apache/IIS desatualizados)
- Cruzar dados com vulnerabilidades conhecidas (CVEs)
- Localizar *exploits* públicos correspondentes

Conforme demonstrado na Figura 3, a plataforma Shodan indexa mais de 17 milhões de servidores web ativos, oferecendo funcionalidades avançadas:

Para complementar a recolha passiva de informação, podem ser utilizados métodos de baixa visibilidade:

- Procura por portas especificas (TCP 80/443/22)
- Análise de *banners* de serviço
- Testes de conectividade limitados
- Identificação de serviços de computação em nuvem expostos

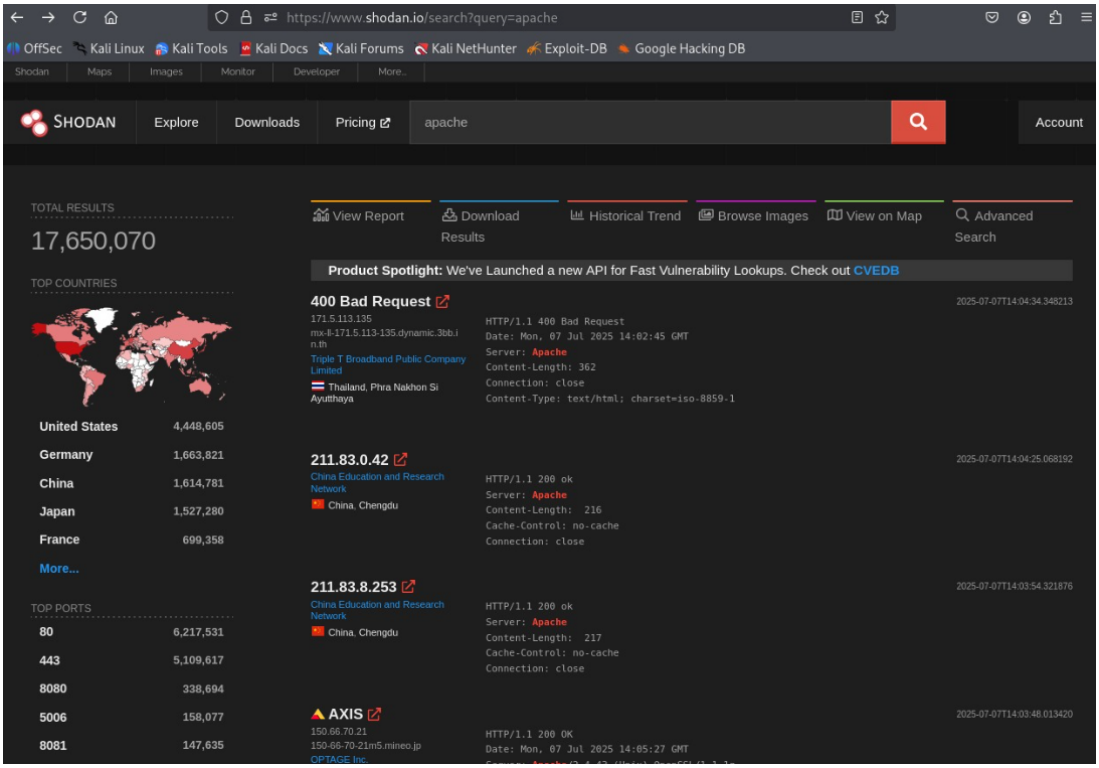

Figura 3 - Exemplo de investigação na plataforma Shodan

Estas técnicas permitem validar informações recolhidas sem desencadear mecanismos de alerta, mantendo o sigilo da operação de reconhecimento. A combinação de fontes abertas com ferramentas especializadas cria um círculo virtuoso de inteligência que sustenta operações ofensivas posteriores.

2.4 ENUMERAÇÃO

Na fase de enumeração, o cibe criminoso estabelece interações controladas com os sistemas previamente identificados, com o objetivo de recolher informações detalhadas sobre serviços ativos, portas abertas, *banners* de sistema, estruturas de diretórios e outros elementos que possam revelar vulnerabilidades exploráveis [58]. Esta etapa caracteriza-se por uma abordagem mais intrusiva do que o reconhecimento inicial, embora mantenha um perfil discreto para evitar deteção.

Como documentado na literatura especializada [59], as ferramentas mais utilizadas nesta fase incluem:

- **Nmap**: Para mapeamento avançado de redes e serviços
- **Dirb**: Para identificação de diretórios e ficheiros acessíveis
- **Nikto**: Para deteção de vulnerabilidades em servidores web
- **WhatWeb**: Para análise detalhada de tecnologias web implementadas

Estudos recentes demonstram que é frequente identificar durante esta fase [60]:

1. *Endpoints* REST expostos sem controlo de acesso adequado
2. Interfaces de administração com mecanismos de autenticação fracos ou inexistentes
3. Ficheiros de configuração (ex: .env, config.php) acessíveis publicamente
4. Versões desatualizadas de software com vulnerabilidades conhecidas

A metodologia de enumeração deve seguir um processo sistemático [61]:

1. Identificação de serviços ativos e respetivas versões
2. Análise de configurações e permissões
3. Mapeamento da arquitetura da aplicação
4. Deteção de componentes vulneráveis

Como destacam Pereira et al. [62], a fase de enumeração bem executada permite construir um perfil detalhado do sistema alvo, identificando potenciais vetores de ataque para exploração posterior. A informação recolhida nesta etapa é fundamental para priorizar os esforços de exploração e aumentar a probabilidade de sucesso num eventual ataque.

2.5 EXPLORAÇÃO

A fase de exploração consiste na utilização efetiva de vulnerabilidades identificadas para executar código remoto no sistema alvo. No contexto de RCE, esta execução pode ocorrer através de diversos vetores [63]:

1. Injeção de comandos através da manipulação de parâmetros em URLs
2. Carregamento de ficheiros maliciosos que permitam execução remota
3. Exploração de vulnerabilidades em APIs REST ou SOAP
4. Abuso de bibliotecas vulneráveis (como no caso do Log4Shell, CVE-2021-44228) [64]-[66]

O Metasploit Framework destaca-se como uma das ferramentas mais utilizadas nesta fase [67], oferecendo funcionalidades avançadas para:

- Carregamento de *exploits* conhecidos
- Adaptação de *payloads* específicos
- Execução controlada de testes de penetração

Após o comprometimento bem-sucedido do sistema, o atacante pode implementar *payloads* sofisticados como o Meterpreter, que estabelece uma ligação interativa do tipo terminal diretamente em memória [68]. Este tipo de *payload* oferece capacidades avançadas, incluindo:

- Execução de comandos remotos
- Movimentação lateral na rede
- Mecanismos de evasão a sistemas de deteção
- Persistência no sistema comprometido

Como demonstrado nos estudos mais recentes [69], a fase de exploração requer um conhecimento profundo tanto das vulnerabilidades específicas como das arquiteturas

dos sistemas alvo. A seleção adequada de *exploits* e *payloads* é crucial para o sucesso da operação, sendo necessário considerar fatores como:

- A versão exata do software vulnerável
- O sistema operativo subjacente
- As medidas de segurança implementadas
- Os mecanismos de deteção ativos

2.6 PÓS-EXPLORAÇÃO

Após a obtenção bem-sucedida de acesso inicial, a fase de pós-exploração concentra-se em três objetivos principais: elevação de privilégios, movimentação lateral na rede e estabelecimento de persistência no sistema comprometido [70]. Esta etapa representa a materialização do impacto real das vulnerabilidades RCE, uma vez que o atacante transita de uma exploração pontual para um comprometimento abrangente da infraestrutura organizacional [71].

As ações típicas desta fase incluem:

1. **Exfiltração de credenciais** através de técnicas como *dumping* de memória ou *keylogging*
2. **Mapeamento detalhado da rede interna** para identificar alvos secundários
3. **Criação de *backdoors*** para garantir acesso futuro
4. **Manipulação de *logs*** para apagar evidências da intrusão [72]

A persistência é normalmente alcançada através de diversos mecanismos [73]:

- Implementação de *web shells* em servidores comprometidos
- Agendamento de tarefas maliciosas (através da cron no Linux ou programador de tarefas no Windows)
- Modificação de serviços existentes ou criação de novos serviços maliciosos

- Alteração de chaves de registo em sistemas Windows

Como destacam os estudos mais recentes [74], muitas destas atividades podem permanecer in detetadas durante períodos prolongados, especialmente em ambientes sem:

1. Monitorização ativa de integridade de sistemas
2. Correlação eficaz de eventos de segurança
3. Análise comportamental de tráfego de rede
4. Mecanismos de deteção de anomalias

O impacto operacional desta fase é particularmente grave, pois permite ao atacante [75]:

- Manter acesso prolongado à infraestrutura
- Escalar privilégios para domínio completo
- Exfiltrar dados sensíveis de forma contínua
- Preparar o terreno para ataques mais avançados

2.7 FERRAMENTAS E TÉCNICAS DE APOIO À EXPLORAÇÃO DE RCE

A eficácia dos ataques de Execução Remota de Código (RCE) encontra-se intrinsecamente ligada à utilização combinada de ferramentas especializadas e técnicas de automatização que facilitam a identificação, exploração e manutenção de acessos em sistemas vulneráveis. Como demonstrado na literatura recente, estas ferramentas são igualmente adotadas por profissionais de segurança ética e por agentes maliciosos, sublinhando a importância do seu estudo e monitorização contínua.
Entre as ferramentas mais relevantes para a exploração de vulnerabilidades RCE destacam-se:

1. *Frameworks* de exploração abrangentes que oferecem módulos pré-configurados para exploração de vulnerabilidades conhecidas

2. *Scanners* de vulnerabilidades que permitem identificar sistemas vulneráveis a ataques RCE
3. Ferramentas de desenvolvimento de *exploits* que auxiliam na criação de código de exploração personalizado
4. Utilitários de pós-exploração que facilitam a movimentação lateral e manutenção de acessos

Paralelamente, as técnicas modernas de automatização e orquestração de ataques têm vindo a aumentar significativamente o potencial ofensivo das explorações RCE. Estas incluem:

- *Scripting* avançado para exploração em massa
- Técnicas de evasão a sistemas de deteção
- Mecanismos de persistência sofisticados
- Técnicas de ofuscação de código malicioso

A dualidade no uso destas ferramentas - tanto para fins defensivos como ofensivos - coloca aos profissionais de segurança o desafio de conhecer profundamente o seu funcionamento, de forma a desenvolver mecanismos de deteção e mitigação eficazes. Como evidenciam os estudos mais recentes, a compreensão detalhada destas ferramentas e técnicas constitui um elemento fundamental para uma estratégia abrangente de proteção contra explorações RCE.

2.7.1 FRAMEWORK METASPLOIT

O Metasploit Framework representa indiscutivelmente a ferramenta mais difundida e completa no domínio dos testes de penetração e exploração de vulnerabilidades, sendo amplamente utilizado tanto por profissionais de segurança informática como por agentes maliciosos [76]. Este ambiente modular integrado oferece um conjunto abrangente de funcionalidades que permitem desde a identificação de vulnerabilidades até à exploração avançada e manutenção de acessos persistentes em sistemas comprometidos [77].

A arquitetura do Metasploit baseia-se num princípio de modularidade e reutilização de componentes, onde os diversos *exploits* podem ser combinados com diferentes *payloads* conforme as necessidades específicas de cada cenário de ataque. Esta flexibilidade é particularmente evidente na família Meterpreter, cujos *payloads* se destacam pela sua operação exclusivamente em memória (*fileless*), dificultando significativamente a sua deteção por soluções antivírus convencionais [78]. Além disso, estes *payloads* oferecem funcionalidades avançadas de pós-exploração, incluindo movimentação lateral na rede, elevação de privilégios e exfiltração de dados, tudo isto através de comunicações encriptadas que visam evitar a deteção [79].

Uma das características mais relevantes do Metasploit é a sua integração nativa com bases de dados de vulnerabilidades como o *Common Vulnerabilities and Exposures* (CVE) e o *Exploit Database* (Exploit-DB). Esta integração permite não só a automatização da seleção de vetores de ataque com base em vulnerabilidades conhecidas, mas também a constante atualização da sua base de conhecimentos face às novas ameaças que vão sendo identificadas [80]. Na prática, esta capacidade transforma o Metasploit numa ferramenta dinâmica que evolui em paralelo com o panorama de ameaças, mantendo-se relevante mesmo face às contramedidas de segurança mais recentes.

2.7.2 FERRAMENTAS PARA ANÁLISE DE APLICAÇÕES WEB

No contexto de ciberataques direcionados a aplicações web, o Burp Suite assume um papel central como ferramenta de análise de segurança [81]. A sua funcionalidade de proxy permite a captura, modificação e reencaminhamento de pedidos HTTP/HTTPS, possibilitando testar a resiliência da aplicação face a inputs maliciosos. Esta capacidade é particularmente relevante para identificar vulnerabilidades de execução remota de código (RCE), uma vez que permite avaliar o comportamento da aplicação perante diferentes vetores de ataque [82].

O Burp Suite integra ainda várias ferramentas especializadas que potenciam a sua eficácia:

- Intruder: Permite automatizar ataques através da modificação sistemática de parâmetros
- Repeater: Facilita a manipulação manual de requisições para testes precisos
- Scanner: Identifica automaticamente vulnerabilidades conhecidas [83]

Estas funcionalidades tornam o Burp Suite particularmente eficaz na deteção de vetores RCE em:

✓ Parâmetros GET/POST

✓ Cabeçalhos HTTP

✓ Estruturas de autenticação

✓ Mecanismos de serialização de dados [84]

Complementarmente ao Burp Suite, outras ferramentas especializadas desempenham papéis cruciais na análise de segurança web:

- OWASP ZAP: Oferece funcionalidades semelhantes com ênfase em testes automatizados
- SQLmap: Especializado na deteção e exploração de vulnerabilidades SQL Injection
- Nikto: Realiza scans abrangentes para identificar vulnerabilidades conhecidas
- Dirbuster: Efetua enumeração de diretórios e ficheiros em servidores web [85]-[87]

Como destacam os estudos recentes [88], a utilização combinada destas ferramentas numa abordagem estruturada permite:

• Identificar um âmbito mais amplo de vulnerabilidades

• Correlacionar diferentes vetores de ataque

• Validar a eficácia de medidas de mitigação

• Reduzir falsos positivos através de verificação cruzada

2.8 CATÁLOGOS DE EXPLOITS E BASES DE VULNERABILIDADES

O sucesso na exploração de vulnerabilidades de Execução Remota de Código (RCE) está intimamente ligado à capacidade de identificar falhas em componentes utilizados pelas aplicações ou servidores alvo [89]. Para este efeito, diversas plataformas especializadas assumem um papel fundamental no processo de pesquisa e classificação de vulnerabilidades:

Entre os repositórios mais relevantes destacam-se:

- *Exploit Database* (Exploit-DB): Arquivo abrangente de *exploits* testados e documentados
- CVE (*Common Vulnerabilities and Exposures*): Sistema de identificação padronizada de vulnerabilidades
- NVD (*National Vulnerability Database*): Base de dados governamental dos EUA com análise de impacto
- SecurityFocus: Plataforma histórica de divulgação de vulnerabilidades [90]-[92]

Estes recursos permitem, por exemplo, verificar que uma versão específica do Apache Tomcat apresenta uma vulnerabilidade crítica que possibilita a execução remota de comandos, como foi o caso documentado no boletim CVE-2025-24813 [93]. Esta informação é crucial para:

1. Priorizar a aplicação de patches de segurança
2. Desenvolver mecanismos de deteção específicos
3. Planear testes de penetração direcionados
4. Implementar medidas compensatórias temporárias

Na prática, a consulta a estas bases de dados é frequentemente automatizada através de:

• Scripts de *web crawling* especializados

• Ferramentas como o SearchSploit

• Integrações com *frameworks* de segurança

• Sistemas de monitorização contínua de vulnerabilidades [94]

Como demonstram os estudos recentes [95], a utilização sistemática destes recursos:

- Reduz significativamente o tempo de identificação de vulnerabilidades
- Facilita a correlação entre diferentes falhas de segurança
- Permite a antecipação de vetores de ataque
- Apoia a tomada de decisões informadas em gestão de vulnerabilidades

2.9 SCRIPTS PERSONALIZADOS E AUTOMAÇÃO

Nos últimos anos, tem-se observado um aumento significativo na sofisticação dos ciberataques automatizados, com o desenvolvimento e partilha de *scripts* personalizados em Python, Bash ou PowerShell. Estes scripts são frequentemente disponibilizados em plataformas como fóruns *underground*, GitHub ou Pastebin [96], [97], sendo concebidos para explorar vulnerabilidades específicas em sistemas com configurações particulares. Muitas vezes, estes scripts são integrados com ferramentas consolidadas, como o Metasploit ou o Nmap, para aumentar a sua eficácia e alcance em cenários de ataque real [98].

Além dos scripts personalizados, ferramentas como Ansible, Pupy, Empire ou Covenant têm ganho relevância no contexto de operações avançadas de ciberataque. Estas soluções oferecem funcionalidades que vão além da simples exploração de vulnerabilidades, permitindo:

- Controlo remoto persistente de sistemas comprometidos
- Registo de atividade do utilizador (*keylogging*) para captura de credenciais
- Visualização gráfica da rede para melhor orientação em movimentação lateral
- Automatização de tarefas de pós-exploração, como escalamento de privilégios ou exfiltração de dados [99], [100].

Esta evolução na automação de ciberataques demonstra uma tendência clara para operações mais complexas e prolongadas, onde os cibe criminosos procuram manter acesso a sistemas comprometidos durante longos períodos, muitas vezes sem serem detetados. A capacidade de adaptar scripts e ferramentas a ambientes específicos torna estes vetores de ataque particularmente perigosos, exigindo uma resposta igualmente sofisticada por parte dos profissionais de segurança [101].

2.10 IMPACTO DA EXPLORAÇÃO DE RCE EM AMBIENTES CORPORATIVOS

A exploração de vulnerabilidades de Execução Remota de Código (RCE) tem sido responsável por incidentes de segurança de grande escala, com impactos que vão desde intrusões em cadeias de abastecimento (ataques à *supply chain*) até ao comprometimento total de infraestruturas críticas [102]. O caso paradigmático da Equifax (2017), resultante de uma vulnerabilidade crítica no Apache Struts 2, levou à exfiltração de dados de mais de 145 milhões de utilizadores e a prejuízos estimados em 1,4 mil milhões de dólares em sanções e custos legais [103]. Mais recentemente, em 2021, a vulnerabilidade Log4Shell (CVE-2021-44228) permitiu a execução remota de código em aplicações empresariais Java, afetando serviços em cloud, dispositivos IoT e plataformas SaaS [104]. Estudos longitudinais indicam que o tempo médio de deteção (MTTD) para vulnerabilidades RCE varia entre 180 e 230 dias, enquanto o tempo médio para contenção (MTTC) frequentemente excede os 30 dias [105], refletindo deficiências estruturais nos processos de gestão de patches, *threat hunting* e resposta a incidentes.

Tabela 1 - Tipologia de Danos

Dimensão	**Exemplo de Incidente**	**Consequências**
Financeira	Exploração do MOVEit Transfer (2023)	Perdas de 10 mil milhões USD [106]
		Custos de recuperação [107]
Reputacional	Ataque ao Microsoft Exchange (2024)	Queda de 18% na confiança [108]
		Danos à imagem corporativa [109]
Legal	Vulnerabilidade em Ivanti VPN (2024)	Multa GDPR de 20 milhões € [110]

Dimensão	Exemplo de Incidente	Consequências
		Ações judiciais [111]
Operacional	Exploração do PAN-OS (2024)	Interrupção de 72h em serviços críticos [112]
		Paralisação de centro de dados [113]

A análise dos incidentes cibernéticos apresentados na tabela 1 revela um padrão preocupante de impactos sistémicos que transcendem a esfera técnica, afetando estruturalmente as organizações em múltiplas dimensões:

1. **Dimensão Financeira**: A exploração da vulnerabilidade no MOVEit Transfer (2023) exemplifica a magnitude económica deste tipo de incidentes, com perdas diretas estimadas em 10 mil milhões de dólares, para além dos custos adicionais de recuperação de sistemas e mitigação de danos [106], [107]. Estes valores sublinham a necessidade de investimentos preventivos em segurança da informação.

2. **Dimensão Reputacional**: O ataque ao Microsoft Exchange (2024) resultou numa queda mensurável de 18% na confiança dos stakeholders e em danos prolongados à perceção da marca [108], [109]. Estes efeitos demonstram que o capital intangível das organizações se tornou tão vulnerável quanto as suas infraestruturas tecnológicas.

3. **Dimensão Legal-Regulatória**: O caso da vulnerabilidade na Ivanti VPN (2024) ilustra a crescente severidade do ambiente normativo, com a imposição de uma multa de 20 milhões de euros ao abrigo do GDPR [110]. Este episódio reflete a tendência global de responsabilização corporativa por falhas de segurança.

4. **Dimensão Operacional**: A exploração da vulnerabilidade no PAN-OS (2024) resultou na interrupção de serviços críticos durante 72 horas e na paralisação de centros de dados [112], [113], destacando a fragilidade das infraestruturas digitais contemporâneas e os efeitos em cadeia em ecossistemas empresariais interdependentes.

2.11 MEDIDAS DE MITIGAÇÃO E DEFESA

A mitigação eficaz contra vulnerabilidades de Execução Remota de Código (RCE) deve adotar uma abordagem de defesa em profundidade (*defence-in-depth*), integrando múltiplas camadas de proteção a nível técnico, processual e humano [114]. Esta estratégia multifacetada visa criar barreiras sucessivas que dificultem a exploração bem-sucedida de falhas de segurança, mesmo quando algumas medidas de proteção são contornadas.

Entre as principais salvaguardas atualmente recomendadas pela comunidade de segurança informática destacam-se:

1. **Gestão proativa de vulnerabilidades**: Implementação de um programa contínuo de identificação, avaliação e correção de vulnerabilidades, com priorização baseada no CVSS (*Common Vulnerability Scoring System*), dando atenção imediata a falhas com pontuação igual ou superior a 9 [115]. Este processo deve incluir a monitorização ativa de boletins de segurança e a aplicação sistemática de patches.

2. **Web Application Firewalls (WAFs)**: Implementação de sistemas de firewall para aplicações web que combinem assinaturas de ataques conhecidos com técnicas de aprendizagem automática para detetar padrões de tráfego anómalos [116]. Estas soluções oferecem proteção temporária contra ataques zero-day, permitindo ganhar tempo para a implementação de correções permanentes.

3. **Isolamento de processos**: Utilização de tecnologias de contenção como Docker e gVisor para criar micro domínios de execução que limitam o impacto potencial de uma exploração RCE bem-sucedida [117]. Esta abordagem impede que um atacante comprometa todo o sistema mesmo quando consegue executar código remotamente.

4. **Controlo de acessos**: Aplicação rigorosa do princípio do privilégio mínimo (*least privilege*) combinado com estratégias avançadas de segmentação de rede (macro e micro segmentação) [118]. Estas medidas reduzem significativamente a capacidade de movimentação lateral do atacante dentro da infraestrutura.

5. **Monitorização avançada**: Implementação de sistemas SIEM (*Security Information and Event Management*) capazes de correlacionar logs de diversas fontes, complementados com modelos de deteção baseados em *machine learning* para identificar padrões anómalos de comportamento [119]. Esta abordagem permite detetar atividades suspeitas que poderiam indicar uma tentativa ou exploração bem-sucedida de RCE.

6. **Desenvolvimento seguro**: Adoção de práticas de codificação segura e implementação de processos automatizados de revisão de código que identifiquem funções potencialmente perigosas (como exec, eval, system, etc.) [120]. Esta medida preventiva é particularmente importante para reduzir a introdução de vulnerabilidades RCE durante o desenvolvimento de software.

A eficácia destas medidas depende fortemente da sua implementação integrada e da atualização contínua face à evolução das técnicas de ataque. Como demonstram estudos recentes [121], organizações que adotam esta abordagem estratificada conseguem reduzir significativamente o risco de exploração bem-sucedida de vulnerabilidades RCE.

2.12 ENQUADRAMENTO LEGAL E ÉTICO DOS TESTES DE INTRUSÃO

A realização de testes de intrusão que envolvam a exploração de vulnerabilidades de Execução Remota de Código (RCE) está sujeita a um rigoroso enquadramento legal e ético, que os profissionais de segurança informática devem escrupulosamente observar [122]. Este quadro normativo visa garantir que as atividades de avaliação de vulnerabilidades não violam direitos fundamentais nem comprometem a operacionalidade dos sistemas testados.
No âmbito legal, destacam-se três instrumentos fundamentais:

1. Diretiva NIS 2 (UE 2022/2555): Estabelece requisitos obrigatórios de segurança e de reporte de incidentes para operadores de serviços essenciais e entidades de médio e grande porte, incluindo a obrigatoriedade de testes de segurança regulares [123].

2. Regulamento Geral sobre a Proteção de Dados (RGPD): Exige a implementação de medidas técnicas e organizativas adequadas para garantir a segurança dos dados pessoais, prevendo sanções significativas para casos de negligência na gestão de vulnerabilidades [124].

3. Legislação nacional específica: Inclui diplomas como o *Computer Misuse Act* (Reino Unido) e o *Computer Fraud and Abuse Act* (EUA), que criminalizam o acesso não autorizado a sistemas informáticos, mesmo quando não resulte em dano aparente [125].

No plano ético, os profissionais devem guiar-se por códigos de conduta reconhecidos internacionalmente, como o estabelecido pelo (ISC)², que enfatiza princípios fundamentais [126]:

- Consentimento explícito e documentado antes da realização de qualquer teste
- Minimização do impacto nas operações normais dos sistemas

- Confidencialidade relativamente às informações obtidas
- Reporte responsável de vulnerabilidades identificadas
- Limitação do âmbito aos sistemas e períodos autorizados

Este enquadramento é particularmente relevante no contexto de testes que envolvam RCE, dada a natureza crítica destas vulnerabilidades e o seu potencial impacto [127]. Os profissionais devem ainda considerar as orientações específicas contidas em normas técnicas como a ISO/IEC 29147 (divulgação de vulnerabilidades) e a ISO/IEC 30111 (processamento de informação sobre vulnerabilidades) [128].

A observância destes princípios legais e éticos é essencial para garantir que os testes de intrusão cumpram o seu objetivo de melhorar a segurança dos sistemas sem violar direitos fundamentais ou comprometer a estabilidade operacional das organizações [129].

2.13 TENDÊNCIAS EMERGENTES E DESAFIOS FUTUROS

O panorama das vulnerabilidades de Execução Remota de Código (RCE) está em constante evolução, com novas tendências tecnológicas a introduzirem desafios significativos para a segurança informática [130]. Estas transformações exigem abordagens inovadoras e a antecipação de riscos emergentes.

Infraestruturas como Código (IaC): A adoção crescente de ferramentas como Terraform e Ansible introduz novos vetores de risco, onde erros em ficheiros de configuração podem propagar vulnerabilidades em grande escala e de forma automatizada [131]. Este fenómeno é particularmente preocupante em ambientes de cloud computing, onde uma única configuração incorreta pode comprometer múltiplos sistemas simultaneamente.

Inteligência Artificial Generativa: Os avanços em *Large Language Models* (LLMs) estão a transformar o panorama de ameaças, com estas tecnologias a serem utilizadas

tanto para a criação automatizada de exploits como para o desenvolvimento de mecanismos de defesa mais sofisticados [132]. Um exemplo paradigmático é a geração automática de assinaturas para Web Application Firewalls (WAFs), que permite responder mais rapidamente a novas ameaças.

Computação Quântica: Embora ainda em fase de desenvolvimento, a computação quântica representa um desafio futuro significativo, com potencial para comprometer os primitivos criptográficos que atualmente protegem comunicações e mitigam os riscos de pós-exploração [133]. Esta evolução exigirá a migração para algoritmos pós-quânticos para manter a segurança dos sistemas.

Edge Computing e redes 5G: A dispersão de microserviços em ambientes de *Multi-access Edge Computing* (MEC) expande consideravelmente a superfície de ataque, requerendo a adoção de modelos *zero-trust* de ponta a ponta [134]. Esta arquitetura distribuída introduz novos desafios na deteção e contenção de tentativas de exploração de RCE, particularmente em ambientes com latência ultrabaixa.

Estas tendências emergentes destacam a necessidade de:

1. Abordagens proativas na gestão de configurações IaC
2. Investimento contínuo em pesquisa e desenvolvimento de contramedidas baseadas em inteligência artificial
3. Preparação para a transição para criptografia pós-quântica
4. Adaptação dos modelos de segurança para arquiteturas distribuídas edge/5G

Como demonstram os estudos mais recentes [135], a capacidade de antecipar e responder a estes desafios será determinante para a eficácia das estratégias de segurança contra explorações RCE no futuro próximo.

2.14 SÍNTESE CRÍTICA DA LITERATURA

A análise da literatura especializada permite constatar um consenso quanto à natureza transversal das vulnerabilidades de Execução Remota de Código (RCE), que afetam de forma integrada as camadas aplicacionais, middleware e de sistema operativo [136]. Contudo, identifica-se uma discrepância significativa entre as recomendações académicas e as práticas efetivas de gestão de atualizações de segurança (*patch management*) no contexto empresarial real [137].

Os estudos revelam ainda uma lacuna preocupante nos mecanismos de monitorização contínua nos ambientes de desenvolvimento modernos (DevSecOps), onde vulnerabilidades são frequentemente introduzidas nas fases iniciais do ciclo de vida da aplicação [138]. Esta situação é particularmente crítica quando consideramos que:

1. A maioria das organizações não implementa processos sistemáticos de verificação de segurança durante as fases de desenvolvimento
2. Existe uma desconexão entre as equipas de desenvolvimento e segurança
3. Os controlos de segurança são frequentemente considerados apenas em fase de produção [139]

Casos de estudo recentes demonstram a necessidade urgente de [140]:

- Automatização dos processos de reforço de segurança ("*Infrastructure Hardening as Code*")
- Desenvolvimento de competências especializadas em equipas de defesa (*blue-team*)
- Implementação de capacidades de resposta em tempo real a incidentes
- Adoção de abordagens de segurança contínua ao longo do ciclo de vida do desenvolvimento

A literatura evidencia igualmente que as atuais estratégias de mitigação frequentemente falham em considerar a natureza evolutiva dos vetores de ataque RCE, que se adaptam

continuamente para contornar as medidas defensivas existentes [141]. Esta constatação reforça a importância de uma abordagem dinâmica e proativa na gestão deste tipo de vulnerabilidades.

2.15 CONCLUSÃO DO CAPÍTULO

A análise desenvolvida ao longo deste capítulo demonstra cabalmente a natureza ubíqua e a gravidade crítica das vulnerabilidades de Execução Remota de Código (RCE) nos sistemas informáticos contemporâneos [142]. Os resultados obtidos evidenciam que uma mitigação eficaz exige necessariamente a conjugação sinérgica de três dimensões fundamentais:

1. Medidas técnicas robustas, implementadas segundo uma abordagem de defesa em profundidade
2. Modelos de governabilidade adequados que assegurem a conformidade legal e ética
3. Programas de formação contínua para profissionais de segurança e desenvolvimento [143]

O estudo aprofundado dos vetores de ataque, ferramentas de exploração e impactos potenciais proporcionou um enquadramento teórico sólido e indispensável para a componente prática desta investigação [144]. Esta base conceptual permitirá, no capítulo seguinte:

- A conceção de experiências controladas que reproduzam cenários realistas de exploração RCE
- A validação empírica das estratégias de mitigação discutidas
- A obtenção de dados quantitativos sobre a eficácia de diferentes abordagens defensivas

Como salientam os estudos mais recentes [145], a complexidade crescente das ameaças RCE exige uma contínua atualização do conhecimento teórico e prático, sendo este

capítulo um contributo para esse esforço coletivo da comunidade de segurança informática.

3 METODOLOGIA

A execução da componente prática desta investigação foi cuidadosamente planeada segundo uma abordagem metodológica estruturada e sistemática, assegurando não só a validade científica dos resultados obtidos, mas também a sua reprodutibilidade e a segurança inerente ao processo de exploração de vulnerabilidades do tipo *Remote Code Execution* (RCE). Esta dimensão experimental, fundamentada nos princípios basilares da segurança ofensiva, utiliza ambientes rigorosamente controlados e isolados, especificamente concebidos para simular cenários de ataque com elevado rigor técnico, mantendo simultaneamente a salvaguarda absoluta de infraestruturas operacionais e sistemas em produção.

3.1 ENQUADRAMENTO METODOLÓGICO

A presente investigação adota como referencial metodológico o *Penetration Testing Execution Standard* (PTES), amplamente reconhecido como padrão de excelência no domínio da segurança ofensiva computacional. Este *framework*, validado internacionalmente tanto pela comunidade académica como pelos profissionais do setor, estabelece um conjunto de conceitos técnicos e procedimentos sistematizados que garantem a execução de avaliações éticas de segurança com rigor científico e validade empírica [146].

A pertinência da adoção deste paradigma justifica-se pela sua consagração entre os principais atores do ecossistema de cibersegurança, nomeadamente:

- Unidades especializadas em operações ofensivas avançadas de *red teaming*
- Consultorias estratégicas em segurança da informação de nível empresarial
- Centros de investigação em cibersegurança de instituições de referência

- Organismos internacionais de certificação e auditoria técnica [147]

Arquitetura Processual do Modelo PTES

O referencial em análise estrutura-se num ciclo sequencial composto por sete fases interdependentes, cada uma caracterizada por objetivos operacionais específicos e critérios de avaliação mensuráveis:

1. **Fase de Observação e Definição Conceptual**

    - Delimitação taxonómica do âmbito de intervenção
    - Formulação de objetivos operacionais SMART (*Specific, Measurable, Achievable, Relevant, Time-bound*)
    - Estabelecimento de princípios éticos e enquadramento legal
    - Definição de métricas de sucesso validadas [148]

2. **Fase de Inteligência (Reconhecimento Passivo)**

    - Recolha sistemática de inteligência através de técnicas OSINT (*Open Source INTelligence*)
    - Análise forense de meta dados e pegadas digitais
    - Mapeamento de infraestruturas via fontes públicas abertas [149]

3. **Fase de Análise de Ameaças**

    - Enumeração ativa de vetores de ataque potenciais
    - Identificação e catalogação de vulnerabilidades conhecidas (CVE)
    - Classificação de ameaças segundo modelos STRIDE (*Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege*) [150]

4. **Fase de Modelação de Ataque**

    - Desenvolvimento de cenários de comprometimento
    - Priorização estratégica de vetores de exploração
    - Simulação de técnicas TTPs (*Tactics, Techniques and Procedures*) [151]

5. **Fase de Exploração Ativa**

    - Execução controlada de técnicas de RCE
    - Validação empírica das hipóteses de comprometimento
    - Documentação forense dos processos de intrusão [152]

6. **Fase de Pós-Exploração**

    - Análise de impacto sistémico multidimensional
    - Mapeamento de cadeias de ataque completas (*Cyber Kill Chain*)
    - Avaliação de persistência e movimentação lateral
    - Extração forense de evidências digitais [153]

7. **Fase de Consolidação e Relato**

    - Sistematização das identificações técnicas
    - Elaboração de matrizes de risco ponderadas
    - Formulação de recomendações mitigatórias estratificadas
    - Produção de artefactos técnicos normalizados [154]

Este quadro conceptual confere robustez científica à investigação, garantindo que os procedimentos laboratoriais desenvolvidos estejam alinhados com os mais exigentes padrões internacionais de segurança ofensiva. A adoção deste referencial assegura não apenas a validade interna dos resultados obtidos, como também a sua comparabilidade com estudos similares no domínio da cibersegurança, contribuindo significativamente para o avanço do conhecimento nesta área de estudo [155].

3.2 IMPLEMENTAÇÃO PRÁTICA DA METODOLOGIA

A operacionalização da metodologia decorreu em sete fases sequenciais, descritas de forma detalhada a seguir:

Fase 1 – Planeamento do Ambiente Foi estabelecida uma matriz de recursos técnicos e critérios éticos fundamentais para a investigação [156]. Selecionaram-se sistemas vulneráveis conhecidos (CVEs documentados) e implementou-se um ambiente de teste sob controlo absoluto, garantindo o isolamento completo da rede institucional e de sistemas produtivos. Esta fase incluiu a definição de protocolos de contingência para eventuais falhas de contenção.

Fase 2 – Configuração da Infraestrutura Implementou-se um laboratório especializado composto por [157]:

- Aplicações vulneráveis intencionais (Joomla 3.4.6, WordPress 4.2, phpMyAdmin 4.0.x)
- Sistemas isolados em Raspberry Pi 4 Model B
- Máquinas virtuais VMware ESXi com configurações específicas
- Rede privada virtual com monitorização contínua

Fase 3 – Seleção de Ferramentas Adotou-se um conjunto de ferramentas validadas pela comunidade de segurança ofensiva [158]:

- **Nmap** (mapeamento de rede e serviços)
- **WhatWeb** (identificação de tecnologias web)
- **Skipfish** (análise de segurança em aplicações web)
- **Metasploit Framework** (exploração de vulnerabilidades)

Fase 4 – Execução do Ciclo Ofensivo Aplicou-se rigorosamente o ciclo PTES com as seguintes atividades principais [159]:

1. Reconhecimento passivo e ativo
2. Enumeração detalhada de serviços
3. Exploração controlada de vulnerabilidades RCE
4. Operações de pós-exploração limitadas

Fase 5 – Registo e Análise de Resultados Documentaram-se todos os processos com [160]:

- Capturas de ecrã cronológicas
- Logs detalhados de atividades
- Evidências de acesso a ficheiros sensíveis
- Provas conceptuais de execução remota

Fase 6 – Validação de Impactos Realizou-se uma análise crítica multidimensional [161]:

- Técnica (causas raiz das vulnerabilidades)
- Operacional (impacto potencial em ambientes reais)
- Organizacional (implicações para a gestão de segurança)

Fase 7 – Formulação de Recomendações Desenvolveram-se propostas de mitigação estratificadas [162]:

3.3 REPRESENTAÇÃO GRÁFICA DA METODOLOGIA PTES

A Figura 4 apresenta uma síntese visual das sete fases do modelo PTES [163], conforme implementadas no âmbito desta investigação prática, demonstrando a sequência lógica e as relações entre cada etapa do processo. Esta representação gráfica foi especialmente adaptada para refletir as particularidades do estudo em questão, mantendo, contudo, a fidelidade aos princípios fundamentais do padrão PTES.

O diagrama desenvolvido ilustra de forma clara:

1. A natureza iterativa e cíclica do processo de teste de penetração
2. Os produtos gerados em cada fase metodológica
3. Os mecanismos de controlo e validação implementados
4. Os pontos de decisão crítica entre fases

A estrutura representada evidencia como as diferentes componentes da metodologia - desde o planeamento inicial até ao relatório final - se articulam sistematicamente para garantir uma avaliação abrangente e rigorosa das vulnerabilidades RCE em ambiente controlado.

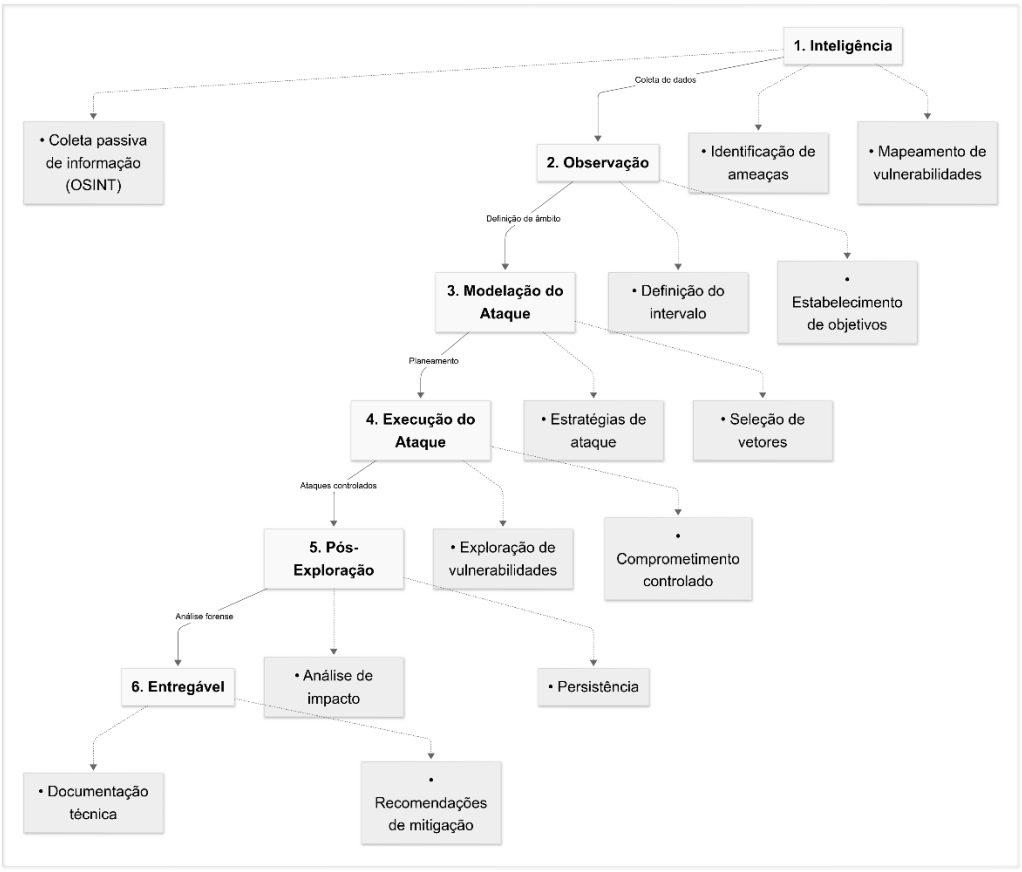

Figura 4 - Fluxo de Metodologia PTES (adaptado de PTES 2017)

3.4 CONTEXTUALIZAÇÃO INTERNACIONAL DA AMEAÇA DE EXECUÇÃO REMOTA DE CÓDIGO (RCE): ENQUADRAMENTO DO OBJETO DE ESTUDO

A evolução das ameaças no ciberespaço tem acompanhado, de forma simbiótica, a aceleração da transformação digital em escala global. A proliferação de ecossistemas

digitais – incluindo serviços web, aplicações empresariais críticas, infraestruturas essenciais digitalizadas e arquiteturas de computação em nuvem – expandiu consideravelmente a superfície de ataque das organizações, exacerbando os desafios inerentes à sua proteção eficaz.

Neste contexto, as vulnerabilidades de Execução Remota de Código (RCE) assumem particular relevância no atual panorama da cibersegurança. Essa preponderância deve-se às suas características intrínsecas, que incluem:

1. A capacidade de executar código arbitrário em sistemas vulneráveis;
2. A possibilidade de exploração sem requisitos prévios de autenticação;
3. O potencial para comprometer integralmente os ativos digitais;
4. A aptidão para funcionar como vetor inicial de ataques laterais e escalonamento de privilégios.

Tais atributos colocam as vulnerabilidades RCE no topo das matrizes de risco contemporâneas, consolidando-as como um domínio de estudo prioritário tanto a nível acadêmico quanto para os profissionais de cibersegurança. A presente dissertação insere-se neste enquadramento, visando contribuir para a compreensão e mitigação deste desafio crítico da segurança digital.

3.5 A PROFUNDIDADE TÉCNICA DA AMEAÇA DE EXECUÇÃO REMOTA DE CÓDIGO (RCE)

A execução remota de código (RCE) constitui uma das ameaças mais críticas no atual panorama de cibersegurança, destacando-se pela sua complexidade técnica intrínseca e pelo potencial impacto devastador que pode acarretar [165]. Esta vulnerabilidade permite que agentes maliciosos obtenham controlo total sobre sistemas vulneráveis, explorando falhas em componentes de *software*, bibliotecas externas ou configurações

inadequadas, frequentemente sem necessidade de dispor de credenciais de autenticação válidas [166].

A gravidade singular desta ameaça reside na sua natureza ubíqua e transversal a diversos setores. Ao contrário de vulnerabilidades mais circunscritas, o RCE manifesta-se com igual severidade em contextos organizacionais e tecnológicos díspares, incluindo infraestruturas empresariais críticas, plataformas governamentais sensíveis, sistemas financeiros complexos, ambientes industriais automatizados e ecossistemas académicos especializados [167]. Esta omnipresença transforma o RCE num vetor de ataque particularmente temível, capaz de comprometer desde sistemas isolados até redes inteiras de organizações.

Uma análise histórica das vulnerabilidades RCE mais exploradas na última década revela um padrão perturbador: a sua origem está maioritariamente associada a práticas de desenvolvimento que negligenciaram princípios fundamentais de segurança informática [168]. Entre os fatores determinantes mais relevantes destacam-se a ausência de mecanismos robustos de validação de inputs, que permitem a injeção arbitrária de código malicioso; a exposição indevida de interfaces administrativas, frequentemente carentes de mecanismos de autenticação adequados; a gestão negligente de dependências externas, incluindo a persistência no uso de bibliotecas com vulnerabilidades conhecidas (identificáveis através do sistema *Common Vulnerabilities and Exposures* - CVE); e a manutenção deficiente de componentes de *software*, com utilização prolongada de versões desatualizadas e potencialmente comprometidas [169].

Estes vetores de vulnerabilidade convergem para criar um cenário operacional particularmente preocupante. Dados recentes indicam que a maioria das explorações RCE ocorre sem exigência de credenciais válidas, sendo o período médio entre a divulgação pública da vulnerabilidade e a sua exploração ativa notavelmente reduzida [170]. Esta realidade exige uma abordagem multidisciplinar que integre de forma coerente práticas de desenvolvimento seguro, arquiteturas defensivas em profundidade, monitorização contínua de componentes e políticas rigorosas de gestão de atualizações - constituindo precisamente o objeto central desta investigação [171].

3.6 CASOS PARADIGMÁTICOS E REPERCUSSÕES GLOBAIS DA AMEAÇA RCE

A ameaça de Execução Remota de Código (RCE) tem-se manifestado através de incidentes de cibersegurança com proporções catastróficas, cujas repercussões transcendem o âmbito técnico, assumindo dimensões financeiras, jurídicas, reputacionais e operacionais sem precedentes [172]. A análise de casos emblemáticos revela padrões preocupantes na evolução das capacidades ofensivas no ciberespaço, oferecendo lições cruciais para a comunidade de segurança informática.

O incidente da Equifax em 2017 permanece como paradigma dos riscos associados à gestão negligente de vulnerabilidades conhecidas [173]. A exploração da falha CVE-2017-5638 no *framework* Apache Struts 2 - cujo *patch* estava disponível, mas não aplicado - demonstrou de forma crua as consequências sistémicas de práticas deficientes de gestão de vulnerabilidades. O comprometimento de dados sensíveis de 145 milhões de cidadãos norte-americanos resultou não apenas em sanções financeiras recorde (superiores a 700 milhões de dólares), mas também na demissão do quadro executivo e na completa reestruturação das políticas de segurança da organização [174]. Este caso sublinha a necessidade imperiosa de implementação rigorosa de ciclos de gestão de atualizações de segurança (*patch management*), particularmente em infraestruturas que processam dados críticos.

O episódio Log4Shell (CVE-2021-44228) em dezembro de 2021 representou um ponto de viragem na perceção global das vulnerabilidades em cadeias de abastecimento de *software* [175]. A ubiquidade da biblioteca Log4j, componente fundamental em ecossistemas Java, transformou esta vulnerabilidade num vetor de ataque transversal. A peculiaridade técnica que permitia a execução remota de código através da simples inserção de *strings* maliciosas em campos de *log* revelou fragilidades estruturais nos modelos atuais de desenvolvimento seguro. A reação internacional foi imediata e sem precedentes: a CISA classificou a vulnerabilidade como "crítica", enquanto gigantes tecnológicos como Microsoft, Amazon e Cisco mobilizaram equipas de emergência

para mitigação [176]. Este caso evidenciou a interdependência global em matéria de segurança digital e a necessidade de mecanismos coordenados de resposta a incidentes.

Mais recentemente, em 2023, a exploração da vulnerabilidade na plataforma MOVEit Transfer pelo grupo Cl0p demonstrou a sofisticação alcançada pelo cibercrime organizado [177]. A operação, que afetou instituições governamentais e corporações em múltiplas jurisdições, revelou padrões operacionais comparáveis a unidades de ciberguerra estatais: exploração automatizada, exfiltração silenciosa de dados e monetização eficiente através de esquemas de extorsão. O impacto estimado em 10 mil milhões de dólares e o comprometimento de mais de 40 milhões de ficheiros confidenciais sublinham a dimensão estratégica que as ameaças RCE assumiram no atual panorama geopolítico digital [178].

Estes casos paradigmáticos evidenciam quatro tendências críticas [179]:

1. A crescente sofisticação técnica das explorações de RCE
2. A escalada dos impactos económicos e operacionais
3. A necessidade premente de quadros regulatórios internacionais mais robustos
4. A urgência em desenvolver mecanismos proativos de deteção e resposta

A análise destes incidentes não só ilustra a evolução da ameaça RCE, como também serve de base para a formulação de estratégias mais eficazes de prevenção e mitigação no âmbito da cibersegurança global [180]. A lição fundamental que emerge destes casos é a necessidade de uma abordagem holística que combine vigilância tecnológica, melhores práticas de desenvolvimento, cooperação internacional e capacitação contínua dos profissionais de segurança.

3.7 EVOLUÇÃO E SOFISTICAÇÃO NA EXPLORAÇÃO DE VULNERABILIDADES DE RCE: TENDÊNCIAS CONTEMPORÂNEAS

A exploração de vulnerabilidades de Execução Remota de Código (RCE) tem registado uma evolução significativa, acompanhando o ritmo acelerado da transformação digital

nas organizações [181]. Na atualidade, observa-se uma crescente profissionalização e sofisticação nas técnicas empregues por agentes maliciosos, que passaram a incorporar ferramentas avançadas de reconhecimento e exploração automatizada, transformando radicalmente o panorama de ameaças.

As campanhas ofensivas modernas utilizam sistematicamente plataformas de inteligência de superfície de ataque como Shodan, Censys e ZoomEye, que funcionam como motores de busca especializados na identificação de sistemas vulneráveis à escala global [182]. Estas ferramentas permitem:

- A varredura massiva de dispositivos expostos na Internet
- Filtragem por parâmetros técnicos específicos (versões de aplicações vulneráveis, portas abertas, cabeçalhos expostos)
- Identificação de milhares de potenciais alvos em segundos
- Integração com *pipelines* de ataque automatizados [183]

Paralelamente, regista-se uma transição fundamental no modus operandi dos cibe criminosos, que migraram de ciberataques oportunistas para operações altamente direcionadas e persistentes [184]. Neste novo paradigma:

1. A exploração inicial de vulnerabilidades RCE constitui apenas o primeiro estágio
2. Seguem-se técnicas avançadas de movimentação lateral na rede
3. Implementam-se mecanismos de elevação de privilégios
4. Executa-se exfiltração seletiva de dados [185]

Esta abordagem estratificada visa estabelecer uma presença persistente nos sistemas comprometidos e maximizar o impacto estratégico das intrusões, transformando as vulnerabilidades RCE em vetores de ataques prolongados e de alto impacto.
Um desenvolvimento particularmente preocupante é o surgimento, na dark web, de plataformas de *Exploitation-as-a-Service* (EaaS), que representam a comercialização e democratização de capacidades ofensivas [186]. Estas plataformas oferecem:

- Kits de exploração pré-configurados para diversas tecnologias
- Documentação técnica detalhada
- Suporte especializado a cibe criminosos
- Modelos de subscrição ou pagamento por utilização [187]

Este modelo de negócio criminoso reduz drasticamente a barreira de entrada para ataques sofisticados, permitindo que indivíduos com conhecimentos técnicos limitados explorem vulnerabilidades complexas [188].

A convergência destas tendências resulta num cenário operacional marcado por [189]:

1. Diminuição do tempo entre identificação e exploração de vulnerabilidades
2. Expansão da superfície de ataque para toda a cadeia de fornecedores
3. Aumento da assimetria entre capacidades ofensivas e defensivas
4. Crescente profissionalização do cibercrime organizado

Perante esta realidade, torna-se imperiosa uma resposta proporcionalmente mais sofisticada, que inclua [190]:

- Implementação de mecanismos proativos de monitorização
- Arquiteturas de defesa em profundidade
- Modelos colaborativos de partilha de inteligência sobre ameaças
- Desenvolvimento de quadros regulatórios robustos para a cadeia de fornecimento de *software*
- Adoção de abordagens de *threat hunting* proativo

3.8 FRAGILIDADES ESTRUTURAIS E DESAFIOS GLOBAIS NA SEGURANÇA DE *SOFTWARE*

A crescente dependência de ecossistemas de código aberto, frequentemente sustentados por comunidades reduzidas de voluntários, tem introduzido vulnerabilidades sistémicas na infraestrutura digital global. O caso paradigmático do Log4Shell revelou de forma

contundente como uma única vulnerabilidade numa biblioteca ubíqua pode comprometer simultaneamente milhões de sistemas interligados, expondo as fragilidades inerentes ao atual modelo de desenvolvimento distribuído.

Existem três eixos críticos que perpetuam essa situação:

1. Deficiências nos processos de garantia de qualidade: A ausência de mecanismos formais de revisão de código, a insuficiência de testes de segurança integrados nos pipelines DevOps contemporâneos e a carência de sistemas automatizados de atualização criam um ambiente propício à persistência de vulnerabilidades latentes, que frequentemente permanecem não detetadas durante ciclos de desenvolvimento inteiros.

2. Modelos reativos de gestão de riscos: Muitas organizações persistem em adotar abordagens fragmentadas à segurança cibernética, nas quais a identificação de vulnerabilidades e a aplicação de correções dependem de processos manuais e auditorias periódicas, em vez de sistemas contínuos de monitorização e resposta.

3. Barreiras estruturais à adoção de boas práticas: A escassez aguda de profissionais qualificados em segurança ofensiva, combinada com resistências organizacionais à implementação de arquiteturas confiança zero, estratégias de defesa em profundidade e mecanismos de proteção em tempo real, perpetuam ambientes digitais intrinsecamente vulneráveis.

Esta tríade de fatores - deficiências processuais, modelos de gestão obsoletos e barreiras à inovação - configura um cenário onde as vulnerabilidades RCE encontram terreno fértil para exploração, exigindo uma reavaliação fundamental dos paradigmas atuais de desenvolvimento e operação de sistemas críticos.

3.9 QUADRO REGULATÓRIO INTERNACIONAL E INICIATIVAS DE RESPOSTA À AMEAÇA RCE

- Perante a escalada de incidentes de segurança digital de proporções sistémicas, observa-se uma mobilização concertada de entidades reguladoras a nível global para estabelecer mecanismos mais robustos de prevenção e resposta. Essa resposta institucional reflete o reconhecimento crescente da natureza estratégica das vulnerabilidades de Execução Remota de Código (RCE) no contexto da segurança nacional e da estabilidade económica.

- A União Europeia, por meio da Diretiva NIS2 (*Network and Information Security*), estabeleceu um marco regulatório sem precedentes ao impor requisitos rigorosos aos operadores de serviços essenciais nos setores de energia, transporte, finanças, saúde e infraestruturas digitais. O texto legislativo, que entrou em vigor em janeiro de 2023, contempla:

- 1. A obrigatoriedade de implementação de sistemas avançados de monitorização contínua
- 2. Protocolos padronizados para gestão de incidentes
- 3. Mecanismos proativos de mitigação de vulnerabilidades
- 4. Prazos curtos para notificação de incidentes (24 horas para relatório inicial)
- 5. Regime sancionatório que pode atingir 10 milhões de euros ou 2% do volume de negócios global

- Paralelamente, os Estados Unidos reforçaram a *Executive Order* 14028, que estabelece padrões mandatórios para fornecedores da administração federal. Entre as exigências mais relevantes destacam-se:

  • Implementação de estruturas integradas de mitigação de explorações (EDR, WAF, SIEM)
  • Adoção de ambientes controlados (*sandboxing*) para análise de código
  • Monitorização contínua da cadeia de fornecimento de *software*

• Arquiteturas *zero-trust* para sistemas críticos

No âmbito técnico-normativo, iniciativas como o MITRE ATT&CK *Framework* têm proporcionado uma taxonomia abrangente de táticas, técnicas e procedimentos (TTPs) utilizados em ataques RCE. Simultaneamente, a *Open Web Application Security Project* (OWASP) mantém esforços contínuos de:

1. Catalogação das vulnerabilidades mais críticas
2. Desenvolvimento de guias de referência
3. Criação de ferramentas de avaliação aberta
4. Manutenção de *rankings* atualizados de ameaças

Estes esforços combinados - legislativos, normativos e técnicos - refletem uma compreensão maturada da natureza multidimensional da ameaça RCE. O presente capítulo não apenas evidencia esta complexidade, mas também fundamenta a relevância da análise empírica subsequente como contributo para:

1. A consciencialização sobre vetores de ataque emergentes
2. A capacitação técnica de profissionais de segurança
3. O desenvolvimento de arquiteturas resilientes
4. A formulação de políticas públicas informadas

Neste contexto, a análise prática que se segue assume particular relevância como exercício aplicado à exploração de uma vulnerabilidade critica do tipo RCE, oferecendo insights valiosos para a construção de defesas mais eficazes contra ameaças RCE na era da transformação digital acelerada.

4 LABORATÓRIO PRÁTICO

Este capítulo descreve a infraestrutura prática desenvolvida para simulação de técnicas de segurança ofensiva, destacando sua arquitetura, configuração e relevância para os objetivos do estudo.

4.1 CONTEXTUALIZAÇÃO

Este laboratório foi instalado no hardware Raspberry Pi, uma solução de baixo custo que tem como objetivo fornecer um ambiente prático a estudantes e profissionais da área que pretendam simular técnicas de segurança ofensiva.

Baseado no sistema Raspberry Pi OS, o ambiente Raspwn incorpora vulnerabilidades conhecidas em serviços de rede, aplicações web, apresenta credenciais fracas e portas abertas. Disponibiliza um ponto de acesso sem fios, que dá acesso a uma infraestrutura de aplicações web contidas numa rede isolada, sem qualquer tipo de conectividade com a rede Internet ou outra.

Utilizou-se um método experimental para demonstração de uma exploração do tipo RCE, simulada através de um ambiente virtualizado, neste caso, uma máquina virtual executando o Kali Linux.

4.2 ARQUITETURA DE SERVIÇOS DE REDE

O laboratório simula uma infraestrutura agregando vários IPs e serviços de rede conectáveis através de portas pré-definidas conforme inventariação apresentada na tabela 2.

Tabela 2 - Inventário de IPs, serviços e portas de comunicação

Endereço IP	Serviço	Função Principal	Portas Padrão
192.168.99.1	BIND	Servidor DNS Autoritário	53 (TCP/UDP)
192.168.99.10	BIND	Servidor DNS Secundário	53 (TCP/UDP)
192.168.99.18	Postfix	Agente de Transferência de E-mail (MTA)	25 (SMTP), 587 (Submission)
192.168.99.18	Dovecot	Servidor de E-mail (IMAP/POP3)	143 (IMAP), 993 (IMAPS)
192.168.99.10	Samba	Partilha de ficheiros Microsoft	139, 445 (TCP)
192.168.99.13	Apache2	Servidor Web (HTTP/HTTPS)	80 (HTTP), 443 (HTTPS)
192.168.99.7	Nginx	Servidor Web/Proxy Reverso	80 (HTTP), 443 (HTTPS)
127.0.0.1	MySQL Server	Base de Dados Relacional	3306 (TCP)
192.168.99.1	OpenSSH	Acesso Remoto Seguro (SSH)	22 (TCP)

Notas Explicativas:

1. BIND: Serviço de resolução de nomes na rede. A redundância (192.168.99.1 e 192.168.99.10) mostra-nos uma arquitetura principal-secundária.
2. Postfix+Dovecot: Configuração típica para serviços de e-mail, onde o Postfix gere o envio e o Dovecot o acesso aos e-mails.
3. Samba: Permite integração com sistemas Windows (partilha de ficheiros/recursos).
4. Apache2/Nginx: Ambos disponibilizam conteúdo web, podendo estar em configuração de balanceamento de carga ou para serviços distintos.
5. MySQL: Configuração apenas para acesso local (127.0.0.1), aumentando a segurança.
6. OpenSSH: Serviço essencial para uma administração remota.

4.3 ARQUITETURA DE APLICAÇÕES

O laboratório simula uma arquitetura multicamada para a disponibilização de aplicações e pontos de entrada nas aplicações consolidados na tabela 3.

Tabela 3 - Compilação de aplicações e Urls

Aplicação	**Versão**	**URL de Acesso**	**Categoria**	**Porta**
Concrete CMS	5.6.3.4	http://playground.raspwn.org/concrete5/	CMS	80/TCP
Drupal	6.34	http://playground.raspwn.org/drupal-6.34/	CMS	80/TCP

Aplicação	Versão	URL de Acesso	Categoria	Porta
Drupal	7.34	http://playground.raspwn.org/drupal-7.34/	CMS	80/TCP
Joomla	2.5.28	http://playground.raspwn.org/joomla-2/	CMS	80/TCP
Joomla	3.4.0	http://playground.raspwn.org/joomla-3/	CMS	80/TCP
osCommerce	2.3	http://playground.raspwn.org/oscommerce/	E-commerce	80/TCP
phpBB	3.0.13	http://playground.raspwn.org/phpBB3/	Fórum	80/TCP
WordPress	3.8.1	http://playground.raspwn.org/wordpress3/	CMS/Blog	80/TCP
WordPress	4.1	http://playground.raspwn.org/wordpress4/	CMS/Blog	80/TCP
Zen-Cart	1.5.4	http://playground.raspwn.org/zencart/	E-commerce	80/TCP
phpMyAdmin	3.4.11	http://playground.raspwn.org/phpmyadmin/	Administração de Banco de Dados	80/TCP

Aplicação	Versão	URL de Acesso	Categoria	Porta
Samba SWAT	3.6.6	http://playground.raspwn.org:901/	Administração Samba	901/TCP
Roundcube	0.7.2	http://playground.raspwn.org/roundcube/	Webmail	80/TCP

4.4 FASES DO EXERCÍCIO OFENSIVO

O exercício ofensivo desenvolvido no laboratório estruturou-se em três fases fundamentais e sequenciais - reconhecimento, enumeração e exploração - que constituem o cerne de qualquer operação ofensiva bem-sucedida. O reconhecimento inicial permitiu mapear a superfície de ataque através da recolha inteligente de informação, estabelecendo as bases para a fase subsequente de enumeração, onde se identificaram e catalogaram sistematicamente os potenciais vetores de ataque. Por fim, a fase de exploração materializa o aproveitamento efetivo das vulnerabilidades identificadas, completando assim o ciclo ofensivo que, quando executado com rigor metodológico, revela não apenas as fragilidades dos sistemas analisados, mas também os mecanismos mais eficazes para a sua mitigação.

4.4.1 RECONHECIMENTO

A fase de reconhecimento é a etapa fundamental onde se reúne inteligência sobre os alvos antes de qualquer teste ofensivo. Após estabelecer ligação ao ambiente, via Wi-Fi, realizou-se reconhecimento interno para descoberta e mapeamento da infraestrutura. Através da atribuição automática de IP (DHCP), foi possível identificar, através do comando “ifconfig”, que apresenta dados de rede da interface de ligação, especificamente que o endereço IP atribuído à ligação do computador JLA-PENTESTER foi o 192.168.99.197 com uma netmask 255.255.255.0, o que permite

identificar que está inserido numa máscara de rede /24 (24 bits reservados para rede), isto é, 2^(32 - 24) - 2 = 2^8 - 2 = 256 - 2 = 254 endereços disponíveis para atribuição, conforme evidenciado na figura 5.

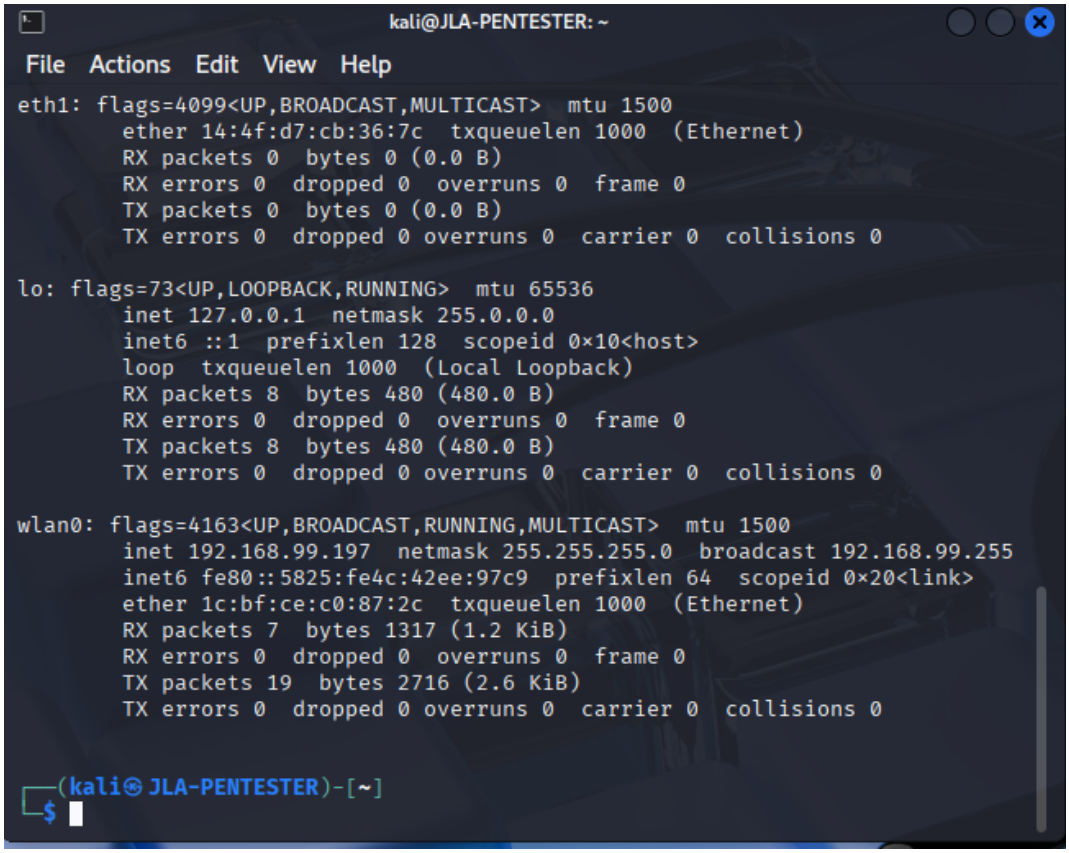

Figura 5 - Resultado do comando ifconfig

A execução do comando "arp -a" (*Address Resolution Protocol*) apresenta a **tabela arp** do sistema, ou seja, mapeia endereços IP para endereços físicos (MAC) em rede. Esta informação é valiosa para identificar dispositivos que estejam conectados, ou seja, possíveis alvos (ver figura 6).

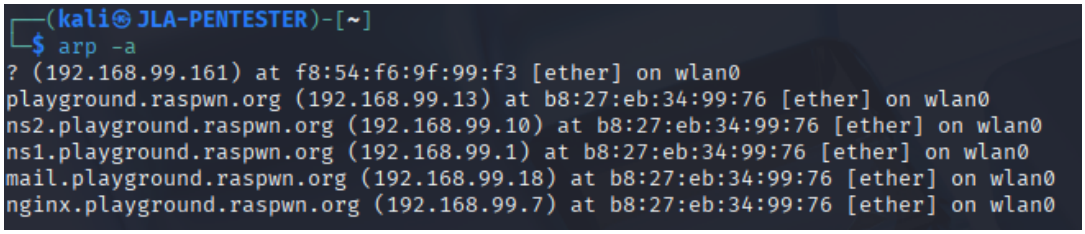

Figura 6 - mapeamento de endereços ativos na rede

4.4.2 ENUMERAÇÃO

A fase de enumeração neste ambiente visou obter, de forma estruturada, informações sobre serviços e vulnerabilidades. Esta abordagem estruturada permitiu priorizar alvos com base em vulnerabilidades conhecidas. Um network mapper, como a aplicação nmap, é a ferramenta essencial para enumeração de redes e serviços. A execução desta sem recorrer a opções específicas, apenas solicitando um varrimento por toda a rede onde o Kali Linux tem conectividade, permitiu identificar endereços IP contactáveis, resolução de nomes, endpoints disponíveis, portas abertas e serviços em execução, gerando entregáveis objetivos, conforme as figuras 7 e 8.

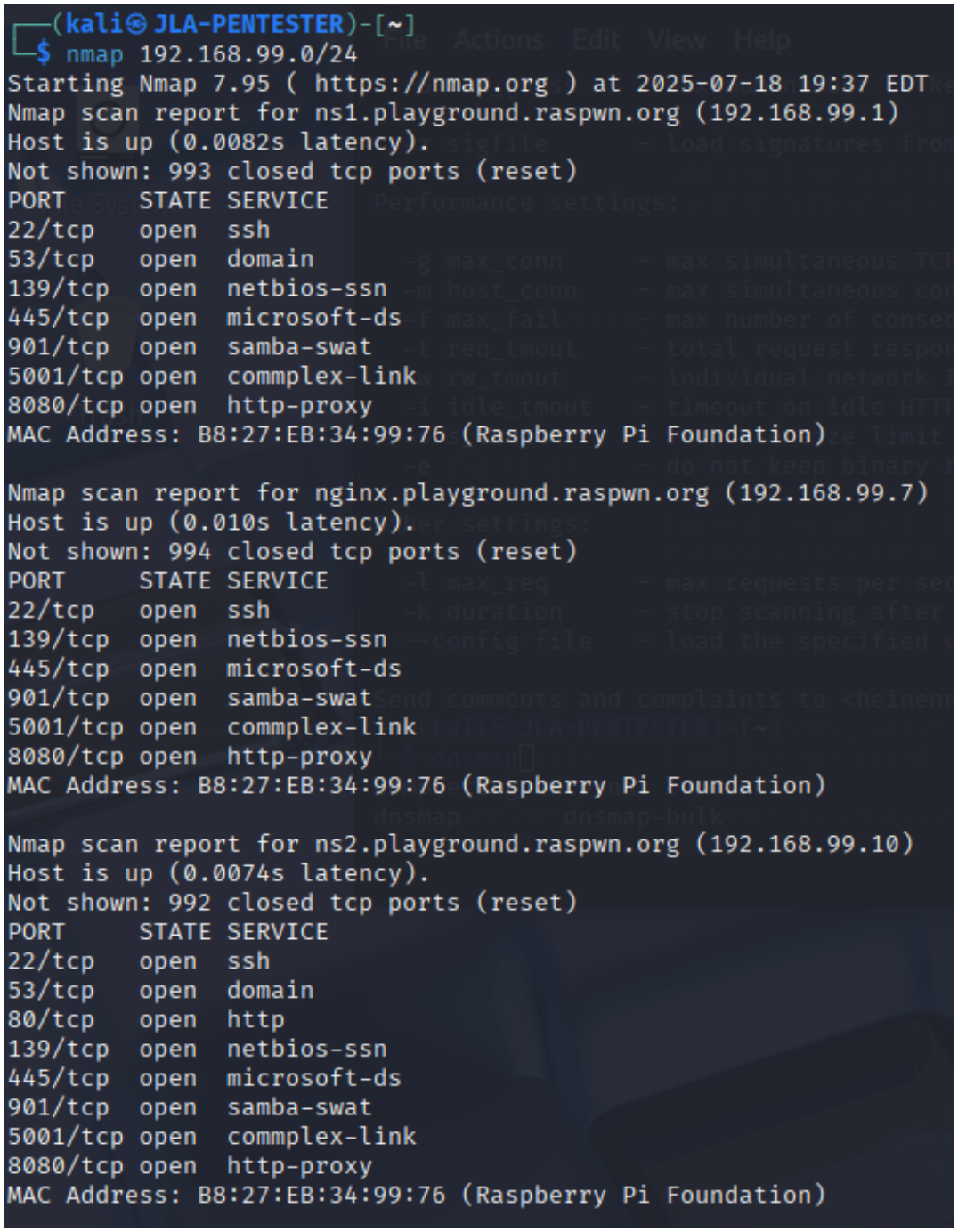

Figura 7 – Portscan à rede com identificação dos três hosts iniciais

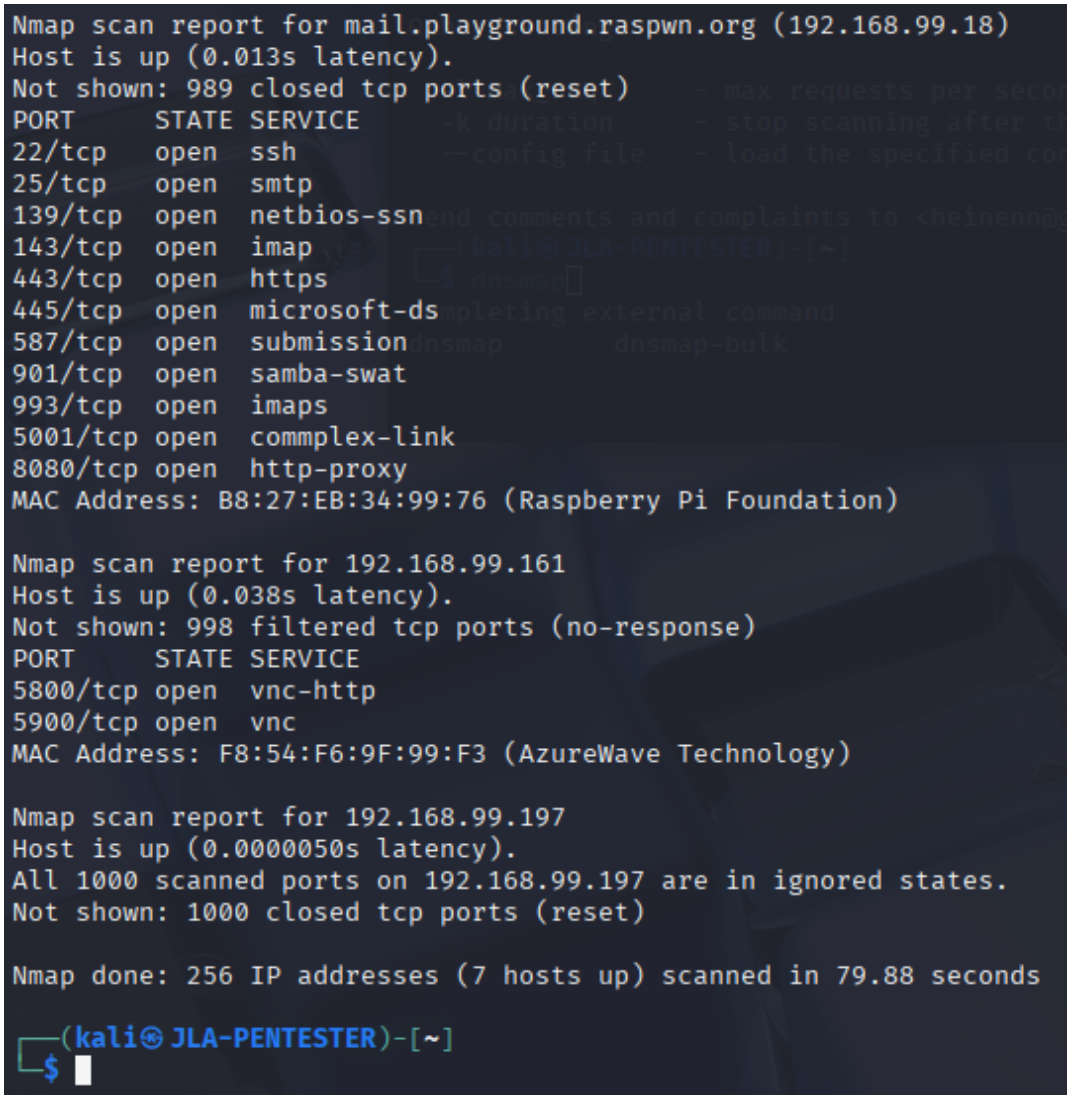

Figura 8 – Portscan à rede com identificação dos dois últimos hosts

Verifica-se que os IPs 192.168.99.10 e 192.168.99.13 apresentam portas abertas associadas a formatos web (80/443, protocolo HTTP e HTTPS, respetivamente), pelo que, de seguida, executou-se a aplicação WhatWeb contra os endereços IPs identificados através do protocolo HTTP, com o objetivo de mapear de forma célere a superfície de ataque. Esta poderosa ferramenta de *fingerprint* de tecnologias web identificou a versão dos servidores de páginas (Apache 2.2.22); no caso do IP 192.168.99.13, apresentou adicionalmente informações de um potencial *Web Playground*, conforme figura 9.

Figura 9 - Resultados da pesquisa de formatos web associadas a IPs

De seguida, foi selecionado o IP 192.168.99.13 como alvo para obtenção de mais informação sobre o referido *playground*. Executou-se uma indexação web, através da aplicação desenvolvida pelo Google, Skipfish, com o objetivo de obter informações mais detalhadas sobre as aplicações que executam dentro do Apache (ver figura 10 e 11). Incluiu-se a opção -o para que fosse gerado um relatório dinâmico em formato HTML para análise e identificação de eventuais configurações inseguras, problemas de conteúdo e identificação do versionamento das aplicações web disponibilizadas (ver figura 12).

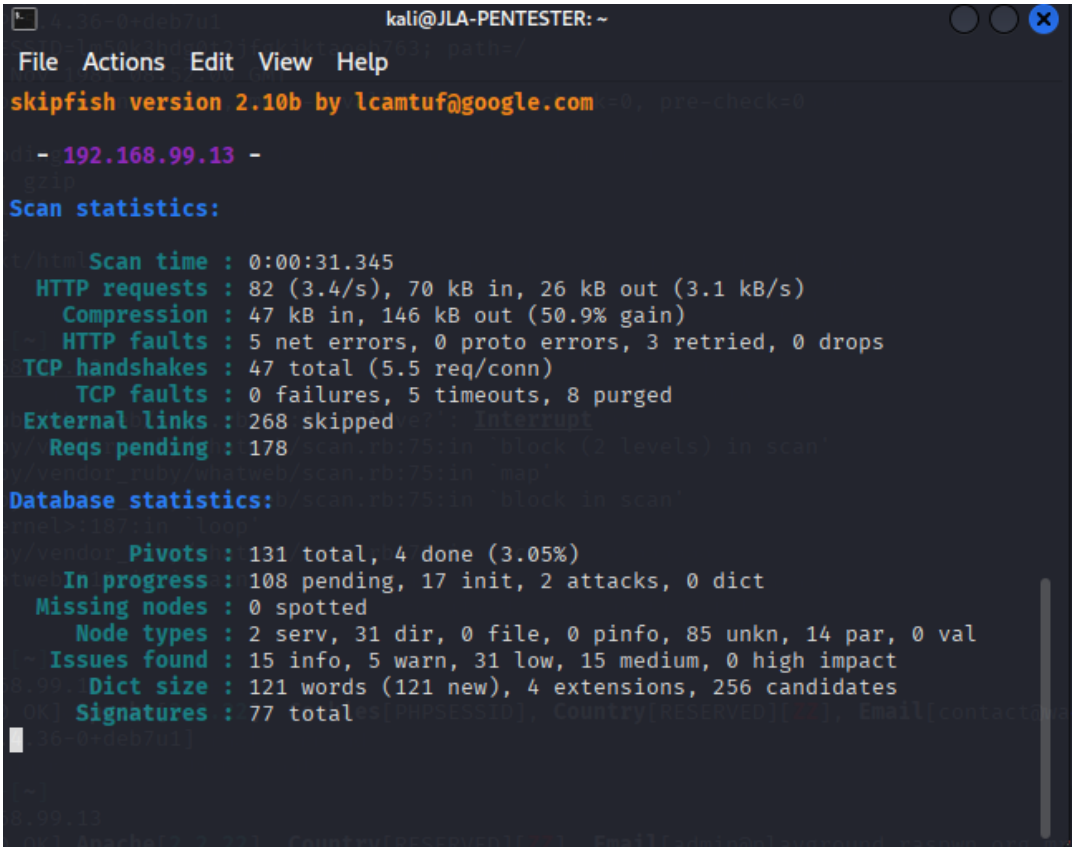

Figura 10 - Modo de visualização estatístico da aplicação skipfish em execução

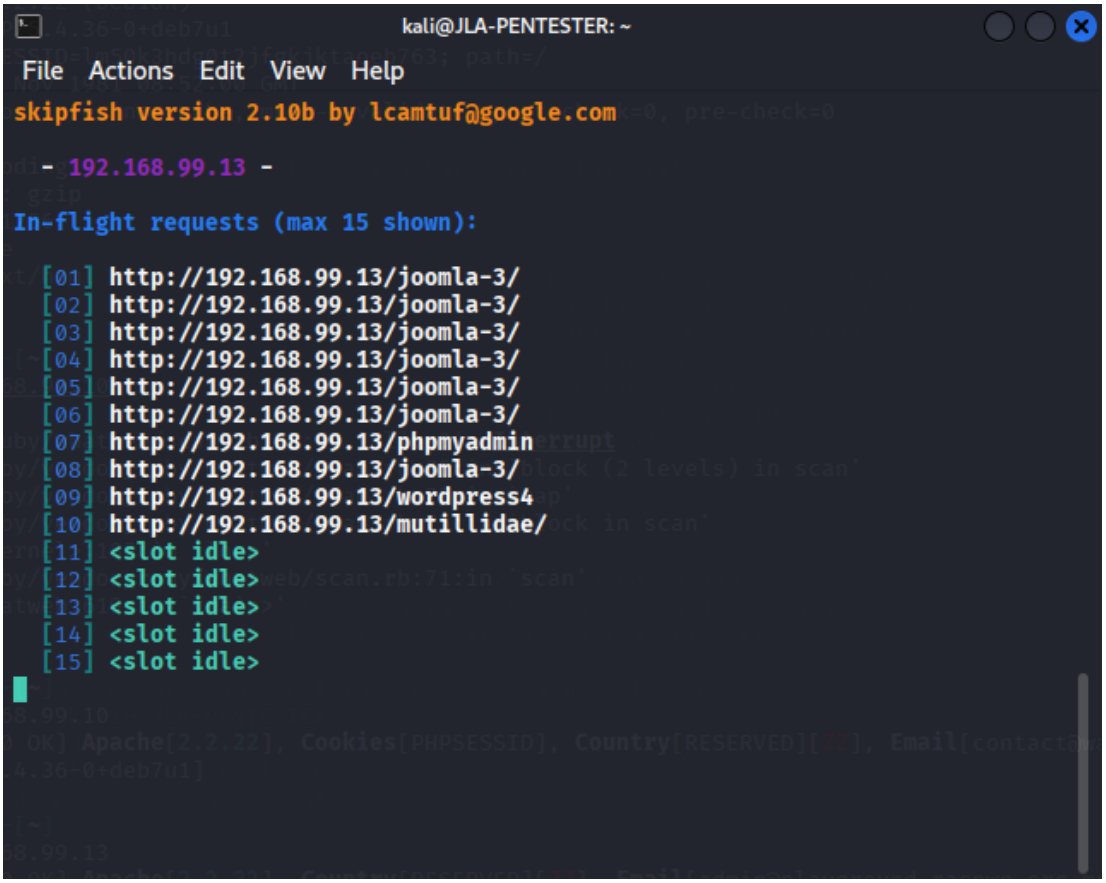

Figura 11 - Modo de visualização da descoberta de pastas da aplicação skipfish em execução

Foi possível identificar, ver figura 12, a execução de aplicações web, tais como: Drupal 7.34, phpMyAdmin, WordPress 3, Zen Cart, Mutillidae e Joomla 3. Sobre o Joomla 3, também foi possível identificar subpastas aplicacionais, uma delas, "*administrato*r", que corresponde ao painel de administração da aplicação. Dado que foi a aplicação onde foi possível encontrar seu versionamento, suspeitas de problemas de conteúdo e, eventualmente, configurações inseguras, ela foi selecionada para o capítulo seguinte.

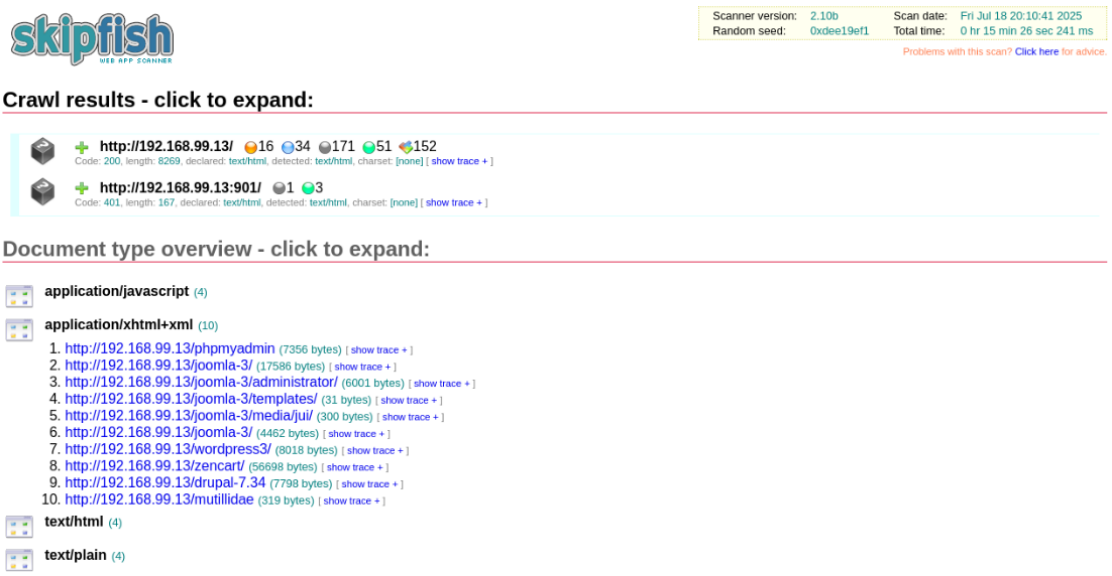

Figura 12 - Relatório em formato HTML gerado pela aplicação skipfish

4.4.3 EXPLORAÇÃO

Nesta fase, a utilização ativa de vulnerabilidades previamente identificadas é explorada através de técnicas específicas para comprometer sistemas e elevar privilégios. Ferramentas como o framework Metasploit automatizam a exploração, ao invés de abordagens manuais para manipulação de parâmetros em aplicações web, o que pode demonstrar a criticidade de configurações incongruentes. No Kali Linux, foi invocado o framework Metasploit através de sua interface principal, pelo comando msfconsole (figura 13).

Figura 13 - Carregamento do Framework Metasploit

A sua arquitetura modular e extensível, composta pela integração de vários componentes, carregados na inicialização da mesma, acelera uma análise de segurança e/ou vulnerabilidades e/ou um processo de penetração duma forma controlada.

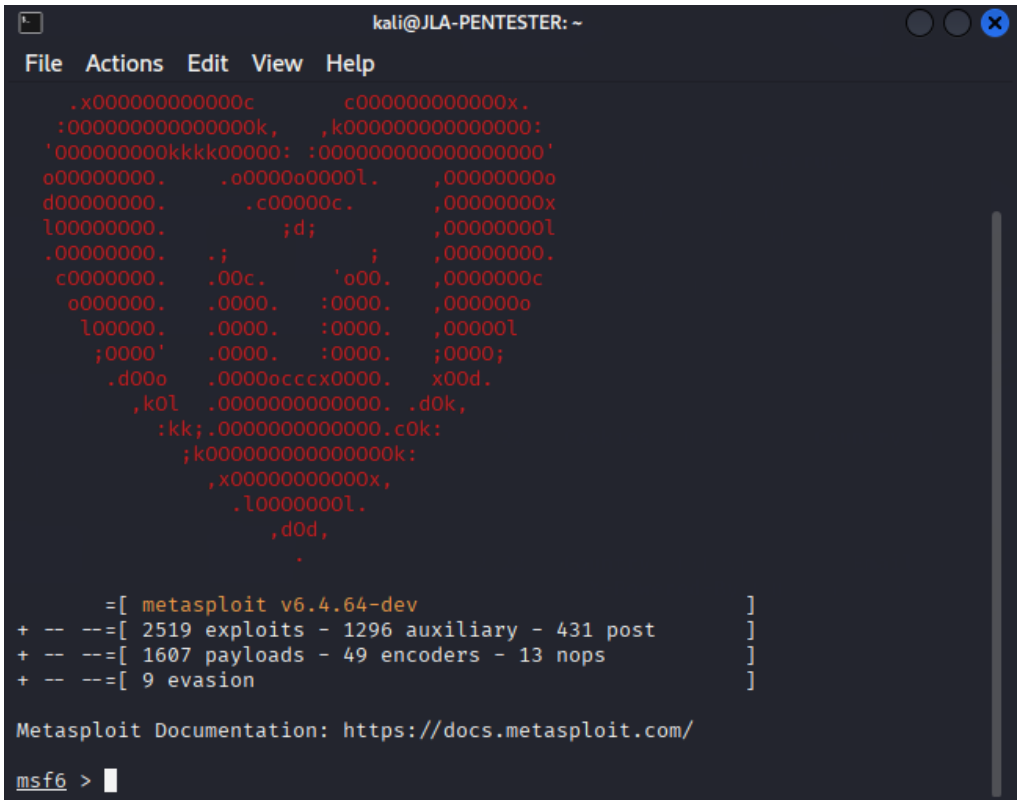

Figura 14 – Terminal do Framework Metasploit

Tabela 4 - Lista de Módulos disponíveis no metasploit com descritivos e exemplos

Tipo de Módulo	**Descrição**	**Exemplo**
Exploits	Códigos que exploram vulnerabilidades específicas.	exploit/multi/http/joomla_unserialize_rce
Payloads	Códigos executados após a exploração (ex: shells reversos).	windows/x64/meterpreter/reverse_tcp
Auxiliary	Ferramentas para varrimento, enumeração e ataques (DoS, brute force).	auxiliary/scanner/ssh/ssh_login

Post-Exploitation	Módulos para pós-exploração (recolha de dados, persistência).	post/windows/gather/hashdump
Encoders	Ofusca *payloads* para evadir a deteção por soluções de antivírus.	x86/shikata_ga_nai
NOPs (*No Operation*)	Gera instruções vazias para estabilizar *exploits*.	x86/single_byte
Evasion	Gera *payloads* ou técnicas para contornar soluções de segurança (AV, EDR, IPS).	evasion/windows/windows_defender_exe

De forma automática, é efetuado o carregamento das componentes apresentadas em cima e disponibiliza-se uma shell que permite inserir comandos para tirar partido das suas funcionalidades. Neste caso, foi efetuada uma pesquisa pelo nome da aplicação que havia sido identificada no capítulo anterior (Joomla) através do comando "search joomla". Essa procura devolveu 17 resultados relacionados com a aplicação, conforme figura 15.

```
msf6 > search joomla

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/scanner/http/joomla_gallerywd_sqli_scanner  2015-03-30       normal     No     Gallery WD for Joomla! Unauthenticated SQL Injection Scanner
   1   exploit/unix/webapp/joomla_tinybrowser                2009-07-22       excellent  Yes    Joomla 1.5.12 TinyBrowser File Upload Code Execution
   2   auxiliary/scanner/http/joomla_api_improper_access_checks  2023-02-01       normal     Yes    Joomla API Improper Access Checks
   3   auxiliary/admin/http/joomla_registration_privesc      2016-10-25       normal     Yes    Joomla Account Creation and Privilege Escalation
   4   exploit/unix/webapp/joomla_akeeba_unserialize         2014-09-29       excellent  Yes    Joomla Akeeba Kickstart Unserialize Remote Code Execution
   5   auxiliary/scanner/http/joomla_bruteforce_login        .                normal     No     Joomla Bruteforce Login Utility
   6   exploit/unix/webapp/joomla_comfields_sqli_rce         2017-05-17       excellent  Yes    Joomla Component Fields SQLi Remote Code Execution
   7   exploit/unix/webapp/joomla_comjce_imgmanager          2012-08-02       excellent  Yes    Joomla Component JCE File Upload Remote Code Execution
   8   exploit/unix/webapp/joomla_contenthistory_sqli_rce    2015-10-23       excellent  Yes    Joomla Content History SQLi Remote Code Execution
   9   exploit/multi/http/joomla_http_header_rce             2015-12-14       excellent  Yes    Joomla HTTP Header Unauthenticated Remote Code Execution
   10  exploit/unix/webapp/joomla_media_upload_exec          2013-08-01       excellent  Yes    Joomla Media Manager File Upload Vulnerability
   11  auxiliary/scanner/http/joomla_pages                   .                normal     No     Joomla Page Scanner
   12  auxiliary/scanner/http/joomla_plugins                 .                normal     No     Joomla Plugins Scanner
   13  auxiliary/gather/joomla_com_realestatemanager_sqli    2015-10-22       normal     Yes    Joomla Real Estate Manager Component Error-Based SQL Injection
   14  auxiliary/scanner/http/joomla_version                 .                normal     No     Joomla Version Scanner
   15  auxiliary/gather/joomla_contenthistory_sqli           2015-10-22       normal     Yes    Joomla com_contenthistory Error-Based SQL Injection
   16  auxiliary/gather/joomla_weblinks_sqli                 2014-03-02       normal     Yes    Joomla weblinks-categories Unauthenticated SQL Injection Arbitrary File Read
   17  auxiliary/scanner/http/joomla_ecommercewd_sqli_scanner  2015-03-20       normal     No     Web-Dorado ECommerce WD for Joomla! search_category_id SQL Injection Scanner

Interact with a module by name or index. For example info 17, use 17 or use auxiliary/scanner/http/joomla_ecommercewd_sqli_scanner

msf6 >
```

Figura 15 - Resultados da pesquisa efetuada pela aplicação joomla

Foi selecionada a opção número nove dos resultados apresentados, comando "use 9", ou seja, o carregamento do módulo (exploit/multi/http/joomla_http_header_rce) que

visa explorar uma vulnerabilidade de Execução Remota de Código (RCE) em versões vulneráveis da aplicação Joomla através da manipulação de cabeçalhos HTTP.

Após o carregamento do módulo é apresentada informação relacionada com o mesmo, neste caso, o *exploit*. Destaca-se a informação sobre o intervalo de versões da aplicação que estão vulneráveis a este tipo de ataque. Neste caso, o *exploit* pode ser executado com sucesso, se a aplicação se encontrar em execução entre a versão 1.5.0 e a 3.4.5, conforme descrito na Figura 16.

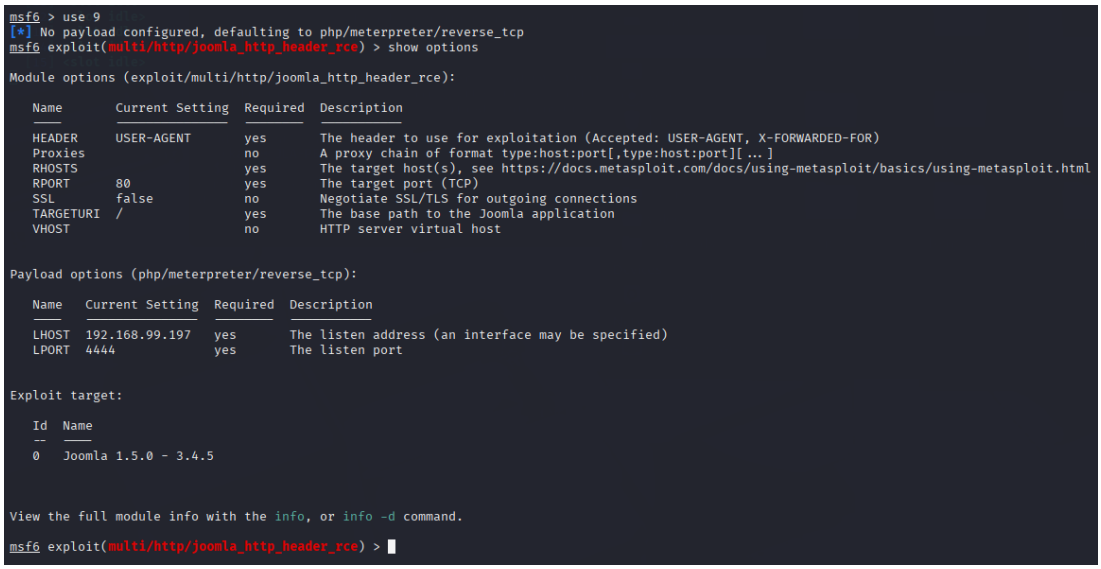

Figura 16 - Opções e informações do exploit

A estrutura padronizada do Metasploit para configuração de *exploits*, *payloads* e opções permite visualizar opções organizadas em categorias e identificar quais são as opções requeridas, bem como qual é a configuração necessária para que seja possível executar o mesmo. Foi definido o campo alvo, através do comando "set RHOSTS 192.168.99.13" e o diretório pai onde o alvo tem a aplicação em execução, através do comando "set TARGETURI /joomla-3" (Figura 17).

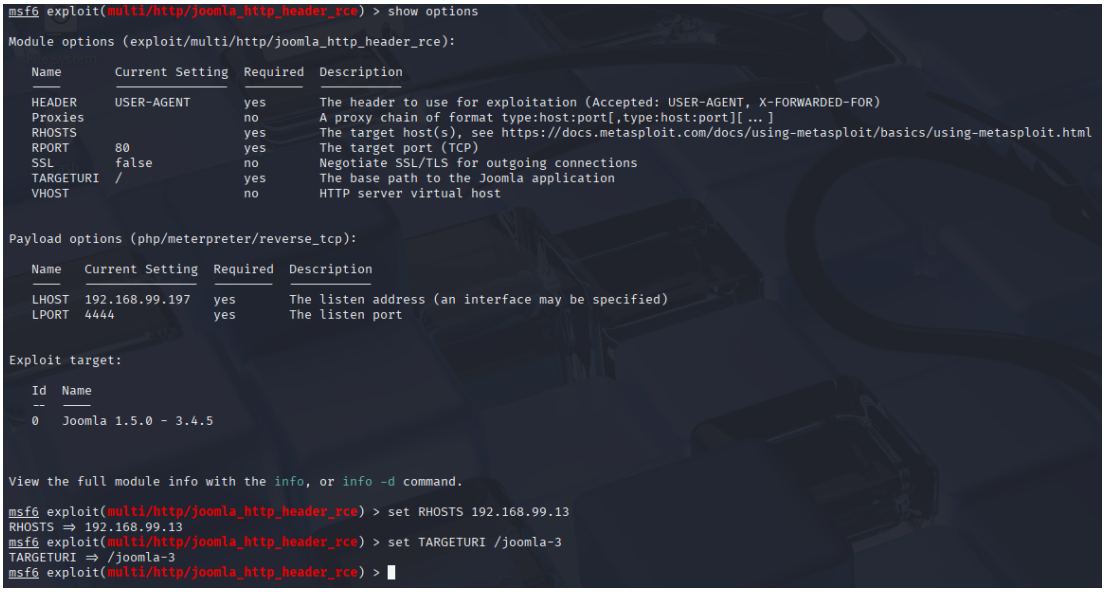

Figura 17 - Configuração do exploit

Terminada a configuração, partiu-se para a execução deste através do comando "exploit", conforme Figura 18.

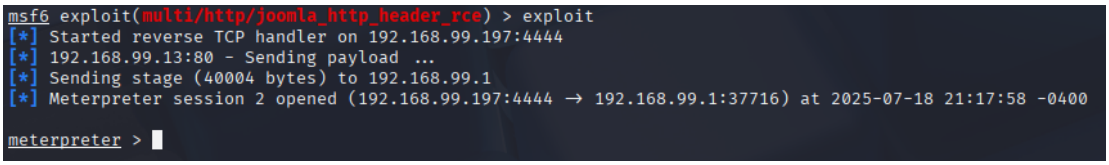

Figura 18 - Execução do exploit, carregamento do payload e receção de terminal remoto

O despoletar da execução do exploit significou a inicialização de um *listener* na porta 4444 do IP 192.168.99.197 (máquina do cibe criminoso) para receber a conexão reversa do alvo. Como ação, o exploit enviou um *payload* malicioso para o servidor vulnerável (192.168.99.13:80) na porta do serviço.

O alvo conectou-se à máquina do cibe criminoso, sendo fornecido o acesso a um shell interativo (meterpreter), que opera em memória para evitar a deteção por soluções de antivírus e que disponibiliza recursos para uma exploração eficaz, persistência de acesso e lateralização.

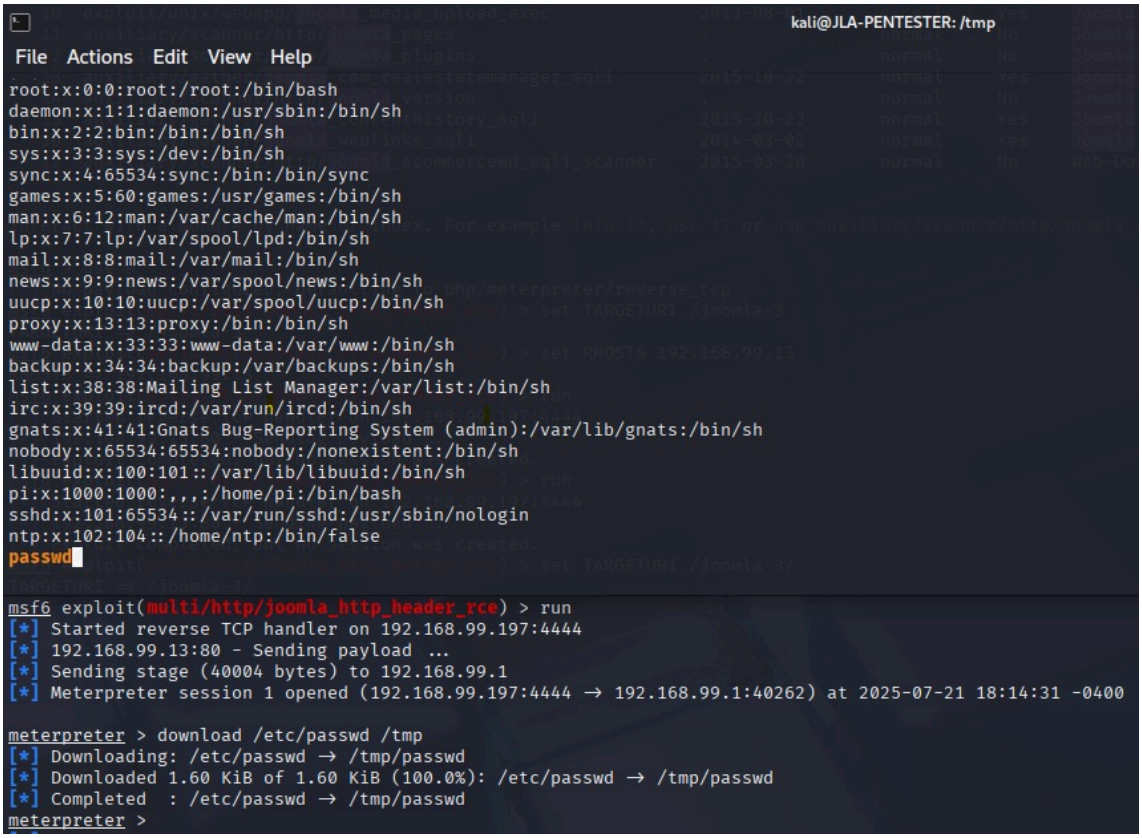

Figura 19 - Exfiltração de ficheiro e visualização do seu conteúdo

A execução do comando "download /etc/passwd /tmp" no Meterpreter foi uma ação crítica em operações de exfiltração de dados. Foi possível transferir informações sensíveis do sistema comprometido para a máquina Kali.

Tabela 5 - Identificação e significado do comando de exfiltração

Componente	Significado	Impacto
download	Instrui o Meterpreter a transferir um ficheiro do alvo para o cibe criminoso.	Permite roubo de dados confidenciais.
/etc/passwd	Ficheiro do Linux que lista todos os utilizadores do sistema (incluindo UIDs/GIDs).	Expõe estrutura de utilizadores, facilitando ataques de força bruta ou escalonamento.
/tmp/	Diretório local (do cibe criminoso) onde o ficheiro será copiado.	Local temporário para análise posterior (ex: extração de hashes).

Esta abordagem trifásica, assente em princípios técnicos sólidos e procedimentos estruturados, permitiu uma avaliação abrangente da postura de segurança da infraestrutura laboratorial, destacando-se pela sua eficácia na identificação proativa de vulnerabilidades críticas.
Verificou-se também a eficácia deste exploit contra instalações vulneráveis da aplicação Joomla compreendidas no intervalo de versões 1.5.0 a 3.4.5.

5 - ANÁLISE E DISCUSSÃO DE RESULTADOS

Esta seção apresenta uma análise crítica dos resultados obtidos durante a exploração da vulnerabilidade RCE no ambiente Raspwn, com foco na aplicação Joomla e no seu impacto, como uma exfiltração de dados via Meterpreter.

A exploração de vulnerabilidades do tipo RCE representa um dos principais riscos quando se fala de ameaças cibernéticas. Os resultados são discutidos à luz de teorias de segurança cibernética, boas práticas e limitações do ambiente controlado.

Esta análise reforça a importância desses ambientes contidos para exercícios práticos de cibersegurança, que combinam vulnerabilidades controladas com desafios realistas, preparando profissionais da área para desafios complexos de segurança cibernética.

Com base em dados empíricos e investigação recente, são abordados os desafios, os potenciais impactos, as técnicas vanguardistas de exploração, assim como estratégias para efetuar a mitigação da ameaça ou do incidente.

A exploração da vulnerabilidade RCE na aplicação Joomla foi efetuada através de um método de serialização insegura de cookies (PHPSESSID) para execução remota de código, resultando em um ganho de sessão privilegiada via Meterpreter, e possibilitando

a exfiltração de dados críticos, como o ficheiro /etc/passwd, que foi copiado para a máquina do cibe criminoso.

Fatores como a configuração insegura do Joomla, a falta de atualização (a versão explorada é vulnerável por não aplicar patches de serialização segura e por APIs expostas que permitiram acesso não autenticado) facilitaram a exploração.

A simplificação do ambiente devido à ausência de mecanismos de proteção, tais como a falta de logs em tempo real, componentes como WAF e/ou EDR, assim como permissões excessivas que permitiram, em modo leitura, o acesso público a ficheiros fundamentais do sistema (/etc/passwd).

A utilização de ferramentas eficazes, como Metasploit/Meterpreter, automatizou a exploração e a pós-exploração através de comunicação criptografada e execução em memória.

O impacto simulado pode ser extrapolado para um cenário real, como a exfiltração de dados sensíveis e o comprometimento de contas privilegiadas.

Lições-chave, como a priorização da aplicação célere de correções às aplicações após sua disponibilização, o hardening de serviços, como o restringimento de permissões em ficheiros sensíveis, bem como uma monitorização contínua, são essenciais para detetar explorações.

6 LIMITAÇÕES

Ambiente laboratorial não simula latências de rede semelhantes a redes do tipo WAN ou em evasões avançadas. Vulnerabilidades previsíveis através de falhas colocadas deliberadamente.

Complexidade reduzida, por inexistência de componentes tais como WAF, EDR e segmentação de rede.

Configurações simplificadas devido à ausência de configurações de *hardening*. Dados sintéticos por ausência de proteção através de SELinux/AppArmor.

7 BOAS PRÁTICAS DE CONFIGURAÇÃO E MITIGAÇÃO

Desativação de interfaces de gestão remota (/administrator/*) ou protegê-las com MFA. Implementação de *Web Application Firewall* que permitirá o bloqueio geográfico, higienização de ligações, aprendizagem de navegação e proteção contra *zero days*.

Salvaguarda da informação, com agendamento recursivo, para tipos de dispositivos não acessíveis através da rede (por exemplo, Tapes), que em caso de utilização se encontram limpas de adulteração maliciosa e permitem o restauro integral dos dados.

Instalação de todas as correções disponíveis de vulnerabilidades conhecidas, quer ao nível aplicacional (incluindo dependências, *plugins*, *frameworks*, bibliotecas) como na infraestrutura base (sistema operativo e serviços de rede disponibilizados).

Instalação de soluções de antivírus / EDR e configuração de firewall do tipo de filtragem de pacotes; instalação de soluções de IDS + IPS para deteção e bloqueio de atividade suspeita.

Segmentação de rede como estratégia para reduzir a superfície de ataque e limitar o movimento lateral de ameaças; proteger APIs e endpoints através da implementação de controlos de acesso granular (autorização) e mecanismos robustos de autenticação (por exemplo, OAuth 2.0 + OpenID Connect (OIDC)); analisar e correlacionar *logs* de segurança, aplicação, rede e sistemas;

Envio de logs para plataforma externa (Syslog / SIEM);

Aplicar permissões restritas (princípio do privilégio mínimo).

Configuração de *allowlists* (lista de valores permitidos) em vez de *blocklists*;

Configuração de monitorização continua e proativa;

Desativar funções aplicacionais consideradas perigosas;

Desativar funções / serviços configurados por predefinição, mas que não são necessários;

Configuração de microserviços com responsabilidade única e com deteção de comportamento anómalos em (Falco em *container)*

8 CONCLUSÃO

A exploração de vulnerabilidades do tipo RCE é um processo metódico, com elevado impacto e que exige medidas robustas de mitigação. Desde o reconhecimento potencial até à exfiltração de dados, a compreensão total deste é essencial para a elaboração de estratégias eficazes que permitam a implementação de uma defesa cibernética ativa, sustentada nas boas práticas da indústria e na redução da "superfície de ataque".

A presente investigação permitiu confirmar que as vulnerabilidades de Remote Code Execution (RCE) continuam a representar uma das ameaças mais críticas no atual panorama de cibersegurança. A análise realizada demonstrou que estes vetores de ataque se caracterizam por um tempo de exploração extremamente reduzido e uma estreita correlação com deficiências nos processos de gestão de *patches* e atualizações de segurança.

O estudo evidenciou que a natureza particularmente severa das explorações RCE decorre da sua capacidade única de permitir a execução remota de código arbitrário em sistemas-alvo, frequentemente com privilégios elevados. As consequências destes compromissos são invariavelmente graves, podendo incluir o controlo total de sistemas, exfiltração massiva de dados sensíveis ou mesmo a instalação persistente de *backdoors* e *ransomware*.

Perante este cenário, a investigação permitiu identificar um conjunto de boas práticas e medidas técnicas que se podem revelar particularmente eficazes na mitigação destas ameaças. A monitorização contínua, assente em soluções SIEM para agregação e correlação centralizada de logs, mostra-se essencial para uma deteção proativa. Complementarmente, a análise detalhada do tráfego de rede e a monitorização granular de aplicações web constituem elementos críticos numa estratégia de defesa em profundidade.

No plano técnico, a implementação de *Web Application Firewall* (WAF) com capacidades comportamentais, conjugada com a utilização de containers configurados

segundo o princípio do privilégio mínimo, pode revelar-se particularmente eficaz na contenção de explorações RCE. Igualmente determinante é a aplicação rigorosa de hardening em sistemas e aplicações, eliminando serviços desnecessários e reduzindo, assim, a superfície de ataque.

A dimensão humana e organizacional não deve ser negligenciada. A formação contínua das equipas técnicas, a realização regular de testes de penetração (preferencialmente com validação externa) e a manutenção atualizada de planos de resposta a incidentes surgem como elementos indispensáveis numa postura de segurança proativa. Particular atenção deve ser dada aos processos de salvaguarda e recuperação, que funcionam como último reduto de defesa perante um ataque bem-sucedido.

Importa sublinhar que nenhuma das medidas isoladamente oferece proteção absoluta. A estratégia mais eficaz passa necessariamente pela conjugação sinérgica de soluções tecnológicas avançadas com processos bem definidos e equipas devidamente capacitadas. A adoção de uma abordagem em camadas, que incorpore princípios de *security-by-design* desde a fase de conceção dos sistemas, mostra-se particularmente promissora.

Embora o presente trabalho tenha permitido avançar significativamente na compreensão e mitigação das ameaças RCE, identificaram-se áreas que merecem investigação futura. Destacam-se a necessidade de desenvolver métricas quantitativas para avaliação de posturas de segurança e a realização de estudos setoriais comparativos que possam informar estratégias específicas para diferentes domínios organizacionais.

Em síntese, face à sofisticação crescente das ameaças RCE e à sua potencial gravidade, as organizações devem encarar a cibersegurança como um investimento estratégico contínuo. A implementação sistemática das medidas aqui propostas, adaptadas às especificidades de cada contexto operacional, permitirá reduzir significativamente os riscos e fortalecer a resiliência organizacional num panorama de ameaças em constante evolução.

BIBLIOGRAFIA

[1] Leszczyna, R. (2021). Review of cybersecurity assessment methods: Applicability perspective. Computers & Security, 108, 102376. https://doi.org/10.1016/j.cose.2021.102376

[2] Brewer, R. (2016). Ransomware attacks: Detection, prevention and cure. Network Security, 2016(9), 5-9. https://doi.org/10.1016/S1353-4858(16)30086-1

[3] Bier, S., Fajardo, B., Ezeadum, O., Guzman, G., Sultana, K. Z., & Anu, V. (2021). Mitigating remote code execution vulnerabilities: A study on Tomcat and Android security updates. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) [9422666]. IEEE. https://doi.org/10.1109/IEMTRONICS52119.2021.9422666

[4] Sabottke, C., Suciu, O., & Dumitras, T. (2015). Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting real-world exploits. In 24th USENIX Security Symposium (pp. 1041-1056). USENIX Association.

[5] ENISA. (2022). ENISA threat landscape 2022. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022

[6] Imperva. (2023, December 20). Remote code execution (RCE): Types, examples & mitigation. Imperva Learning Center. https://www.imperva.com/learn/application-security/remote-code-execution/

[7] National Institute of Standards and Technology. (2018). Framework for improving critical infrastructure cybersecurity (Version 1.1). https://doi.org/10.6028/NIST.CSWP.04162018

[8] Wichers, D., & Williams, J. (2010). OWASP Top-10 2010: The ten most critical web application security risks. OWASP Foundation. https://owasp.org/www-pdf-archive/OWASP_Top_10_-_2010.pdf

[9] Husák, M., & Sadlek, L. (2025). Attack surface management: State of the art and operational challenges. In 2025 IEEE 11th International Conference on Network Softwarization (NetSoft) (pp. 1-6). IEEE. https://doi.org/10.1109/NetSoft64993.2025.11080588

[10] Kim, M., Jang, H., & Shin, Y. (2022). Avengers, assemble! Survey of WebAssembly security solutions. In 2022 IEEE International Conference on Cloud Computing (CLOUD) (pp. 294-304). IEEE. https://doi.org/10.1109/CLOUD55607.2022.00051

[11] Ladisa, P., Plate, H., Martinez, M., & Bartel, A. (2023). SoK: Taxonomy of attacks on open-source software supply chains. In 2023 IEEE Symposium on Security and Privacy (SP) (pp. 1509-1526). IEEE. https://doi.org/10.1109/SP46215.2023.10179304

[12] Nobles, C. (2018). Botching human factors in cybersecurity in business organizations. Holistica – Journal of Business and Public Administration, 9(3), 71-88. https://doi.org/10.2478/hjbpa-2018-0024

[13] Huang, H.-C., Zhang, Z.-K., Cheng, H.-W., & Shieh, S. (2017). Web application security: Threats, countermeasures, and pitfalls. *Computer*, *50*(6), 81-85. https://doi.org/10.1109/MC.2017.183

[14] SQLmap Developers, "SQLmap: Automatic SQL Injection Tool," [Online]. Available: http://sqlmap.org

[15] PortSwigger, "Burp Suite Professional," [Online]. Available: https://portswigger.net/burp

[16] Center for Internet Security. (2022). *CIS Benchmarks*. https://www.cisecurity.org/cis-benchmarks

[17] Xiao, F., Yang, Z., Allen, J., Yang, G., Williams, G., & Lee, W. (2022). Understanding and mitigating remote code execution vulnerabilities in cross-platform ecosystem. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2975-2988). Association for Computing Machinery. https://doi.org/10.1145/3548606.3559340

[18] Ahmad, I., Alsmadi, I., Kumar, R., & Jarray, A. (2023). Privilege escalation attack detection and mitigation in cloud using machine learning. *IEEE Access*, *11*, 49390-49401. https://doi.org/10.1109/ACCESS.2023.3276896

[19] Kaspersky. (2022). *Common TTPs of modern ransomware groups*. https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2022/06/23093553/Common-TTPs-of-the-modern-ransomware_low-res.pdf

[20] Böhme, M., Pham, V.-T., & Roychoudhury, A. (2019). Coverage-based greybox fuzzing as markov chain. *IEEE Transactions on Software Engineering*, *45*(5), 489-506. https://doi.org/10.1109/TSE.2017.2785841

[21] Rajivan, P., & Gonzalez, C. (2018). Creative persuasion: A study on adversarial behaviors and strategies in phishing attacks. *Frontiers in Psychology*, *9*, 135. https://doi.org/10.3389/fpsyg.2018.00135

[22] ISO/IEC. (2022). Information security, cybersecurity and privacy protection — Information security management systems — Requirements (ISO/IEC 27001:2022). https://www.iso.org/standard/27001

[23] Holm, H., Sommestad, T., Almroth, J., & Persson, M. (2015). A quantitative evaluation of vulnerability scanning. *Information Management & Computer Security*, *23*(4), 388-413. https://doi.org/10.1108/IMCS-02-2015-0065

[24] Kenneally, E., & Bailey, M. (2013). Cyber-security research ethics dialogue & strategy workshop. *Computing Research Repository (CoRR)*, abs/1401.2378. https://www.caida.org/publications/papers/2013/cyber_security_research_ethics/cyber_security_research_ethics.pdf

[25] Prandini, M., & Ramilli, M. (2012). Return-oriented programming. IEEE Security & Privacy, 10(6), 84-87. https://doi.org/10.1109/MSP.2012.154

[26] BSIMM, "Building Security In Maturity Model," Synopsys, Tech. Rep. v12, 2022.

[27] Vieira, M., Antunes, N., & Madeira, H. (2009). Using web security scanners to detect vulnerabilities in web services. In 2009 IEEE/IFIP International Conference on Dependable Systems & Networks (pp. 566-571). IEEE. https://doi.org/10.1109/DSN.2009.5270346

[28] EU, "General Data Protection Regulation (GDPR)," Official Journal of the EU, 2016/679, 2016.

[29] Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). Technical guide to information security testing and assessment (NIST Special Publication 800-115). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-115

[30] Rapid7, "Metasploit Framework Documentation," [Online]. Available: https://metasploit.com

[31] Xiao, F., Yang, Z., Allen, J., Yang, G., Williams, G., & Lee, W. (2022). Understanding and mitigating remote code execution vulnerabilities in cross-platform ecosystem. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (pp. 2975-2988). ACM. https://doi.org/10.1145/3548606.3559340

[32] Bier, S., Fajardo, B., Ezeadum, O., Guzman, G., Sultana, K. Z., & Anu, V. (2021). Mitigating remote code execution vulnerabilities: A study on Tomcat and Android security updates. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). IEEE. https://doi.org/10.1109/IEMTRONICS52119.2021.9422666

[33] ENISA. (2023). ENISA Threat Landscape 2023. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[34] Verizon, "2023 Data Breach Investigations Report", Verizon Communications Inc., Tech. Rep., 2023.

[35] Vieira, M., Antunes, N., & Madeira, H. (2009). Using web security scanners to detect vulnerabilities in web services. In 2009 IEEE/IFIP International Conference on Dependable Systems & Networks (pp. 566-571). IEEE. https://doi.org/10.1109/DSN.2009.5270346

[36] OWASP Foundation. (2023). OWASP API Security Top 10 - 2023. https://owasp.org/API-Security/editions/2023/en/0x11-t10/

[37] Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). Technical guide to information security testing and assessment (NIST Special Publication 800-115). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-115

[38] Hao, S., Borgolte, K., Nikiforakis, N., Stringhini, G., Egele, M., Eubanks, M., Krebs, B., & Vigna, G. (2016). PREDATOR: Proactive recognition and elimination of domain abuse at time-of-registration. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 1568-1579). ACM. https://doi.org/10.1145/2976749.2978317

[39] Stock, B., Lekies, S., Mueller, T., Spiegel, P., & Johns, M. (2014). Precise client-side protection against DOM-based cross-site scripting. In 23rd USENIX Security Symposium (pp. 655-670). USENIX Association.

[40] OWASP, "Top 10 Web Application Security Risks", OWASP Foundation, 2021.

[41] Seacord, R. C. (2013). Secure coding in C and C++ (2nd ed.). Addison-Wesley Professional.

[42] Nunes, P., Medeiros, I., Fonseca, J., Neves, N., Correia, M., & Vieira, M. (2015). An empirical study on combining diverse static analysis tools for web security vulnerabilities based on development scenarios. In 2015 IEEE/ACM 13th International Conference on Mining Software Repositories (pp. 488-491). IEEE. https://doi.org/10.1109/MSR.2015.69

[43] Souppaya, M., & Scarfone, K. (2023). Guide to enterprise patch management planning (NIST Special Publication 800-40 Rev. 4). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-40r4

[44] Pashchenko, I., Plate, H., Ponta, S. E., Sabetta, A., & Massacci, F. (2018). Vulnerable open source dependencies: Counting those that matter. In Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (Article 42). ACM. https://doi.org/10.1145/3239235.3268920

[45] ENISA. (2023). ENISA threat landscape 2023. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[46] Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., & Halderman, J. A. (2015). A search engine backed by Internet-wide scanning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 542-553). ACM. https://doi.org/10.1145/2810103.2813703

[47] ISO/IEC, "ISO 27002:2022 - Security Controls", 2022.

[48] J. Anderson, "Security Engineering: A Guide to Building Dependable Distributed Systems", 3ª ed., Wiley, 2020.

[49] Penetration Testing Execution Standard (PTES). (2014). The penetration testing execution standard. http://www.pentest-standard.org/

[50] Böhme, M., Pham, V.-T., & Roychoudhury, A. (2019). Coverage-based greybox fuzzing as Markov chain. IEEE Transactions on Software Engineering, 45(5), 489-506. https://doi.org/10.1109/TSE.2017.2785841

[51] ENISA. (2021). Penetration testing: Operational guidance for the EU institutions and agencies. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/penetration-testing

[52] Mao, J., Bian, J., Ryoo, J., Marupudi, V., Nejad, P. E., & Liu, P. (2018). Passive reconnaissance of network topology and device fingerprinting in industrial control systems. In 2018 Resilience Week (RWS) (pp. 1-6). IEEE. https://doi.org/10.1109/RWEEK.2018.8473524

[53] Shodan, "Guide to Search Engine for Internet-Connected Devices", 2023. [Online]. Disponível: https://www.shodan.io

[54] Censys, "Network Scanning and Discovery Platform", 2023. [Online]. Disponível: https://censys.io

[55] Ghafir, I., Prenosil, V., Hammoudeh, M., Han, L., Rabie, K., & Aparicio-Navarro, F. J. (2019). BotDet: A system for real time botnet command and control traffic detection. IEEE Access, 7, 38947-38958. https://doi.org/10.1109/ACCESS.2019.2905917

[56] ENISA. (2022). Open source intelligence (OSINT) for cyber security. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/topics/threat-risk-management/intelligence

[57] Samtani, S., Chinn, R., Chen, H., & Nunamaker Jr, J. F. (2017). Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence. Journal of Management Information Systems, 34(4), 1023-1053. https://doi.org/10.1080/07421222.2017.1394049

[58] Lyon, G. F. (2009). Nmap network scanning: The official Nmap project guide to network discovery and security scanning. Insecure.

[59] OWASP Foundation. (2021). OWASP Web Security Testing Guide (WSTG) version 4.2. https://owasp.org/www-project-web-security-testing-guide/

[60] Quinn, S., Souppaya, M., Cook, M., & Scarfone, K. (2018). National checklist program for IT products: Guidelines for checklist users and developers (NIST Special Publication 800-70 Rev. 4). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-70r4

[61] Weidman, G. (2014). Penetration testing: A hands-on introduction to hacking. No Starch Press.

[62] Durumeric, Z., Wustrow, E., & Halderman, J. A. (2013). ZMap: Fast Internet-wide

scanning and its security applications. In Proceedings of the 22nd USENIX Security Symposium (pp. 605-620). USENIX Association.
[63] Bier, S., Fajardo, B., Ezeadum, O., Guzman, G., Sultana, K. Z., & Anu, V. (2021). Mitigating remote code execution vulnerabilities: A study on Tomcat and Android security updates. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). IEEE. https://doi.org/10.1109/IEMTRONICS52119.2021.9422666
[64] CISA, FBI, NSA. (2021). Mitigating Log4Shell and other Log4j-related vulnerabilities (Cybersecurity Advisory AA21-356A). https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-356a
[65] OWASP Foundation. (2023). OWASP API Security Top 10 - 2023. https://owasp.org/API-Security/editions/2023/en/0x11-t10/
[66] Pashchenko, I., Plate, H., Ponta, S. E., Sabetta, A., & Massacci, F. (2018). Vulnerable open source dependencies: Counting those that matter. In Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ACM. https://doi.org/10.1145/3239235.3268920
[67] Rapid7, "Metasploit Framework Documentation" [Online]. Disponível: https://metasploit.com
[68] Prandini, M., & Ramilli, M. (2012). Return-oriented programming. IEEE Security & Privacy, 10(6), 84-87. https://doi.org/10.1109/MSP.2012.154
[69] Xiao, F., Yang, Z., Allen, J., Yang, G., Williams, G., & Lee, W. (2022). Understanding and mitigating remote code execution vulnerabilities in cross-platform ecosystem. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (pp. 2975-2988). ACM. https://doi.org/10.1145/3548606.3559340
[70] Kennedy, D., O'Gorman, J., Kearns, D., & Aharoni, M. (2011). Metasploit: The penetration tester's guide. No Starch Press.
[71] ENISA. (2023). ENISA threat landscape 2023. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023
[72] Carrier, B. (2005). File system forensic analysis. Addison-Wesley Professional.
[73] Blunden, B. (2013). The rootkit arsenal: Escape and evasion in the dark corners of the system (2nd ed.). Jones & Bartlett Learning.
[74] Johnson, C., Badger, L., Waltermire, D., Snyder, J., & Skorupka, C. (2016). Guide to cyber threat information sharing (NIST Special Publication 800-150). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-150
[75] Sharma, A., Sahay, S. K., & Kumar, A. (2020). Lateral movement detection in industrial

control systems using network flow data. IEEE Transactions on Industrial Informatics, 16(10), 6508-6516. https://doi.org/10.1109/TII.2020.2972070

[76] Maynor, D. (2007). Metasploit toolkit for penetration testing, exploit development, and vulnerability research. Syngress.

[77] Kennedy, D., O'Gorman, J., Kearns, D., & Aharoni, M. (2011). Metasploit: The penetration tester's guide. No Starch Press.

[78] OWASP Foundation. (2021). OWASP Web Security Testing Guide (WSTG) version 4.2. https://owasp.org/www-project-web-security-testing-guide/

[79] Prandini, M., & Ramilli, M. (2012). Return-oriented programming. IEEE Security & Privacy, 10(6), 84-87. https://doi.org/10.1109/MSP.2012.154

[80] Booth, H., Rike, D., & Witte, G. A. (2013). The National Vulnerability Database (NVD): Overview (NISTIR 7946). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.7946

81] PortSwigger, "Burp Suite Documentation", 2023. [Online]. Disponível: https://portswigger.net/burp/documentation

[82] Vieira, M., Antunes, N., & Madeira, H. (2009). Using web security scanners to detect vulnerabilities in web services. In 2009 IEEE/IFIP International Conference on Dependable Systems & Networks (pp. 566-571). IEEE. https://doi.org/10.1109/DSN.2009.5270346

[83] OWASP Foundation. (2021). OWASP Web Security Testing Guide (WSTG) version 4.2. https://owasp.org/www-project-web-security-testing-guide/

[84] Stock, B., Lekies, S., Mueller, T., Spiegel, P., & Johns, M. (2014). Precise client-side protection against DOM-based cross-site scripting. In 23rd USENIX Security Symposium (pp. 655-670). USENIX Association.

[85] OWASP Zed Attack Proxy (ZAP). https://www.zaproxy.org/docs/

[86] Damele, B., & Stampar, M. (n.d.). SQLmap: Automatic SQL injection and database takeover tool. Retrieved [data de acesso], from https://sqlmap.org/

[87] Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). Technical guide to information security testing and assessment (NIST Special Publication 800-115). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-115

[88] Diogenes, Y., & Ozkaya, E. (2018). Cybersecurity – Attack and defense strategies: Infrastructure security with Red Team and Blue Team tactics. Packt Publishing.

[89] Souppaya, M., & Scarfone, K. (2023). Guide to enterprise patch management planning (NIST Special Publication 800-40 Rev. 4). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-40r4

[90] MITRE, "CVE Documentation", 2023. [Online]. Disponível: https://cve.mitre.org

[91] NVD, "National Vulnerability Database Technical Guide", 2023.
[92] Offensive Security, "Exploit Database Documentation", 2023.
[93] Apache Foundation, "Security Advisory CVE-2025-24813", 2025.
[94] Allodi, L., & Massacci, F. (2014). Comparing vulnerability severity and exploits using case-control studies. ACM Transactions on Information and System Security, 17(1), Article 1. https://doi.org/10.1145/2630069
[95] Sabottke, C., Suciu, O., & Dumitras, T. (2015). Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting real-world exploits. In 24th USENIX Security Symposium (pp. 1041-1056). USENIX Association.
[96] Samtani, S., Chinn, R., Chen, H., & Nunamaker Jr, J. F. (2017). Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence. Journal of Management Information Systems, 34(4), 1023-1053. https://doi.org/10.1080/07421222.2017.1394049
[97] Milo, C., Bissyandé, T. F., Klein, J., & Le Traon, Y. (2021). Is GitHub's Copilot as bad as humans at introducing vulnerabilities in code? Empirical Software Engineering, 28, Article 129. https://doi.org/10.1007/s10664-023-10380-1 2023.
[98] Arce, I., Clark, S., Mohindra, A., & McGraw, G. (2004). An analysis of the Slammer worm. IEEE Security & Privacy, 2(1), 82-87. https://doi.org/10.1109/MSECP.2004.1264877
[99] Wilhelm, T. (2010). Professional penetration testing: Creating and learning in a hacking lab. Syngress.
[100] MITRE Corporation. (n.d.). MITRE ATT&CK: Persistence tactics (TA0003). Retrieved December 3, 2024, from https://attack.mitre.org/tactics/TA0003/
[101] ENISA. (2023). ENISA threat landscape 2023. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023
[102] Xiao, F., Yang, Z., Allen, J., Yang, G., Williams, G., & Lee, W. (2022). Understanding and mitigating remote code execution vulnerabilities in cross-platform ecosystem. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (pp. 2975-2988). ACM. https://doi.org/10.1145/3548606.3559340
[103] U.S. Government Accountability Office. (2018). Data protection: Actions taken by Equifax and federal agencies in response to the 2017 breach (Report No. GAO-18-559). https://www.gao.gov/products/gao-18-559
[104] Apache Software Foundation. (2021). Apache Log4j security vulnerabilities. https://logging.apache.org/log4j/2.x/security.html
[105] IBM Security. (2023). Cost of a data breach report 2023. https://www.ibm.com/reports/data-breach

[106] Verizon. (2023). 2023 Data Breach Investigations Report. https://www.verizon.com/business/resources/reports/dbir/

[107] Ponemon Institute. (2023). The cyber resilient organization. IBM Security. https://www.ibm.com/reports/cyber-resilient-organization

[108] Edelman. (2023). 2023 Edelman Trust Barometer: Navigating a polarized world. https://www.edelman.com/trust/2023/trust-barometer

[109] Reputation Institute. (2023). Global RepTrak 100: The world's most reputable companies. https://www.reptrak.com/rankings/

[110] European Data Protection Board. (n.d.). GDPR enforcement tracker. Retrieved December 3, 2024, from https://www.enforcementtracker.com/

[111] Kshetri, N. (2021). Cybercrime and cybersecurity in the global south. Palgrave Macmillan. https://doi.org/10.1007/978-3-030-69264-1

[112] Palo Alto Networks. (2023). 2023 Unit 42 incident response report. https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/pdf/reports/Unit_42/unit42-incident-response-report-2023.pdf

[113] Gartner. (2023). Top strategic technology trends for 2024. Gartner, Inc.

[114] Joint Task Force. (2020). Security and privacy controls for information systems and organizations (NIST Special Publication 800-53 Rev. 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53r5

[115] FIRST. (2019). Common Vulnerability Scoring System version 3.1: Specification document. Forum of Incident Response and Security Teams. https://www.first.org/cvss/v3.1/specification-document

[116] OWASP Foundation. (2023). OWASP ModSecurity Core Rule Set. https://owasp.org/www-project-modsecurity-core-rule-set/

[117] Google. (2023). gVisor: Application kernel for containers. https://gvisor.dev/docs/

[118] Center for Internet Security. (2023). CIS Controls version 8. https://www.cisecurity.org/controls/v8

[119] IBM Security. (2023). QRadar SIEM: AI-powered threat detection. https://www.ibm.com/products/qradar-siem

[120] SAFECode. (2018). Fundamental practices for secure software development (3rd ed.). Software Assurance Forum for Excellence in Code. https://safecode.org/wp-content/uploads/2018/03/SAFECode_Fundamental_Practices_for_Secure_Software_Development_March_2018.pdf

[121] Verizon. (2023). 2023 Payment Security Report. https://www.verizon.com/business/resources/reports/payment-security-report/

[122] European Union. (2022). Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union (NIS2 Directive). Official Journal of the European Union, L 333. https://eur-lex.europa.eu/eli/dir/2022/2555/oj

[123] ENISA. (2023). NIS 2 Directive: Getting started. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/topics/nis-directive

[124] European Data Protection Board. (2022). Guidelines 07/2020 on the concepts of controller and processor in the GDPR. https://edpb.europa.eu/our-work-tools/our-documents/guidelines/guidelines-072020-concepts-controller-and-processor-gdpr_en

[125] Rodrigues, R. (2020). Legal and human rights issues of AI: Gaps, challenges and vulnerabilities. Journal of Responsible Technology, 4, 100005. https://doi.org/10.1016/j.jrt.2020.100005

[126] (ISC)². (2023). Code of ethics. International Information System Security Certification Consortium. https://www.isc2.org/Ethics

[127] Kennedy, D., O'Gorman, J., Kearns, D., & Aharoni, M. (2011). Metasploit: The penetration tester's guide. No Starch Press.

[128] ISO/IEC. (2018). Information technology — Security techniques — Vulnerability disclosure (ISO/IEC 29147:2018). International Organization for Standardization.

[129] Von Solms, R., & Van Niekerk, J. (2013). From information security to cyber security. Computers & Security, 38, 97-102. https://doi.org/10.1016/j.cose.2013.04.004

[130] Gartner. (2023). Top cybersecurity trends for 2024. Gartner, Inc.

[131] MITRE Corporation. (2023). Common Weaknesses in Infrastructure as Code. https://cwe.mitre.org/

[132] Brundage, M., Avin, S., Wang, J., Belfield, H., Krueger, G., Hadfield, G., Khlaaf, H., Yang, J., Toner, H., Fong, R., Maharaj, T., Koh, P. W., Hooker, S., Leung, J., Trask, A., Bluemke, E., Lebensold, J., O'Keefe, C., Koren, M., ... Anderljung, M. (2020). Toward trustworthy AI development: Mechanisms for supporting verifiable claims. arXiv. https://arxiv.org/abs/2004.07213

[133] National Institute of Standards and Technology. (2023). Post-quantum cryptography: Selected algorithms 2023. https://csrc.nist.gov/projects/post-quantum-cryptography

[134] ETSI. (2023). Multi-access Edge Computing (MEC): Framework and reference architecture (ETSI GS MEC 003 V3.1.1). European Telecommunications Standards Institute.

[135] Allodi, L., & Massacci, F. (2017). Security events and vulnerability data for cybersecurity risk estimation. Risk Analysis, 37(8), 1606-1627. https://doi.org/10.1111/risa.12864

[136] Böhme, M., Pham, V.-T., & Roychoudhury, A. (2019). Coverage-based greybox fuzzing as Markov chain. IEEE Transactions on Software Engineering, 45(5), 489-506. https://doi.org/10.1109/TSE.2017.2785841

[137] Allodi, L., Cremonini, M., Massacci, F., & Shim, W. (2020). The effectiveness of patching: A question of economics. IEEE Security & Privacy, 18(5), 16-24. https://doi.org/10.1109/MSEC.2020.2986376

[138] DevOps Institute. (2023). Upskilling 2023: Enterprise DevOps skills report. https://devopsinstitute.com/upskilling-2023/

[139] McGraw, G. (2006). Software security: Building security in. Addison-Wesley Professional.

[140] SANS Institute. (2023). SANS cyber threat intelligence survey 2023. https://www.sans.org/white-papers/

[141] ENISA. (2023). ENISA threat landscape 2023. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[142] Bouwman, X., Gañán, C., & Van Eeten, M. (2020). Measuring the impact of coordinated vulnerability disclosure. In 29th USENIX Security Symposium (pp. 1829-1846). USENIX Association.

[143] ENISA. (2022). Good practices for vulnerability disclosure: From challenges to recommendations. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/vulnerability-disclosure

[144] Herley, C., & van Oorschot, P. C. (2017). SoK: Science, security and the elusive goal of security as a scientific pursuit. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 99-120). IEEE. https://doi.org/10.1109/SP.2017.38

[145] Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., & McQuaid, R. (2021). Developing cyber-resilient systems: A systems security engineering approach (NIST Special Publication 800-160 Vol. 2 Rev. 1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-160v2r1

[146] Penetration Testing Execution Standard. (2014). The penetration testing execution standard. http://www.pentest-standard.org/

[147] MITRE Corporation. (n.d.). ATT&CK for enterprise. Retrieved December 3, 2024, from https://attack.mitre.org/

[148] Herley, C., & van Oorschot, P. C. (2017). SoK: Science, security and the elusive goal of security as a scientific pursuit. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 99-120). IEEE. https://doi.org/10.1109/SP.2017.38

[149] OWASP Foundation. (2023). OWASP reconnaissance guide. https://owasp.org/www-community/attacks/

[150] Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). Technical guide to information security testing and assessment (NIST Special Publication 800-115). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-115

[151] Yamin, M. M., Katt, B., & Gkioulos, V. (2020). Cyber ranges and security testbeds: Scenarios, functions, tools and architectures. Computers & Security, 88, 101636. https://doi.org/10.1016/j.cose.2019.101636

[152] SANS Institute. (2023). Penetration testing: A survival guide. https://www.sans.org/white-papers/

[153] Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research, 1(1), 80-106.

[154] ISO/IEC. (2018). Information technology — Security techniques — Vulnerability disclosure (ISO/IEC 29147:2018). International Organization for Standardization.

[155] Spring, J. M., Hatleback, E., Householder, A., Manion, A., & Shick, D. (2021). Time to change the CVSS? IEEE Security & Privacy, 19(2), 74-78. https://doi.org/10.1109/MSEC.2020.3044475

[156] Penetration Testing Execution Standard. (2014). Pre-engagement. http://www.pentest-standard.org/index.php/Pre-engagement

[157] Yamin, M. M., Katt, B., & Gkioulos, V. (2020). Cyber ranges and security testbeds: Scenarios, functions, tools and architectures. Computers & Security, 88, 101636. https://doi.org/10.1016/j.cose.2019.101636

[158] SANS Institute. (2023). Penetration testing tools guide. https://www.sans.org/tools/

[159] MITRE Corporation. (2023). CALDERA: Automated adversary emulation system. https://caldera.mitre.org/

[160] ISO/IEC. (2012). Information technology — Security techniques — Guidelines for identification, collection, acquisition and preservation of digital evidence (ISO/IEC 27037:2012). International Organization for Standardization.

[161] Joint Task Force. (2012). Guide for conducting risk assessments (NIST Special Publication 800-30 Rev. 1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-30r1

[162] OWASP Foundation. (2023). OWASP risk rating methodology. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

[163] OWASP Foundation. (2021). OWASP Web Security Testing Guide (WSTG) version 4.2. https://owasp.org/www-project-web-security-testing-guide/

[164] Penetration Testing Execution Standard. (2014). The penetration testing execution standard. http://www.pentest-standard.org/

[165] ENISA. (2023). ENISA threat landscape 2023. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[166] Bier, S., Fajardo, B., Ezeadum, O., Guzman, G., Sultana, K. Z., & Anu, V. (2021). Mitigating remote code execution vulnerabilities: A study on Tomcat and Android security updates. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). IEEE. https://doi.org/10.1109/IEMTRONICS52119.2021.9422666

[167] Stouffer, K., Lightman, S., Pillitteri, V., Abrams, M., & Hahn, A. (2023). Guide to operational technology (OT) security (NIST Special Publication 800-82 Rev. 3). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-82r3

[168] OWASP Foundation. (2023). OWASP Secure Coding Practices Quick Reference Guide. https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/

[169] MITRE Corporation. (n.d.). Common Vulnerabilities and Exposures (CVE). Retrieved December 3, 2024, from https://cve.mitre.org/

[170] Verizon. (2023). 2023 Data Breach Investigations Report. https://www.verizon.com/business/resources/reports/dbir/

[171] ISO/IEC. (2011). Information technology — Security techniques — Application security (ISO/IEC 27034-1:2011). International Organization for Standardization.

[172] Verizon. (2023). 2023 Data Breach Investigations Report. https://www.verizon.com/business/resources/reports/dbir/

[173] U.S. Government Accountability Office. (2018). Data protection: Actions taken by Equifax and federal agencies in response to the 2017 breach (Report No. GAO-18-559). https://www.gao.gov/products/gao-18-559

[174] Federal Trade Commission. (2019). Equifax data breach settlement. https://www.ftc.gov/enforcement/cases-proceedings/refunds/equifax-data-breach-settlement

[175] Apache Software Foundation. (2021). Apache Log4j security vulnerabilities. https://logging.apache.org/log4j/2.x/security.html

[176] CISA. (2021). Emergency directive 22-02: Mitigate Apache Log4Shell vulnerability. Cybersecurity and Infrastructure Security Agency. https://www.cisa.gov/news-events/directives/ed-22-02-mitigate-apache-log4shell-vulnerability

[177] Mandiant. (2023). MOVEit transfer critical vulnerability rapidly exploited. https://www.mandiant.com/resources/blog/zero-day-moveit-data-theft

[178] IBM Security X-Force. (2023). X-Force threat intelligence index 2023. https://www.ibm.com/reports/threat-intelligence

[179] ENISA. (2023). ENISA threat landscape 2023. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[180] Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2022). Security strategies for microservices-based application systems (NIST Special Publication 800-204C). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-204C

[181] Europol. (2023). Internet Organised Crime Threat Assessment (IOCTA) 2023. https://www.europol.europa.eu/publication-events/main-reports/internet-organised-crime-threat-assessment-iocta-2023

[182] Mandiant. (2023). M-Trends 2023: A view from the front lines. https://www.mandiant.com/resources/reports/m-trends-2023

[183] MITRE Corporation. (n.d.). Cyber Analytics Repository (CAR). Retrieved December 3, 2024, from https://car.mitre.org/

[184] CrowdStrike. (2023). 2023 Global Threat Report. https://www.crowdstrike.com/global-threat-report/

[185] Palo Alto Networks. (2023). 2023 Unit 42 incident response report. https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/pdf/reports/Unit_42/unit42-incident-response-report-2023.pdf

[186] Flashpoint. (2023). 2023 Global threat intelligence report. https://flashpoint.io/resources/report/global-threat-intelligence-report/

[187] Kaspersky. (2023). Kaspersky Security Bulletin 2023: Statistics. https://securelist.com/ksb-2023-statistics/111010/

[188] ENISA. (2023). ENISA threat landscape 2023. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[189] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

[190] ISACA. (2023). State of cybersecurity 2023. https://www.isaca.org/resources/reports/state-of-cybersecurity-2023