



Universidade do Minho
Escola de Engenharia

Mestrado em Engenharia Informática

Ano letivo 2024/2025

Tecnologias de Segurança

Trabalho Prático #2

Grupo 02

João Rodrigues - pg57880

Rúben Silva - pg57900

Miguel Guimarães - pg55986

Março 2025

Índice

| | | |
|---------|---|----|
| 1 | Introdução | 1 |
| 2 | Parte A | 1 |
| 2.1 | Análise do negocio local ‘Frutas do Cavado’ | 1 |
| 2.1.1 | Redes sociais e website | 1 |
| 2.1.2 | Análise do website | 3 |
| 2.1.3 | Análise de DNS | 4 |
| 2.1.4 | Email Exposto | 5 |
| 2.1.5 | Continuação da análise de DNS | 6 |
| 2.1.6 | Continuação da analise do website | 6 |
| 2.2 | Análise da Grande Corporação ‘YOTEL’ | 7 |
| 2.2.1 | Analise do Website Internacional | 7 |
| 2.2.2 | Análise ao Website do restaurante YOTEL Porto (Komyuniti) | 11 |
| 2.2.2.1 | Ataque de Engenharia Social através de numero telefônico | 11 |
| 2.2.3 | Continuação da análise do website | 12 |
| 2.2.3.1 | Possível spoofing de Email | 12 |
| 3 | Conclusões sobre a Parte A | 14 |
| 4 | Parte B | 15 |
| 4.1 | Cenário de Teste | 15 |
| 4.2 | Questões | 16 |
| 4.2.1 | B1 | 16 |
| 4.2.2 | B2 | 17 |
| 4.2.2.1 | B2.1 | 17 |
| 4.2.2.2 | B2.2 | 19 |
| 4.2.2.3 | B2.3 | 20 |
| 4.2.3 | B3 | 22 |
| 4.2.4 | B4 | 22 |
| 4.2.5 | B5 | 23 |
| 4.2.6 | B6 | 24 |
| 4.2.7 | B7 | 25 |
| 4.2.7.1 | B7.1 | 26 |
| 4.2.7.2 | B7.2 | 26 |
| 5 | Referências | 27 |

Figuras

| | |
|--|----|
| Figura 1 | 1 |
| Figura 2 Criador do website da empresa Frutas do Cávado | 2 |
| Figura 3 Empresa criadora do website da empresa Frutas do Cávado | 2 |
| Figura 4 Competências tecnologias de um estagiário na Uebyou | 3 |
| Figura 5 Fontes do website das Frutas do Cavado | 3 |
| Figura 6 Plugins do Wordpress utilizados pelo website das Frutas do Cávado | 4 |
| Figura 7 whois lookup do website frutasdocavado.com | 4 |
| Figura 8 nslookup do website frutasdocavado.com | 4 |
| Figura 9 host do website frutasdocavado.com | 5 |
| Figura 10 host do website frutasdocavado.com | 5 |
| Figura 11 host do website frutasdocavado.com | 5 |
| Figura 12 host do website frutasdocavado.com | 5 |
| Figura 13 curl do website frutasdocavado.com | 6 |
| Figura 14 Acesso à Área Reservada do website da empresa ‘Frutas do Cavado’ | 6 |
| Figura 15 URL da página de login do website da empresa ‘Frutas do Cavado’ | 6 |
| Figura 16 Website da empresa YOTEL | 8 |
| Figura 17 whois do website yotel.com P1 | 8 |
| Figura 18 whois do website yotel.com P2 | 8 |
| Figura 19 nslookup do website yotel.com | 9 |
| Figura 20 comando host do website yotel.com | 9 |
| Figura 21 Verificação do email contact@yotel.com | 9 |
| Figura 22 Verificação do email jobs@yotel.com | 9 |
| Figura 23 Verificação do email marcia.ang@yotel.com | 9 |
| Figura 24 Verificação do email carolina-perdomo@yotel.com | 9 |
| Figura 25 Informações sobre o email contact@yotel.com | 10 |
| Figura 26 curl do website da Yotel | 11 |
| Figura 27 Website do restaurante do YOTEL Porto | 11 |
| Figura 28 Host do website do restaurante | 12 |
| Figura 29 Nslookup do website do restaurante | 12 |
| Figura 30 Host do website do restaurante | 12 |
| Figura 31 Tentativa de lookup SPF sem resposta | 13 |
| Figura 32 Tentativa de lookup SPF sem resposta 2 | 13 |
| Figura 33 Verificação de DMARC | 13 |
| Figura 34 VM com Kali utilizada como Auditor (exemplo na VBox) | 15 |
| Figura 35 VM com Metasploitable2 utilizada como Alvo (exemplo na VBox) | 15 |
| Figura 36 Rede à qual as VM’s se encontram ligadas (exemplo na VBox) | 15 |
| Figura 37 /etc/network/interfaces no kali | 15 |
| Figura 38 /etc/network/interfaces no metasploitable2 | 15 |
| Figura 39 pacotes_telnet | 16 |
| Figura 40 login_plaintext | 16 |
| Figura 41 passwd_plaintext | 17 |
| Figura 42 ipconfig_plaintext | 17 |
| Figura 43 B2: nmap da varredura 1 | 18 |
| Figura 44 B2: nmap da varredura 2 | 18 |
| Figura 45 B2: parte do nmap da varredura 3 (porta 3306) | 18 |
| Figura 46 B2: parte do nmap da varredura 4 (porta 3306) | 19 |
| Figura 47 login_filtered | 23 |

| | |
|--|----|
| Figura 48 X11_filtered | 23 |
| Figura 49 B5 varredura 1 | 23 |
| Figura 50 B5 varreduras 2,3,4 | 24 |
| Figura 51 B5 varredura 5 | 24 |
| Figura 52 B6 relatório do nikto | 25 |
| Figura 53 Vulnerabilidades Metasploitable 2 - Nessus | 25 |
| Figura 54 Resultado Final do Adv. Scan - Nessus | 26 |

1 Introdução

A análise proativa de infraestruturas digitais é fundamental para identificar e mitigar riscos associados a vulnerabilidades tecnológicas. Este trabalho prático tem como objetivo aplicar técnicas de footprinting e varredura de vulnerabilidades para avaliar a postura de segurança de sistemas online. O projeto está estruturado em duas fases complementares: (i) reconhecimento passivo, utilizando técnicas de Open-Source Intelligence (OSINT) para mapear a superfície de ataque de duas entidades comerciais, e (ii) varredura ativa e enumeração de vulnerabilidades em um ambiente controlado (Metasploitable 2), recorrendo a ferramentas como Nmap, Nikto e Nessus.

Na primeira fase, foram selecionadas duas organizações com presença digital: a Frutas do Cávado, um negócio local sediado em Braga, Portugal, e a YOTEL, uma cadeia hoteleira internacional com operações tecnológicas avançadas. A análise passiva abrangeu a recolha de informações públicas, como registos DNS, configurações de email e tecnologias de frontend e backend, permitindo comparar as práticas de gestão de segurança entre uma entidade de pequena escala e uma corporação global.

A segunda fase envolveu a configuração de um ambiente de teste com máquinas virtuais Kali Linux e Metasploitable 2, onde foram realizadas varreduras para identificar fragilidades exploráveis, com ênfase na detecção de vulnerabilidades associadas a serviços de rede.

Este relatório apresenta uma análise crítica dos riscos identificados, destacando as diferenças nas estratégias de segurança das organizações analisadas e propondo medidas de mitigação baseadas nos resultados das varreduras. As seções seguintes detalham a metodologia, os resultados obtidos e as conclusões derivadas, contribuindo para o avanço do conhecimento em cibersegurança aplicado.

2 Parte A

2.1 Análise do negócio local ‘Frutas do Cávado’

A empresa de negócio local que foi escolhida é a empresa “**Frutas do Cávado**”. É uma empresa especializada em comercialização de fruta, que possui a sua sede em Braga, Barcelos, recorrendo à internet para divulgar os seus serviços.

A primeira etapa desta fase é a fase de reconhecimento. Inicialmente vamos identificar a presença da empresa na internet, criando assim uma coletânea de fontes de informação.

2.1.1 Redes sociais e website

Foi identificada a presença da empresa nas seguintes plataformas:

- **Facebook** sobre o username ‘FrutasDoCavadoLDA’.
- **LinkedIn** sobre o username ‘frutasdocavado’.
- **Instagram** sobre o username ‘frutas.do.cavado’.
- **X** sobre o username ‘frutasdocavado’

Foi encontrado um website com domínio próprio que pertence à empresa “www.frutasdocavado.com”.



Figura 2: Criador do website da empresa Frutas do Cávado

Nota: As redes sociais e o website encontrado podem conter informações importantes sobre as tecnologias utilizadas na empresa, através das vagas de emprego oferecidas por exemplo. No entanto não foram encontradas informações sobre as tecnologias utilizadas pela empresa em quaisquer das redes sociais mencionadas.

Após uma analise do website do negocio local foi possível identificar **a empresa que foi responsável pela sua criação**.

Frutas do Cávado @ 2023 | Desenvolvido por Uebyou . Creative Agency

Figura 3: Empresa criadora do website da empresa Frutas do Cávado

Nota: Gostaríamos de destacar que a empresa Uebyou costuma inserir o seu nome no rodapé dos websites que desenvolve, permitindo descobrir através de uma simples pesquisa todos, ou a maioria, dos websites desenvolvidos pela mesma. Isso significa que ao encontrar uma vulnerabilidade em um dos websites desenvolvidos pela Uebyou sites os restantes poderão encontrar-se expostos aos mesmos riscos. Num cenário real uma analise aos restantes websites desenvolvidos pela Uebyou poderia ser benéfica no processo de ataque, permitindo descobrir informações extra. Consideramos que a inserção do nome no rodapé do website facilita o Reconnaissance a um atacante, não sendo em si uma falha de segurança propriamente dita.

Após esta descoberta será necessário realizar uma análise da empresa **‘Uebyou’** que se apresenta como **a empresa responsável pela criação**, e provável manutenção, do website do negocio local.

Após alguma pesquisa foi possível identificar que a empresa **Uebyou** é uma empresa localizada em Braga que atua na área das tecnologias de informação implementando soluções tecnológicas de web e Cloud Computing.

A empresa encontra-se nas seguintes plataformas:

- **Facebook** sobre o username ‘uebyou.criamosvalor’.
- **Instagram** sobre o username ‘uebyou.agency’.
- **LinkedIn** sobre o username ‘uebyou-ti-criamos-valor’.
- **Behance** sobre o username ‘uebyoucriamosvalor’.

- **Youtube** sobre o username ‘UebyouCriamosValor’.

A empresa possui também um **website** “uebyou.pt”.

Através da analise das redes sociais foi possível identificar alguns dos funcionários que trabalham na empresa, e até os seus perfis pessoais, o que possibilita um ataque através de engenharia social.

Em 2020 a empresa encontrava-se à procura de um funcionário na plataforma **EmpregoXL** (ver referencias), e as competências tecnológicas esperadas pela empresa eram **Wordpress, PHP, javascript, html, css e a Suite Adobe**.

Domínio das ferramentas PHP, Javascript, HTML, CSS, WordPress, Suite Adobe

Figura 4: Competências tecnologias de um estagiário na Uebyou

Estas informações são extremamente importantes, pois indicam, de forma parcial ou até total, quais as **tecnologias** utilizadas pela empresa.

A utilização destas tecnologias será confirmada posteriormente, uma vez que as tecnologias utilizadas pela empresa poderão ser diferentes na atualidade.

2.1.2 Análise do website

A utilização de **Wordpress** poderá indicar a plataforma utilizada pela empresa na construção do website das Frutas do Cavado. No entanto não possuímos a confirmação deste facto de forma concreta.

A utilização de **PHP** indica que a empresa poderá ser responsável pela criação e manutenção das base de dados dos websites que cria, facto confirmado pela descrição da empresa no seu website onde é descrito que a empresa é em parte uma empresa de Cloud Computing.

Os websites desenvolvidos pela Uebyou parecem, então, integrar duas componentes principais: o WordPress para a lógica de frontend e uma base de dados de suporte.

Para confirmar a utilização do Wordpress na criação do website foram consultadas as fontes e os arquivos carregados pelo website do negocio local.

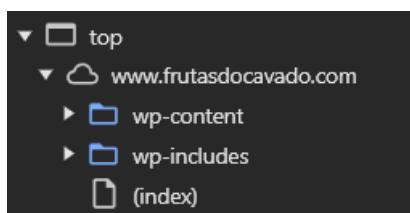


Figura 5: Fontes do website das Frutas do Cavado

É possível verificar que o website das frutas do cavado segue uma estrutura tipica utilizada por websites criados em Wordpress.

A pasta **wp-content**, por exemplo, é a pasta utilizada pelo Wordpress para armazenar o conteúdo criado pelo utilizador, já a pasta **wp-includes** contem os arquivos principais do Wordpress.

Possuímos assim uma confirmação factual de que o website do negocio local foi construído utilizando Wordpress tal como suspeitado a partir da vaga de emprego.

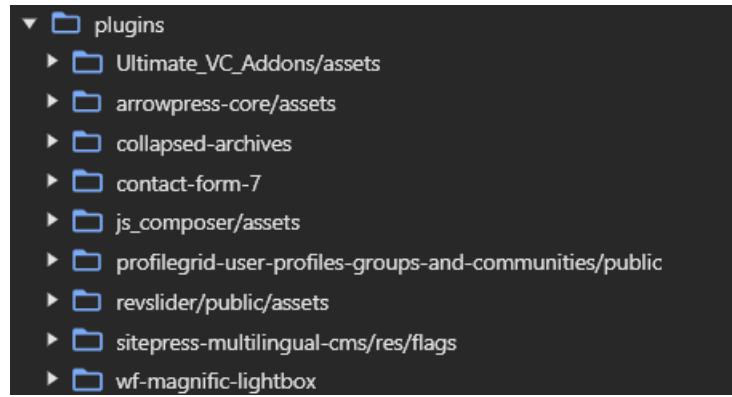


Figura 6: Plugins do Wordpress utilizados pelo website das Frutas do Cávado

Na pasta **wp-content** foi possível identificar os plugins do Wordpress utilizados na construção do website, o que poderá revelar vulnerabilidades na implementação do website.

2.1.3 Análise de DNS

```
Domain Name: FRUTASDOCAVADO.COM
Registry Domain ID: 1391715365_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.publicdomainregistry.com
Registrar URL: www.publicdomainregistry.com
Updated Date: 2024-12-10T10:38:43Z
Creation Date: 2008-01-31T17:36:55Z
Registrar Registration Expiration Date: 2026-01-31T17:36:55Z
Registrar: PDR Ltd. d/b/a PublicDomainRegistry.com
Registrar IANA ID: 303
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registry Registrant ID: GDPR Masked
Registrant Name: GDPR Masked
Registrant Organization: GDPR Masked
Registrant Street: GDPR Masked
Registrant City: GDPR Masked
Registrant State/Province: Braga
Registrant Postal Code: GDPR Masked
Registrant Country: PT
Registrant Phone: GDPR Masked
Registrant Phone Ext:
Registrant Fax: GDPR Masked
Registrant Fax Ext:
Registrant Email: gdpr-masking@gdpr-masked.com
Registry Admin ID: GDPR Masked
Admin Name: GDPR Masked
Admin Organization: GDPR Masked
Admin Street: GDPR Masked
Admin City: GDPR Masked
Admin State/Province: GDPR Masked
Admin Postal Code: GDPR Masked
Admin Country: GDPR Masked
Admin Phone: GDPR Masked
Admin Phone Ext:
Admin Fax: GDPR Masked
Admin Fax Ext:
Admin Email: gdpr-masking@gdpr-masked.com
Registry Tech ID: GDPR Masked
Tech Name: GDPR Masked
Tech Organization: GDPR Masked
Tech Street: GDPR Masked
```

Figura 7: whois lookup do website frutasdocavado.com

Foi realizada uma pesquisa sobre as informações do domínio, infelizmente as informações encontram-se **protegidas** pela lei **GDPR** que é uma lei na UE que protege as pessoas de terem seus dados pessoais expostos sem consentimento.

Ainda assim foi possível verificar que o website foi registado por alguém localizado em **Braga, Portugal**.

```
miguel_linux@DESKTOP-1790S2J:~$ nslookup frutasdocavado.com
;; Got recursion not available from 172.24.0.1
Server:      172.24.0.1
Address:     172.24.0.1#53

Non-authoritative answer:
Name:  frutasdocavado.com
Address: 94.46.168.92
```

Figura 8: nslookup do website frutasdocavado.com

Ao realizar nslookup foi possível descobrir que o nome de domínio se traduz no IP “94.46.168.92”.

```
miguel_linux@DESKTOP-1790S2J:~$ host 94.46.168.92
92.168.46.94.in-addr.arpa domain name pointer uebyou.net.
```

Figura 9: host do website frutasdocavado.com

É possível verificar que o website encontra-se hospedado no mesmo IP que o website da uebyou, que é, como vimos anteriormente, a empresa responsável pela criação e manutenção do website do negocio local.

```
miguel_linux@DESKTOP-1790S2J:~$ host frutasdocavado.com
frutasdocavado.com has address 94.46.168.92
frutasdocavado.com mail is handled by 10 mx1.cleanmx.pt.
frutasdocavado.com mail is handled by 20 mx2.cleanmx.pt.
```

Figura 10: host do website frutasdocavado.com

Ao utilizar o comando host foi possível verificar que o negocio local possui um mailing system, este é gerido pela empresa **Clean MX**, ou seja, o mailing system do negocio local não é gerido pelo próprio negocio local recorrendo a uma empresa especializada nesse serviço, fornecendo mais segurança a esta componente. O facto de existir um mailing system indica que podem existir endereços e-mail do tipo **nome@frutasdocavado.com**, o que poderá ser útil para descobrir mais informações.

Após alguma pesquisa utilizando algumas ferramentas de lookup de e-mail foi possível descobrir alguns dos endereços utilizados pela empresa:

- geral@frutasdocavado.com
- famalicao@frutasdocavado.com
- braga@frutasdocavado.com
- porto@frutasdocavado.com
- mercominho@frutasdocavado.com
- coimbra@frutasdocavado.com

2.1.4 Email Exposto

Utilizando uma das ferramentas sugeridas no OSINT Framework, o ‘haveibeenpwned.com’, foi possível verificar que o email do negocio local foi exposto publicamente

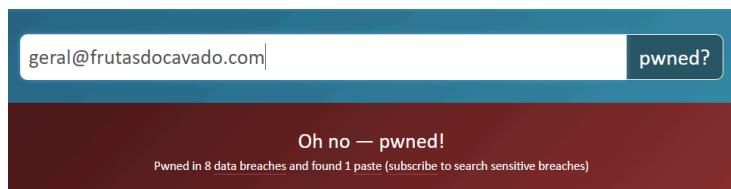


Figura 11: host do website frutasdocavado.com

Isto representa uma **falha crítica**, indicando que as credenciais utilizadas foram expostas publicamente.

| Paste title | Date | Emails |
|-------------|-------------------|--------|
| No title | 2 Jul 2016, 17:09 | 9,780 |

Figura 12: host do website frutasdocavado.com

É possível verificar que os dados foram publicados em 2016 num paste sem nome. Após uma verificação no wayback machine foi possível verificar que a empresa ‘Frutas do Cávado’ já possui website desde 2012, e a área restrita já existe desde 2014, sendo este paste possível.

Estranhamente os restantes **e-mails** não foram expostos publicamente, no entanto ao considerar que a empresa exporta para vários países e que o seu e-mail principal é o geral surge ai uma maior exposição deste email em relação aos restantes que apenas surgem em regiões mais restritas.

2.1.5 Continuação da análise de DNS

Através de nslookup foi possível verificar que o domínio utiliza ‘spf1’ que define quais os servidores/IPs que podem enviar um email usando este domínio, acrescentando assim uma camada de defesa contra spoofing. A análise DNS mostrou ainda que o domínio dispõe de: **DKIM** (*DomainKeys Identified Mail*) e **DMARC** (*Domain-based Message Authentication Reporting & Conformance*) que garantem integridade e validam a autenticidade do remetente.

Ao existirem dois mail servers isso significa que existe um mail server principal e um de backup, garantindo maior disponibilidade do serviço.

```
miguel_linux@DESKTOP-1790S2J: ~ curl -I http://frutasdocavado.com
HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Wed, 02 Apr 2025 14:13:57 GMT
Content-Type: text/html; charset=iso-8859-1
Connection: keep-alive
Location: https://frutasdocavado.com/
X-Scale: YXBvY2FzQGdpdGh1Yg==
```

Figura 13: curl do website frutasdocavado.com

Ao realizar curl do website é possível verificar que a tecnologia utilizada para hospedar o website é o **nginx**, ou, **engine-x**, que é um dos web servers mais utilizados. Caso alguma vulnerabilidade surja no nginx o website do negócio local encontra-se automaticamente vulnerável.

2.1.6 Continuação da análise do website

O website do negócio local contém uma Área Reservada.

[Contactos](#) [Área Reservada](#) [Português](#)

Figura 14: Acesso à Área Reservada do website da empresa ‘Frutas do Cavado’

Ao analisar o URL página de login da área reservada foi possível verificar que a empresa Uebyou utilizou **PHP** na construção da página, tal como suspeitado anteriormente.

  www.frutasdocavado.com/wp-login.php

Figura 15: URL da página de login do website da empresa ‘Frutas do Cavado’

Foi realizada uma tentativa para **listar os utilizadores do website** (técnica OSINT), no entanto o website não forneceu esses dados, exibindo a mensagem “Não tem permissão para listar os utilizadores.”. URL utilizado: “<https://www.frutasdocavado.com/wp-json/wp/v2/users>”

Também verificamos que o website não permite acesso através de **VPN**.

Assumimos que estas medidas são implementadas por defeito pela equipa do Wordpress.

Infelizmente a **página de registo** não se encontra funcional, não enviando emails de confirmação, como tal não foi possível realizar uma pesquisa mais aprofundada desta componente.

2.2 Análise da Grande Corporação ‘YOTEL’

A grande corporação escolhida para esta fase do trabalho é a ‘YOTEL’, que se apresenta como uma cadeia de hotéis internacional, possuindo no distrito do Porto um dos seus hotéis. A cadeia de hotéis ‘YOTEL’ é uma das cadeias de hotéis mais futuristas do mundo, possuindo um grande foco em tecnologia, possuindo quiosques de check-in/out, robôs e aplicações mobile para controlar o quarto.

Esta empresa possui assim algumas superfícies de ataque extra comparado com o negocio local, como a parte da robótica.

Tal como com o negocio local iremos começar por analisar a presença da corporação na internet, iremos realizar este processo para tanto o hotel do Porto como os perfis internacionais, procurando obter informações como vagas de emprego ou os perfis da equipa de segurança por exemplo.

Encontramos a presença do hotel do porto nas seguintes redes sociais:

- **Instagram do hotel**, sobre o username ‘yotelpporto’.
- **Instagram do restaurante**, sobre o username ‘komyunitipporto’.
- **Facebook**, sobre o username ‘YOTELporto’.
- **Youtube**, sobre o username ‘yotelpporto550’.

Já a empresa internacional encontramos as seguintes redes sociais:

- **Instagram**, sobre o username ‘yotel’.
- **Facebook**, sobre o username ‘yotel’.
- **X**, sobre o username ‘yotelhq’.
- **Youtube**, sobre o username ‘yotel’.
- **Vimeo**, sobre o username ‘yotel’.
- **Tiktok**, sobre o username ‘yotelhq’.
- **Linkedin**, sobre o username ‘yotel’.

Foi também encontrada a presença do hotel em plataformas de hotelaria, no entanto essas não fornecem detalhes relevantes no contexto deste projeto.

O hotel do Porto não possui website próprio, existindo apenas o **website internacional** em Português. www.yotel.com/pt-pt/.

No entanto encontramos o **website do restaurante** do Hotel do Porto, sobre o URL “www.komyunitipporto.com”

Infelizmente não foram encontradas quaisquer vagas de informática ou até mesmo robótica na empresa. No entanto foi encontrado o perfil do ‘IT Analyst’ do hotel do Porto, e através de uma pequena analise dos seguidores do hotel do Porto no Instagram foi possível encontrar o seu perfil pessoal.

O ideal seria o cargo encontrar-se oculto do Linkedin, protegendo o analista contra ataques de **spear phishing** e de **engenharia social**.

Já no Facebook todos os seguidores encontram-se ocultos o que revela uma **boa prática de segurança**.

2.2.1 Analise do Website Internacional

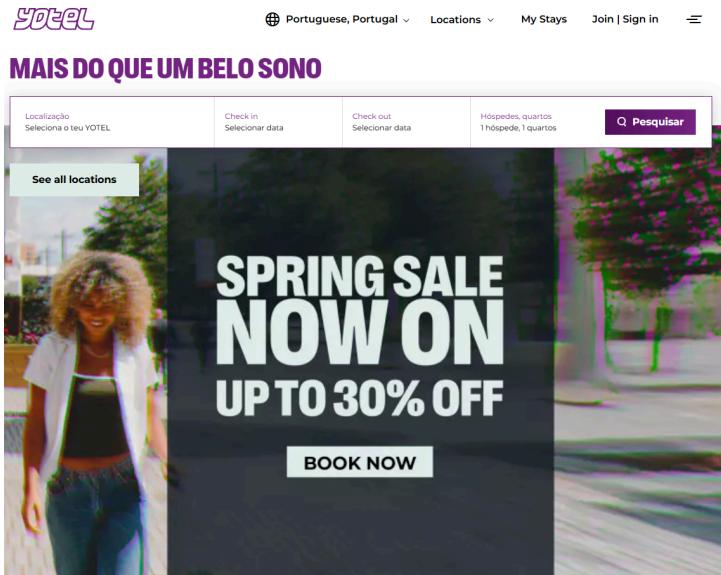


Figura 16: Website da empresa YOTEL

O código fonte do website não revelou nenhuma informação sobre o seu criador.

Foi possível descobrir que o website foi construído utilizando **html, css, javascript e jquery**.

```
miguel_linux@DESKTOP-1790S2J: $ whois yotel.com
Domain Name: YOTEL.COM
Registry Domain ID: 84811238_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: http://www.godaddy.com
Updated Date: 2022-09-13T20:56:43Z
Creation Date: 2002-03-24T19:37:41Z
Registry Expiry Date: 2027-03-24T18:37:40Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: 488-624-2505
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Name Server: AMBER.NS.CLOUDFLARE.COM
Name Server: EUGENE.NS.CLOUDFLARE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2025-04-20T01:04:43Z <<<
```

Figura 17: whois do website yotel.com P1

```
Registry Registrant ID: Not Available From Registry
Registrant Name: Registration Private
Registrant Organization: Domains By Proxy, LLC
Registrant Street: DomainsByProxy.com
Registrant Street: 100 S. Mill Ave, Suite 1600
Registrant City: Tempe
Registrant State/Province: Arizona
Registrant Postal Code: 85281
Registrant Country: US
Registrant Phone: +1.4886242599
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: https://www.godaddy.com/whois/results.aspx?domain=YOTEL.COM&action=contactDomainOwner
Registry Tech ID: Not Available From Registry
Tech Name: Registration Private
Tech Organization: Domains By Proxy, LLC
Tech Street: DomainsByProxy.com
Tech Street: 100 S. Mill Ave, Suite 1600
Tech City: Tempe
Tech State/Province: Arizona
Tech Postal Code: 85281
Tech Country: US
Tech Phone: +1.4886242599
Tech Phone Ext:
Tech Fax:
Tech Fax Ext:
Tech Email: https://www.godaddy.com/whois/results.aspx?domain=YOTEL.COM&action=contactDomainOwner
```

Figura 18: whois do website yotel.com P2

Ao realizar o comando '**whois**' sobre o website da yotel.com foi possível verificar que os dados utilizados no registro do domínio encontram-se ocultos.

Os dados utilizados no registro foram ocultados pelo serviço de privacidade do '**DomainsByProxy**' da empresa de domínios '**GoDaddy**'.

```
miguel_linux@DESKTOP-1790S2J:~$ nslookup yotel.com
;; Got recursion not available from 172.24.0.1
Server:      172.24.0.1
Address:      172.24.0.1#53

Non-authoritative answer:
Name:      yotel.com
Address: 172.66.43.63
Name:      yotel.com
Address: 172.66.40.193
;; Got recursion not available from 172.24.0.1
Name:      yotel.com
Address: 2606:4700:3108::ac42:28c1
Name:      yotel.com
Address: 2606:4700:3108::ac42:2b3f
```

Figura 19: nslookup do website yotel.com

Através do comando nslookup foi possível descobrir quais os ips utilizados pelo website.

```
miguel_linux@DESKTOP-1790S2J:~$ host yotel.com
yotel.com has address 172.66.43.63
yotel.com has address 172.66.40.193
yotel.com has IPv6 address 2606:4700:3108::ac42:28c1
yotel.com has IPv6 address 2606:4700:3108::ac42:2b3f
yotel.com mail is handled by 0 yotel-com.mail.protection.outlook.com.
```

Figura 20: comando host do website yotel.com

É possível verificar que existem e-mails ‘yotel.com’. O sistema de mailing da Yotel é gerido pelo Outlook.

Utilizando algumas bases de dados de procura de emails por domínio foi possível identificar 291 endereços de email.

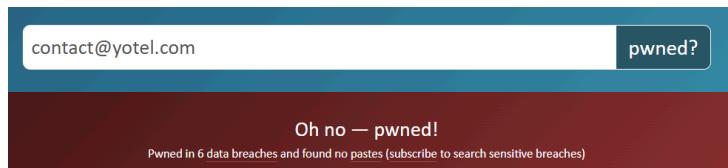


Figura 21: Verificação do email contact@yotel.com

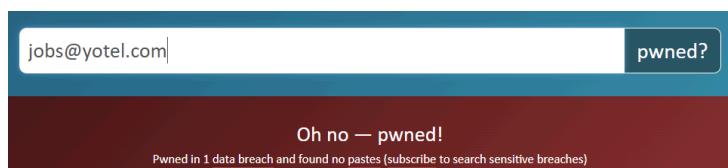


Figura 22: Verificação do email jobs@yotel.com

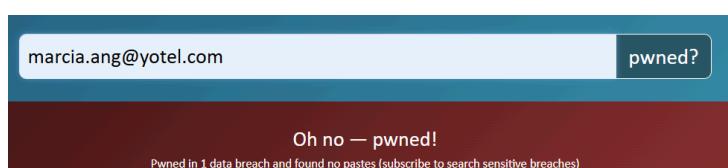


Figura 23: Verificação do email marcia.ang@yotel.com



Figura 24: Verificação do email carolina-perdomo@yotel.com

Com recurso à plataforma ‘haveibeenpwned.com’ foi possível verificar que vários endereços de email da empresa constam em **data breaches** – ou seja, credenciais ou outros dados associados a esses endereços foram extraídos diretamente de bases de dados de serviços comprometidos.

Os **data breaches** destes emails incluem:

- Dados Pessoais em um breach à empresa ‘**People Data Labs (PDL)**’.
- Breach ao serviço de anúncios ‘**DemandScience**’.
- Breach ao serviço de vendas ‘**Apollo**’.
- Breach ao serviço de armazenamento ‘**Dropbox**’.
- Breach ao serviço de verificação de emails ‘**Verifications.io**’.
- Breach à rede social ‘**Tumblr**’. (Nota: Estranhamente o email contact@yotel.com foi identificado neste breach, o que significa que este email é utilizado para vários fins, até para redes sociais. É possível que este email seja utilizado, por exemplo, como entrada na conta do instagram)

Nota: Foram realizadas verificações manuais de vários e-mails da Yotel, no entanto o website ‘haveibeenpwned.com’ fornece uma **API paga** que permite verificar todos os emails expostos de um determinado domínio.

```
{  
  "metadata": {  
    "query": "contact@yotel.com",  
    "timestamp": "2025-04-20T19:02:42.285Z"  
  },  
  "data": {  
    "visitor": {  
      "hibp": {  
        "found": "6",  
        "leaks": [  
          "tumblr.com (2013-02-28)",  
          "dropbox.com (2012-07-01)",  
          "apollo.io (2018-07-23)"  
        ]  
      },  
      "google": {  
        "id": "114714526750026792183",  
        "services": {  
          "google_maps": "https://www.google.com/maps/contrib/114714526750026792183",  
          "google_calendar": "https://calendar.google.com/calendar/u/0/embed?src=contact@yotel.com",  
          "google_plus_archive": "https://web.archive.org/web/*/plus.google.com/114714526750026792183*"  
        }  
      }  
    }  
  }  
}
```

Figura 25: Informações sobre o email contact@yotel.com

O email contact@yotel.com parece ter aparecido em vários breaches, como tal decidimos recorrer à ferramenta de reversão de email ‘**epieos.com**’. Através desta foi possível verificar a data dos ‘data breaches’ e também a presença de uma conta associada no Google maps. No entanto a conta do Google maps não possui quaisquer reviews. Esta ferramenta exibe detalhes limitados, sendo que através de um pagamento é possível obter as restantes datas entre outros detalhes.

```
miguel_linux@DESKTOP-1790S2J:~$ curl -I yotel.com
HTTP/1.1 301 Moved Permanently
Date: Sun, 20 Apr 2025 18:56:33 GMT
Content-Type: text/html
Content-Length: 167
Connection: keep-alive
Cache-Control: max-age=3600
Expires: Sun, 20 Apr 2025 19:56:33 GMT
Location: https://yotel.com/
X-Content-Type-Options: nosniff
Server: cloudflare
CF-RAY: 9336dcbe1fd26ae5-BOD
```

Figura 26: curl do website da Yotel

Como se pode verificar o servidor que hospeda o website pertence à Cloudflare, que é mais uma informação sobre a infraestrutura tecnológica do website.

2.2.2 Análise ao Website do restaurante YOTEL Porto (Komyuniti)

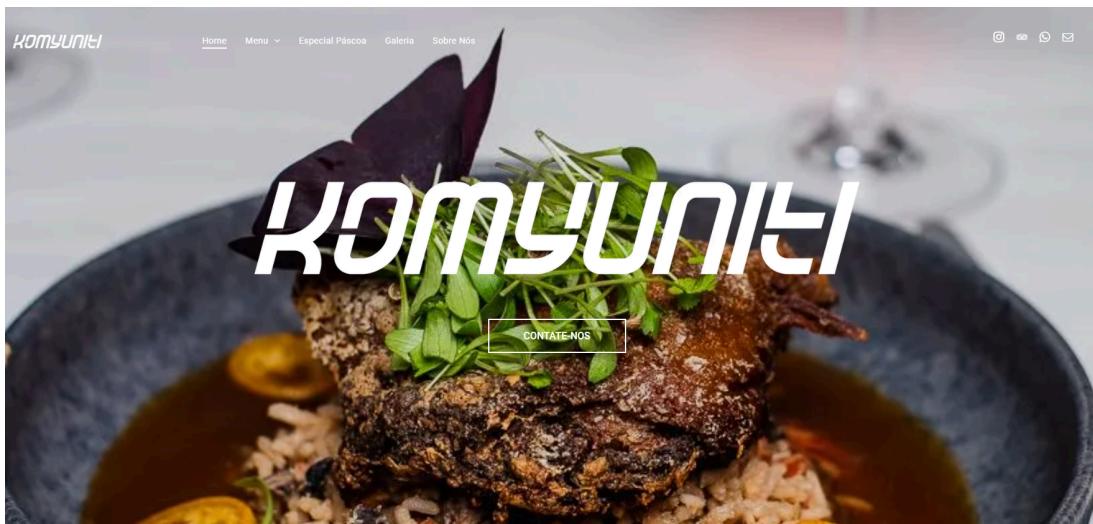


Figura 27: Website do restaurante do YOTEL Porto

Decidimos realizar uma análise ao restaurante do **YOTEL do Porto** ('komyunitiporto.com'), isto porque este encontra-se hospedado em um domínio diferente e podem ser reveladas novas informações úteis.

Após uma verificação do código fonte não foi possível identificar alguma empresa que tenha desenvolvido o website. Assumimos que a própria YOTEL possui uma **equipa responsável** pela sua criação.

Foi realizada uma análise ao código fonte do website e não foram encontradas quaisquer novas informações que sejam relevantes.

Um detalhe interessante é que o website impede acesso à página de galeria caso VPN esteja a ser utilizado, no entanto as restantes páginas permanecem acessíveis.

2.2.2.1 Ataque de Engenharia Social através de numero telefônico

O website apresenta um **numero de contacto telefônico** '(+351) 910 038 xxx' para chamadas ao restaurante. Este é um numero pessoal e pertence à rede da Vodafone.

No WhatsApp, o número exibe a fotografia de uma pessoa. A partir dessa imagem foi possível associá-la a um **perfil do Linkedin**, identificando-a como F&B Manager. Com o nome obtido, localizamos o seu **Instagram público** através da lista de seguidores da página do restaurante, recolhendo ainda mais dados pessoais.

Uma pesquisa adicional revelou o seu **Facebook**, igualmente público, onde se encontram vários perfis (presumivelmente familiares) ligados ao colaborador.

Num cenário real, um atacante poderia juntar todas estas informações e **falsificar o número (spoofing)** para efetuar chamadas em nome do funcionário. Com amostras de voz recolhidas de chamadas telefônicas/vídeos/mensagens, seria viável treinar um modelo de IA para **voice cloning**. Assim, o atacante poderia fazer-se passar por um elemento do hotel e, através de técnicas de **engenharia social**, extrair detalhes do seu funcionamento interno obtendo contacto da equipa de TI ou informações sobre a infraestrutura tecnológica.

Todo este cenário pode ocorrer sem que nenhum elemento do hotel se aperceba.

2.2.3 Continuação da análise do website

```
The Registry database contains ONLY .COM, .NET, .EDU domains and
Registrars.
Domain Name: KOMYUNITIPORTO.COM
Registry Domain ID: 2956939839_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.publicdomainregistry.com
Registrar URL: www.publicdomainregistry.com
Updated Date: 2025-04-08T01:05:37Z
Creation Date: 2025-02-06T18:05:05Z
Registrar Registration Expiration Date: 2026-02-06T18:05:05Z
Registrar: PDR Ltd. d/b/a PublicDomainRegistry.com
Registrar IANA ID: 303
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registry Registrant ID: GDPR Masked
Registrant Name: GDPR Masked
Registrant Organization: GDPR Masked
Registrant Street: GDPR Masked
Registrant City: GDPR Masked
Registrant State/Province: Albufeira
Registrant Postal Code: GDPR Masked
Registrant Country: PT
Registrant Phone: GDPR Masked
Registrant Phone Ext:
Registrant Fax: GDPR Masked
Registrant Fax Ext:
Registrant Email: gdpr-masking@gdpr-masked.com
Registry Admin ID: GDPR Masked
Admin Name: GDPR Masked
Admin Organization: GDPR Masked
Admin Street: GDPR Masked
Admin City: GDPR Masked
Admin State/Province: GDPR Masked
Admin Postal Code: GDPR Masked
Admin Country: GDPR Masked
Admin Phone: GDPR Masked
Admin Phone Ext:
Admin Fax: GDPR Masked
Admin Fax Ext:
Admin Email: gdpr-masking@gdpr-masked.com
Registry Tech ID: GDPR Masked
Tech Name: GDPR Masked
Tech Organization: GDPR Masked
```

Figura 28: Host do website do restaurante

Através do comando ‘host’ é possível verificar que os detalhes do domínio do restaurante encontram-se protegidos pelo **GDPR**. Nota: O website internacional não se encontra protegido pelo **GDPR**. Isto pode ocorrer porque o *registrant* do domínio do restaurante provavelmente encontra-se em Portugal, e como tal, na união europeia, onde o **GDPR** se aplica.

```
miguel_linux@DESKTOP-1790S2J:~$ nslookup komyunitipporto.com
;; Got recursion not available from 172.24.0.1
Server:      172.24.0.1
Address:      172.24.0.1#53

Non-authoritative answer:
Name: komyunitipporto.com
Address: 100.24.208.97
Name: komyunitipporto.com
Address: 35.172.94.1
```

Figura 29: Nslookup do website do restaurante

```
miguel_linux@DESKTOP-1790S2J:~$ host komyunitipporto.com
komyunitipporto.com has address 100.24.208.97
komyunitipporto.com has address 35.172.94.1
komyunitipporto.com mail is handled by 0 komyunitipporto.com.
```

Figura 30: Host do website do restaurante

2.2.3.1 Possível spoofing de Email

Nesta secção iremos expor um pouco do nosso processo de investigação sobre o sistema de emails do restaurante do porto e um cenário de ataque que poderá ocorrer caso exista um servidor de mailing.

Como verificamos anteriormente existe um registo ‘mail exchange’ (mx)

```
miguel_linux@DESKTOP-1790S2J:~$ nslookup -type=TXT komyunitiporto.com
Server: 172.24.0.1
Address: 172.24.0.1#53

Non-authoritative answer:
*** Can't find komyunitiporto.com: No answer

Authoritative answers can be found from:
komyunitiporto.com
    origin = dns1.logotipo.net
    mail addr = suporte.site.pt
    serial = 2025022500
    refresh = 3600
    retry = 1800
    expire = 1209600
    minimum = 86400
```

Figura 31: Tentativa de lookup SPF sem resposta

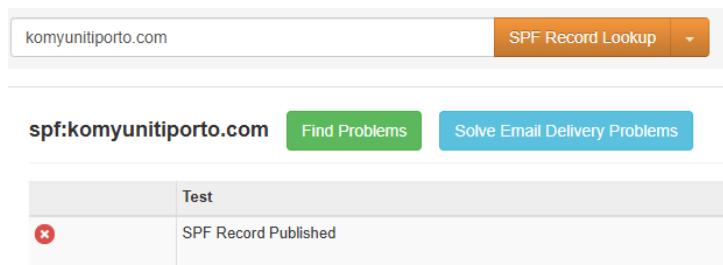


Figura 32: Tentativa de lookup SPF sem resposta 2

Verificamos que o sistema de correio do domínio não possui qualquer registo SPF. Caso um servidor de email exista, sem SPF, qualquer remetente pode enviar um email ‘@komyunitiporto.com’ e o servidor destino não tem como validar o IP, aceitando a mensagem sem sinalização de spoofing.

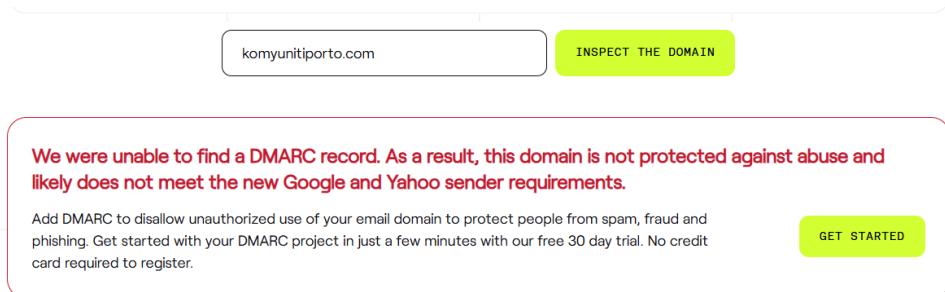


Figura 33: Verificação de DMARC

O mailing system também não possui **DMARC** (*Domain-based Message Authentication Reporting & Conformance*) o que significa não existem instruções a ser seguidas quando os receptores recebem um email em que as verificações de **SPF** (*Sender Policy Framework*) e **DKIM** (*DomainKeys Identified Mail*) falham.

A confirmação final da existência do servidor de correio e de **DKIM** exige técnicas fora do âmbito do **OSINT**, recorrendo ao envio de emails de teste ao próprio servidor **SMTP** para validar que ele aceita e processa emails.

Sem SPF, DKIM nem DMARC, qualquer servidor pode enviar mensagens ‘From: @komyunitiporto.com’ que serão aceites pelos destinatários sem sinalização, expondo o hotel a phishing.

Concluímos, no entanto, que o cenário mais realista é de que o servidor de email não exista.

3 Conclusões sobre a Parte A

As conclusões que podem ser retiradas da ‘**Parte A**’ é de que é mais fácil extrair informações sobre a infraestrutura tecnológica utilizada na empresa ‘Frutas do Cavado’ comparando com a grande corporação ‘YOTEL’.

Foi possível identificar a **empresa criadora do website** das Frutas do Cavado enquanto que na empresa ‘YOTEL’ **não foi possível** descobrir se os websites foram construídos por alguma empresa externa ou qual foi a equipa interna da YOTEL responsável pela sua construção.

No caso do website das Frutas do Cavado **foi possível** identificar uma **vaga de trabalho na empresa** que construiu o website, contendo informações sobre as tecnologias utilizadas na construção do website. Já no caso da YOTEL **não foi possível** encontrar nenhuma vaga de trabalho na área da informática.

O website das Frutas do Cavado foi construído recorrendo ao Wordpress, o que expõe o website a várias vulnerabilidades.

Foi possível verificar que e-mails de ambas as empresas já foram **expostos**.

O sistema de email de ambas as empresas é mantido por **empresas de confiança** (Clean MX e Outlook), levantando-se uma hipótese (improvável) de spoofing nos e-mails do restaurante da YOTEL.

Ambas as empresas encontram-se **sujeitos a ataques de engenharia social**, no entanto, a equipa de informática responsável pela criação do website das ‘Frutas do Cavado’ encontra-se exposta parcialmente no linkedin, ao contrário da YOTEL, em que apenas um dos membros da equipa de TI se encontra exposto publicamente.

4 Parte B

4.1 Cenário de Teste

Nesta secção são apresentadas os elementos utilizados para o cenário de teste.

Auditor

Foi utilizada a VM proposta do kali-linux.



Figura 34: VM com Kali utilizada como Auditor (exemplo na VBox)

Alvo

Foi utilizada a VM proposta do Metasploitable2.

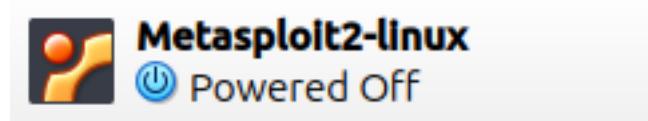


Figura 35: VM com Metasploitable2 utilizada como Alvo (exemplo na VBox)

Rede

Para a finalização do ambiente para o cenário de testes conectamos as duas VM's (auditor e alvo) à mesma “*internal network*” (nomeada TS-TP2). O kali usa o IP 172.25.2.1 e o metasploitable2 usa o IP 172.25.2.2, como podemos ver nas seguintes figuras.

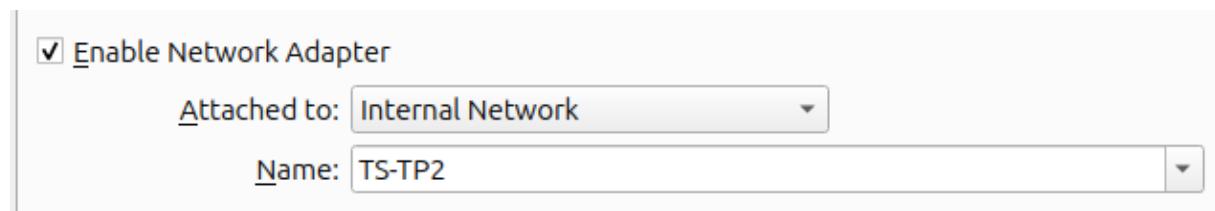


Figura 36: Rede à qual as VM's se encontram ligadas (exemplo na VBox)

```
auto eth0
iface eth0 inet static
address 172.25.2.1
netmask 255.255.255.0
gateway 172.25.2.1
```

Figura 37: /etc/network/interfaces no kali

```
auto eth0
iface eth0 inet static
address 172.25.2.2
netmask 255.255.255.0
gateway 172.25.2.1
```

Figura 38: /etc/network/interfaces no metasploitable2

4.2 Questões

4.2.1 B1

Recorrendo ao telnet entre as duas VMs e à ferramenta Wireshark foi possível visualizar os vários pacotes transmitidos entre estas.

| Index | Time | Source | Destination | Protocol | Length | Info |
|-------|--------------|------------------------|------------------------|----------|--------|-----------------------|
| 10 | 31.954804743 | 172.25.2.2 | 10.0.2.3 | DNS | 83 | Standard query 0x6dc0 |
| 11 | 31.999991715 | PCSSystemtec_04:42:... | PCSSystemtec_73:53:... | ARP | 42 | Who has 172.25.2.2? T |
| 12 | 32.000347400 | PCSSystemtec_73:53:... | PCSSystemtec_04:42:... | ARP | 60 | 172.25.2.2 is at 08:0 |
| 13 | 36.957313356 | 172.25.2.2 | 172.25.2.1 | TELNET | 78 | Do Terminal Type, Do |
| 14 | 36.957376606 | 172.25.2.1 | 172.25.2.2 | TCP | 66 | 46330 → 23 [ACK] Seq= |
| 15 | 36.958027722 | 172.25.2.2 | 172.25.2.1 | TELNET | 111 | Won't Encryption Opti |
| 16 | 36.958041515 | 172.25.2.1 | 172.25.2.2 | TCP | 66 | 46330 → 23 [ACK] Seq= |
| 17 | 36.958473711 | 172.25.2.1 | 172.25.2.2 | TELNET | 149 | Suboption Negotiate A |
| 18 | 36.958740254 | 172.25.2.2 | 172.25.2.1 | TCP | 66 | 23 → 46330 [ACK] Seq= |
| 19 | 36.959070478 | 172.25.2.2 | 172.25.2.1 | TELNET | 69 | Do Echo |
| 20 | 36.959114652 | 172.25.2.1 | 172.25.2.2 | TELNET | 69 | Won't Echo |
| 21 | 36.968462614 | 172.25.2.2 | 172.25.2.1 | TELNET | 69 | Will Echo |
| 22 | 36.968534073 | 172.25.2.1 | 172.25.2.2 | TELNET | 69 | Do Echo |
| 23 | 36.968788608 | 172.25.2.2 | 172.25.2.1 | TELNET | 686 | 620 bytes data |
| 24 | 37.012036056 | 172.25.2.1 | 172.25.2.2 | TCP | 66 | 46330 → 23 [ACK] Seq= |
| 25 | 47.371344007 | 172.25.2.1 | 172.25.2.2 | TELNET | 67 | 1 byte data |
| 26 | 47.371791699 | 172.25.2.2 | 172.25.2.1 | TELNET | 67 | 1 byte data |
| 27 | 47.371820979 | 172.25.2.1 | 172.25.2.2 | TCP | 66 | 46330 → 23 [ACK] Seq= |
| 28 | 47.675647159 | 172.25.2.1 | 172.25.2.2 | TELNET | 67 | 1 byte data |
| 29 | 47.676144580 | 172.25.2.2 | 172.25.2.1 | TELNET | 67 | 1 byte data |
| 30 | 47.676175684 | 172.25.2.1 | 172.25.2.2 | TCP | 66 | 46330 → 23 [ACK] Seq= |

Figura 39: pacotes_telnet

Depois, usando a opção do wireshark de “Follow TCP Stream” conseguimos visualizar todos estes elementos em plaintext (estes são apenas alguns dos exemplos):

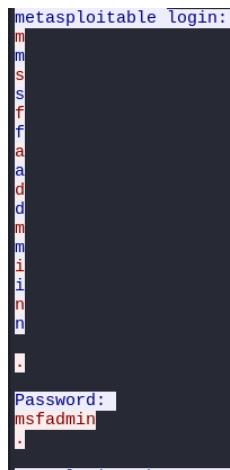


Figura 40: login_plaintext

```
msfadmin@metasploitable:~$ cat /etc/passwd
cat /etc/passwd

.

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:mail:/var/mail:/bin/sh
news:x:9:news:/var/spool/news:/bin/sh
uucp:x:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:proxy:/bin:/bin/sh
www-data:x:33:www-data:/var/www:/bin/sh
```

Figura 41: passwd_plaintext

```
msfadmin@metasploitable:~$ ifconfig
ifconfig

.

eth0      Link encap:Ethernet HWaddr 08:00:27:73:53:f8
          inet addr: 172.25.2.2 Bcast: 172.25.2.255 Mask: 255.255.255.0
          inet6 addr: fe80::a00:27ff:fe73:53f8/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST MTU: 1500 Metric: 1
          RX packets: 111 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 158 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 1000
          RX bytes: 8040 (7.8 KB) TX bytes: 17787 (17.3 KB)
          Base address: 0xd028 Memory: f0200000-f0220000

lo       Link encap: Local Loopback
          inet addr: 127.0.0.1 Mask: 255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING MTU: 16436 Metric: 1
          RX packets: 135 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 135 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 0
          RX bytes: 40109 (39.1 KB) TX bytes: 40109 (39.1 KB)
```

Figura 42: ipconfig_plaintext

Problemas de segurança:

- **Transmissão de credenciais em plaintext:** Na análise do TCP Stream, vemos o utilizador e password (msfadmin) transmitidos sem recurso a qualquer técnica criptográfica.
- **Transmissão de comandos em plaintext:** Todos os comandos executados são visíveis.
- **Transmissão de dados sensíveis em plaintext:** O conteúdo de ficheiros como “/etc/passwd” é transmitido sem proteção.
- **Ausência de mecanismos de integridade:** Não há garantia que os dados não foram alterados no caminho.
- **Vulnerabilidade a ataques Man-in-the-Middle:** Pessoas com acesso à rede podem capturar e analisar o tráfego.

Soluções sugeridas:

- Para mitigar os riscos do tráfego **Telnet** identificado, recomenda-se substituir este protocolo por **SSH**, que utiliza criptografia AES para proteger as credenciais e os comandos transmitidos.
- Propõe-se também a implementação de **autenticação de dois fatores (2FA)**.
- É recomendada a configuração de **firewalls** (ex.: iptables) para limitar as ligações a IPs confiáveis.
- A adoção de sistemas **IDS/IPS**, como Suricata, é sugerida para detetar e alertar sobre atividades anômalas, assegurando uma infraestrutura de rede mais segura e resiliente.

4.2.2 B2

4.2.2.1 B2.1

Nmap da primeira varredura (nmap -sV 172.25.2.2)

O comando ‘**nmap -sV 172.25.2.2**’ realiza uma varredura (scan) das portas do endereço detetando as versões do software utilizado.

A **flag -sV** do **Nmap** é utilizada para identificar não só quais as portas que se encontram abertas, mas também o software que se encontra em execução e a sua versão.

Como resultado da execução deste comando temos a identificação de **23 portas abertas** e os softwares que se encontram em execução com as suas respetivas versões. No entanto, com este comando, nem todas as versões de serviços são **100% específicas**: por exemplo, para o serviço “**netbios-ssn**”, temos a versão aproximada “**Samba smbd 3.X - 4.X**”.

```

net: shown: 971 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login?      Netkit rshd
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1

```

Figura 43: B2: nmap da varredura 1

Nmap da segunda varredura (nmap -sV -p 80 172.25.2.2)

Este comando realiza a mesma análise que o comando utilizado anteriormente mas focando-se apenas na **porta 80**.

Como resultado podemos verificar que à porta 80 está associado o serviço “**http**”, com a versão específica “Apache httpd 2.2.8 ((Ubuntu) DAV/2)”.

```

PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.2.8 ((Ubuntu) DAV/2)
MAC Address: 08:00:27:73:53:E8 (PC Systemtechnik/Oracle)

```

Figura 44: B2: nmap da varredura 2

Nmap da terceira varredura (nmap -sV –script vulners 172.25.2.2)

Tal como os comandos anteriores, este comando Nmap realiza a varredura de portas com detecção de serviços e as respetivas versões.

No entanto, devido ao argumento “**–script vulners**”, este comando também associa cada elemento a vulnerabilidades conhecidas, sendo estas extraídas da base de dados da API pública do ‘Vulners.com’.

Como resultado, para cada serviço/versão temos associada uma **lista de vulnerabilidades** com os seguintes pontos:

- Identificador CVE para cada vulnerabilidade detectada.
- Pontuação CVSS que indica a severidade de cada vulnerabilidade.
- Link associado.
- Possível campo “**EXPLOIT**”, que indica se existe um *exploit* disponível publicamente para a vulnerabilidade identificada.

```

3306/tcp open  mysql      MySQL 5.0.51a-3ubuntu5
| vulners:
|_ cpe:/a:mysql:mysql:5.0.51a-3ubuntu5:
|   SSV:19118  8.5   https://vulners.com/seebug/SSV:19118  *EXPLOIT*
|   CVE-2017-15945 7.8   https://vulners.com/cve/CVE-2017-15945
|   SSV:15006  6.8   https://vulners.com/seebug/SSV:15006  *EXPLOIT*
|   CVE-2009-4028 6.8   https://vulners.com/cve/CVE-2009-4028
|   SSV:15004  6.0   https://vulners.com/seebug/SSV:15004  *EXPLOIT*

```

Figura 45: B2: parte do nmap da varredura 3 (porta 3306)

Nmap da quarta varredura (nmap -A 172.25.2.2)

Este comando, devido ao argumento “**-A**”, realiza uma varredura agressiva, que combina várias técnicas para colecionar o máximo de informações sobre o alvo.

Especificamente, ele realiza:

- **Deteção do Sistema Operacional (-O)**: tenta identificar o sistema operacional do alvo.

- **Deteção das Versões dos Serviços (-sV):** identifica os softwares nas portas abertas e as suas versões.
- **Execução de Scripts “default” (-sC):** executa scripts básicos de segurança e coleção de informações (Contrariamente ao “-script vulners”, que foca especificamente em vulnerabilidades, o “-sC” executa uma variedade de scripts para diferentes finalidades).
- **Traceroute:** Traça a rota dos pacotes até o alvo e mostra os “hops” entre a máquina e o destino.

A varredura agressiva com o nmap -A revelou um perfil mais completo do sistema alvo comparado à varredura básica com -sV.

O **sistema operativo** foi identificado como **Linux** com kernel entre as versões 2.6.9 e 2.6.33.

Foram detectados os mesmos **23 serviços** da varredura anterior, mas com detalhes adicionais, como: **FTP** com login anônimo permitido, chaves de host **SSH** capturadas, **comandos suportados** no servidor de email e **configurações de segurança** do Samba definidas de forma insegura.

Os scripts de segurança executados automaticamente revelaram vulnerabilidades de configuração, dados de autenticação e parâmetros específicos dos serviços.

Também foram recolhidas informações de rede como **endereço MAC**, **fabricante da interface** (VirtualBox) e **relações de domínio**.

Foram ainda obtidos metadados como **certificados SSL** e detalhes de configuração de serviços, como por exemplo o **RPC**.

É importante notar que esta varredura foi consideravelmente mais demorada (cerca de 128 segundos) e gerou mais tráfego de rede, tornando-a mais facilmente detetável por sistemas de segurança. No entanto, forneceu um perfil detalhado da máquina alvo, expondo múltiplas informações críticas que seriam valiosas num contexto de teste de intrusão.

```
2221/tcp open  ssh          172.25.2.2 173.1
3306/tcp open  mysql        MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 11
|   Capabilities flags: 43564
|   Some Capabilities: Support41Auth, SupportsTransactions
|   Status: Autocommit
|_  Salt: @l#dQHSIUPqh5nLne1bz
```

Figura 46: B2: parte do nmap da varredura 4 (porta 3306)

4.2.2.2 B2.2

Tráfego da primeira varredura (nmap -sV 172.25.2.2)

- N° pacotes: 2819 pacotes
- Tentativas de conexão TCP (pacotes SYN) para as portas comuns
- Handshakes TCP completos (SYN, SYN-ACK, ACK) para portas abertas
- Algumas conexões estabelecidas seguidas de tráfego adicional para identificação de serviços
- Tentativas de comunicação com protocolos específicos para determinar versões

Detectabilidade:

- Volume moderado de tráfego
- Duração: 54.7 segundos

Tráfego da segunda varredura(nmap -sV -p 80 172.25.2.2)

- Semelhante ao anterior, com foco exclusivo na porta 80
- Handshake TCP apenas para a porta 80
- Múltiplas requisições HTTP para tentar identificar a versão do servidor web

Detectabilidade:

- Volume de tráfego muito menor, concentrado numa única porta.
- Duração mais curta: 10.4 segundos
- Menos suspeito para um NIDS (pois pode parecer tráfego HTTP legítimo, embora o User-Agent do Nmap seja detectável por regras específicas do Suricata, como o alerta “ET SCAN Possible Nmap User-Agent Observed”).

Tráfego da terceira varredura (nmap -sV –script vulners 172.25.2.2)

- Similar ao padrão do -sV, mas com algum tráfego adicional
- Não explora vulnerabilidades diretamente, pede vulnerabilidades conhecidas ao “Vulners”
- Alguns pacotes de consulta ao “Vulners”

Detectabilidade:

- Volume de tráfego ligeiramente maior que o -sV básico
- Duração ligeiramente maior
- É eficaz para mapear CVEs conhecidas, de forma passiva e não invasiva

Tráfego da quarta varredura (nmap -A 172.25.2.2)

- Maior volume de tráfego entre todas as varreduras
- Tentativas de identificação de SO (pacotes com flags TCP incomuns)
- Execução de múltiplos scripts NSE que geram tráfego variado
- Tentativas de traceroute (pacotes ICMP ou UDP)

Detectabilidade:

- Volume muito alto de tráfego, visivelmente superior às outras varreduras.
- Longa duração: 130 segundos
- Padrões de tráfego altamente variados e distintivos
- Tipos de pacotes incomuns que são mais facilmente detetáveis
- Sequências de pacotes que correspondem a assinaturas conhecidas de ferramentas de varredura

4.2.2.3 B2.3

Análise do Suricata na primeira varredura (nmap -sV 172.25.2.2)

O relatório do Suricata referente à primeira varredura revela alguns alertas que confirmam a utilização do Nmap com a flag -sV, usada para detetar versões de serviços.

Um dos alertas recorrentes é o “Applayer Detect protocol only one direction”, que mostra que o Suricata identificou tráfego onde apenas uma das direções da comunicação apresentou dados válidos para deteção do protocolo de aplicação, o que é típico de varreduras de serviços.

SMTP – Erros de Resposta

Ausência de mensagem de “welcome” do servidor SMTP (no server welcome message) Resposta inválida (SMTP invalid reply)

Durante a tentativa do Nmap de identificar a versão do serviço SMTP, ocorreram respostas inesperadas, provavelmente devido à forma como o Nmap testa serviços que não completam conexões de forma convencional.

Para o serviço SMTP, surgiram alertas associados à ausência de mensagem de boas-vindas (“no server

welcome message") e a respostas inválidas ("SMTP invalid reply"). Estes podem ocorrer quando o Nmap tenta interagir com serviços de forma automatizada e fora dos parâmetros habituais, resultando em respostas inesperadas.

Foram ainda detetados acessos a serviços web nas portas 80 e 8180, com o alerta "ET SCAN Possible Nmap User-Agent Observed". Isto que mostra que o Suricata identificou que o Nmap tentou aceder a serviços web (portas 80 e 8180), deixando rastos típicos como o seu User-Agent.

Tal como foi referido na pergunta B2.2, não foi produzida uma quantidade alta de pacotes. O mesmo aconteceu no relatório do suricata: foram produzidos alguns avisos de deteção, mas não uma quantidade muito alta.

Análise do Suricata na segunda varredura (nmap -sV -p 80 172.25.2.2)

Nesta varredura o Suricata apenas deu o seguinte aviso (várias vezes): ET SCAN Possible Nmap User-Agent Observed.

Tal como referido anteriormente esta é a identificação de que o Nmap tentou aceder a serviços web (neste caso exclusivamente na porta 80), deixando rastos típicos como o seu User-Agent.

Como podemos ver, o número baixo de avisos do Suricata é proporcional ao número de pacotes apanhados pelo *wireshark* (também reduzido). O que faz sentido tendo em conta que esta análise Nmap se focou apenas na porta 80.

Análise do Suricata na terceira varredura (nmap -sV –script vulners 172.25.2.2)

O relatório de alertas gerado pelo Suricata para esta varredura é semelhante ao da primeira (nmap -sV 172.25.2.2), o que é esperado dado o comportamento quase idêntico entre os dois comandos.

Embora este comando inclua a execução do script "vulners", que procura por vulnerabilidades conhecidas com base nas versões dos serviços detetados, essa verificação é feita localmente pelo auditor, através de consultas a uma base de dados externa (Vulners.com) — e não implica interação adicional com o sistema alvo.

Como consequência, o tráfego gerado para o alvo é praticamente o mesmo do "-sV", e não há atividade extra que desperte novos alertas no Suricata. A análise de vulnerabilidades ocorre após a identificação dos serviços e não requer pacotes adicionais enviados ao alvo.

Análise do Suricata na quarta varredura (nmap -A 172.25.2.2)

O relatório relativo a esta varredura apresenta um número significativamente mais elevado de alertas, o que confirma a utilização do Nmap de forma mais evidente do que nas varreduras anteriores.

Primeiramente, existem numerosas entradas com o alerta específico "ET SCAN Possible Nmap User-Agent Observed", que indicam diretamente a deteção da assinatura característica do Nmap nos cabeçalhos HTTP, categorizadas como "Web Application Attack".

Além disso, o padrão de tráfego mostra tentativas de conexão sequenciais a uma variedade de portas, incluindo 80, 8180, 21, 25, 445, 5432, 514, 6667, 5900 e 2121, o que é consistente com o comportamento do Nmap ao realizar um levantamento completo de serviços disponíveis.

O relatório também revela alertas como "SURICATA SMB malformed request dialects" e "SURICATA Applayer Detect protocol only one direction", que são indicativos das técnicas do Nmap para determinar versões específicas de software em execução no sistema alvo.

Foram ainda detetados comandos IRC específicos na porta 6667, bem como várias tentativas de enviar dados incompatíveis com os protocolos esperados, demonstradas pelos alertas "SURICATA TLS invalid record type".

A quantidade alta de avisos neste varredura faz sentido tendo em conta a agressividade da flag "-A", causando mais ruído e mais possibilidade de deteção por um sistema como o Suricata. Podemos

também verificar que o número alto de avisos observados é acompanhado pelo número alto de pacotes obtidos no wireshark (referido na pergunta anterior).

4.2.3 B3

O levantamento realizado na Parte A permite recolher informações de domínio público sobre a infra-estrutura de uma organização sem interagir diretamente com os seus sistemas. Esta recolha prévia de informações oferece várias vantagens para otimizar as varreduras subsequentes:

1. Redução do scope da varredura

Os levantamentos passivos permitem identificar:

- Blocos de IPs efetivamente utilizados pela organização
- Subdomínios ativos e os seus respetivos serviços
- Tecnologias e versões de software utilizadas

Em vez de realizar varreduras amplas que analisam grandes blocos de IP ou todas as portas possíveis, é possível focar apenas nos alvos relevantes identificados previamente. Varreduras mais focadas geram menos tráfego, reduzindo significativamente a probabilidade de detecção.

2. Seleção inteligente de técnicas de varredura

Conhecendo antecipadamente:

- Tipos de sistemas operativos utilizados
- Firewalls e soluções de segurança implementadas
- Configurações de rede

É possível selecionar técnicas de varredura específicas que sejam menos propensas a serem detetadas pelos mecanismos de proteção identificados. Por exemplo, se o footprinting revelar a presença de um IDS específico, é possível escolher técnicas de varredura que evitem os seus gatilhos conhecidos.

3. Temporização e técnicas de evasão apropriadas

O conhecimento prévio sobre:

- Horários de menor atividade da rede alvo
- Políticas de monitorização implementadas
- Configurações de registo (logging)

Permite estabelecer estratégias como:

- Ajuste da velocidade de varredura
- Distribuição das varreduras ao longo do tempo
- Utilização de técnicas de ofuscação específicas para os sistemas alvo

4. Detecção de honeypots e armadilhas

A análise passiva pode revelar inconsistências que indicam a presença de “honeypots”, permitindo evitá-los durante as varreduras ativas e reduzindo o risco de detecção.

4.2.4 B4

Após a aplicação das regras de firewall com iptables, que bloqueiam tráfego TCP para as portas 513 e 6000, foram repetidas as quatro varreduras da questão B2:

- nmap -sV IP_alvo
 - As portas 513 e 6000, que anteriormente apareciam como “open”, passaram agora a surgir como “filtered”. Isto pode indicar que o Nmap não recebeu qualquer resposta (nem SYN/ACK nem RST), presumindo que a firewall está a bloquear silenciosamente o tráfego para essas portas.
- nmap -sV -p 80 IP_alvo

- Esta varredura não apresentou qualquer alteração nos resultados. Como a firewall foi configurada apenas para bloquear as portas 513 e 6000, a análise da porta 80 manteve-se inalterada.
- **nmap -sV -script vulners IP_alvo**
 - O Nmap não conseguiu obter informações de vulnerabilidades para as portas 513 e 6000, que surgem agora também como “filtered”. Isto impede que o Nmap execute scripts de detecção nessas portas, reduzindo a visibilidade da superfície de ataque.
- **nmap -A IP_alvo**
 - Tal como na varredura com -sV, as portas anteriormente acessíveis (513 e 6000) agora são identificadas como “filtered”.

513/tcp filtered login

Figura 47: login_filtered

6000/tcp filtered X11

Figura 48: X11_filtered

4.2.5 B5

nmap -Pn -sA 172.25.2.2

Este tipo de varredura envia pacotes com a flag ACK para detetar regras de firewall. Os resultados mostraram que as portas 513 e 6000 surgem como “filtered”, o que confirma que a firewall está a bloquear o tráfego TCP de entrada nessas portas. Todas as restantes portas são identificadas como “unfiltered”.

```
Not shown: 998 unfiltered tcp ports
PORT      STATE      SERVICE
513/tcp    filtered  login
6000/tcp   filtered X11
```

Figura 49: B5 varredura 1

nmap -Pn -sF 172.25.2.2

nmap -Pn -sN 172.25.2.2

nmap -Pn -sX 172.25.2.2

Estas três técnicas utilizam pacotes com flags incomuns ou ausentes (FIN, Null, e Xmas) com o objetivo de evadir sistemas de deteção. O resultado foi semelhante para todas: todas as portas anteriormente detetadas como abertas passaram a surgir como “open|filtered”.

O resultado “open|filtered” ocorre porque tanto portas abertas quanto portas filtradas por firewall não enviam resposta (RST), o que torna impossível distingui-las. Esta ambiguidade é uma limitação destas técnicas de varredura.

| Not shown: 977 closed tcp ports (res) | | |
|---------------------------------------|---------------|--------------|
| PORT | STATE | SERVICE |
| 21/tcp | open filtered | ftp |
| 22/tcp | open filtered | ssh |
| 23/tcp | open filtered | telnet |
| 25/tcp | open filtered | smtp |
| 53/tcp | open filtered | domain |
| 80/tcp | open filtered | http |
| 111/tcp | open filtered | rpcbind |
| 139/tcp | open filtered | netbios-ssn |
| 445/tcp | open filtered | microsoft-ds |
| 512/tcp | open filtered | exec |
| 513/tcp | open filtered | login |
| 514/tcp | open filtered | shell |
| 1099/tcp | open filtered | rmiregistry |
| 1524/tcp | open filtered | ingreslock |
| 2049/tcp | open filtered | nfs |
| 2121/tcp | open filtered | ccproxy-ftp |
| 3306/tcp | open filtered | mysql |
| 5432/tcp | open filtered | postgresql |
| 5900/tcp | open filtered | vnc |
| 6000/tcp | open filtered | X11 |
| 6667/tcp | open filtered | irc |
| 8009/tcp | open filtered | ajp13 |
| 8180/tcp | open filtered | unknown |

Figura 50: B5 varreduras 2,3,4

nmap -Pn -sU -p 513,6000 172.25.2.2

Nesta varredura por UDP, ambas as portas analisadas foram identificadas como “closed”. Enquanto as portas TCP 513 e 6000 estão bloqueadas para entrada (filtered), as mesmas portas para UDP estão fechadas (closed) e não filtradas. Isto revela que a firewall foi configurada para bloquear o tráfego TCP nestas portas, mas permite que as mensagens ICMP “port unreachable” sejam enviadas para o tráfego UDP.

Para além disso, notamos que o Nmap diferencia corretamente os serviços consoante o protocolo: na porta 513/tcp é identificado o serviço “login” (rlogin), enquanto na mesma porta 513/udp é identificado o serviço “who” (rwho).

| PORT | STATE | SERVICE |
|----------|--------|---------|
| 513/udp | closed | who |
| 6000/udp | closed | X11 |

Figura 51: B5 varredura 5

4.2.6 B6

A varredura do sistema alvo Metasploitable 2 (IP 172.25.2.2) com o comando nikto -h 172.25.2.2 revelou vulnerabilidades críticas no servidor web Apache 2.2.8, operativo na porta 80, que amplificam os riscos de segurança. Os resultados destacam versões desatualizadas do Apache (2.2.8) e PHP (5.2.4-2ubuntu5.10), suscetíveis a vulnerabilidades públicas que permitem execução de código remoto ou negação de serviço. A ausência de cabeçalhos X-Frame-Options e X-Content-Type-Options expõe o sistema a ataques de clickjacking, comprometendo a confidencialidade de dados dos utilizadores. O método HTTP TRACE ativo possibilita ataques de Cross-Site Tracing (XST), permitindo a extração de cookies de sessão. Diretórios navegáveis (/doc/, /test/, /icons/) e a funcionalidade phpinfo() expõem configurações sensíveis, enquanto uma instalação desprotegida do phpMyAdmin e um possível ficheiro wp-config.php aumentam o risco de acesso não autorizado a bases de dados MySQL, potencialmente comprometendo a integridade e confidencialidade do sistema. Estas fragilidades indicam uma gestão inadequada da segurança, tornando o servidor vulnerável a explorações automatizadas.

O tráfego gerado, capturado via Wireshark e monitorizado pelo Suricata, consistiu em requisições HTTP GET e HEAD a caminhos como /phpMyAdmin/ e /wp-config.php, com cerca de 500-1000 pacotes em 20-30 segundos. O Suricata detetou o User-Agent “Nikto/2.x”, gerando alertas como “ET

SCAN Nikto User-Agent Observed” e “ET WEB_SERVER HTTP TRACE Method Enabled”, indicando maior detectabilidade que o comando nmap -sV -p 80 (100 pacotes, 10.4 segundos, alerta “ET SCAN Possible Nmap User-Agent Observed”), mas menor que o nmap -A (milhares de pacotes, 130 segundos, alertas variados como “SURICATA SMB malformed request dialects”). Comparativamente, o nmap -sV -script vulners identificou CVEs do Apache, mas não detetou configurações inseguras como diretórios navegáveis ou phpMyAdmin, enquanto o nmap -A revelou 23 portas abertas e o sistema operativo Linux (2.6.9-2.6.33), sendo menos específico na porta 80. O Nikto complementa o Nmap ao expor vulnerabilidades web críticas, cuja exploração pode levar à compromissão total do sistema, enquanto o Nmap contextualiza riscos em outros serviços. O levantamento passivo da Parte A, que identificou tecnologias como WordPress, poderia otimizar a varredura, reduzindo requisições e detectabilidade. As descobertas exigem atualizações de software, desativação do método TRACE, proteção do phpMyAdmin, e implementação de cabeçalhos de segurança para mitigar os riscos identificados.

```

- Nikto v2.5.0/
+ Target Host: 172.25.2.2
+ Target Port: 80
+ GET /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ GET /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options:
+ GET /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/:
+ GET /index: Unknown header 'tcm' found, with contents: list.
+ GET /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.lbcnCloud.com/vulnerabilities/8275:
+ HEAD Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ ERBFPOQ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ TRACE /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing:
+ GET /phpInfo.php: Output from the phpInfo() function was found.
+ GET /doc/: Directory indexing found.
+ GET /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: CVE-1999-0678:
+ GET /?PHP8885F2A0-3C92-11d3-A3A9-4C7800C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184:
+ GET /?PHE9568F36-0428-11d2-A769-00A0A01ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184:
+ GET /?PHE9568F34-0428-11d2-A769-00A0A01ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184:
+ GET /?PHE9568F35-0428-11d2-A769-00A0A01ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184:
+ GET /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ GET /phpMyAdmin/ChangeLog: Server may leak inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mtime: Tue Dec 9 12:24:00 2008. See: CVE-2003-1418:
+ GET /phpMyAdmin/changelog; phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ GET /test/: Directory indexing found.
+ GET /test/: This might be interesting.
+ GET /phpInfo.php: PHP is installed, and a test script which runs phpInfo() was found. This gives a lot of system information. See: CWE-552:
+ GET /icons/: Directory indexing found.
+ GET /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/:
+ GET /phpMyAdmin/: phpMyAdmin directory found.
+ GET /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ GET /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://typo3.org/:
+ GET /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.

```

Figura 52: B6 relatório do nikto

4.2.7 B7

| Sev | CVSS | VPR | EPS | Name | Family | Count | Actions |
|------------|--------|-----|--------|--|-----------------------|-------|---------|
| □ CRITICAL | 10.0 * | 8.4 | 0.6132 | UnrealIRCd Backdoor Detection | Backdoors | 1 | 🔗 |
| □ CRITICAL | 10.0 | | | Canonical Ubuntu Linux SEoL (8.04.x) | General | 1 | 🔗 |
| □ CRITICAL | 10.0 * | | | VNC Server 'password' Password | Gain a shell remotely | 1 | 🔗 |
| □ CRITICAL | 9.8 | | | SSL Version 2 and 3 Protocol Detection | Service detection | 2 | 🔗 |
| □ CRITICAL | 9.8 | | | Bind Shell Backdoor Detection | Backdoors | 1 | 🔗 |
| □ MIXED | ... | ... | ... | Apache Tomcat (Multiple Issues) | Web Servers | 4 | 🔗 |
| □ CRITICAL | ... | ... | ... | SSL (Multiple Issues) | Gain a shell remotely | 3 | 🔗 |
| □ HIGH | 7.5 * | 7.4 | 0.4664 | rlogin Service Detection | Service detection | 1 | 🔗 |
| □ HIGH | 7.5 * | 7.4 | 0.4664 | rsh Service Detection | Service detection | 1 | 🔗 |
| □ HIGH | 7.5 | 5.9 | 0.7865 | Samba Badlock Vulnerability | General | 1 | 🔗 |
| □ HIGH | 7.5 | | | NFS Shares World Readable | RPC | 1 | 🔗 |
| □ MIXED | ... | ... | ... | SSL (Multiple Issues) | General | 28 | 🔗 |
| □ MIXED | ... | ... | ... | ISC Bind (Multiple Issues) | DNS | 5 | 🔗 |
| □ MEDIUM | 6.5 | | | TLS Version 1.0 Protocol Detection | Service detection | 2 | 🔗 |

Figura 53: Vulnerabilidades Metasploitable 2 - Nessus



Figura 54: Resultado Final do Adv. Scan - Nessus

4.2.7.1 B7.1

A análise de vulnerabilidades conduzida com o Nessus no sistema Metasploitable 2 (IP 172.25.2.2), a partir do Kali Linux (IP 172.25.2.1) na rede interna TS-TP2, revelou um conjunto significativo de fragilidades críticas nos serviços expostos. Utilizando o “Advanced Scan”, o Nessus identificou portas abertas e serviços suscetíveis a exploits, como o FTP na porta 21 (vsftpd 2.3.4, com permissão de login anónimo), o SSH na porta 22 (OpenSSH 4.7p1 Debian-8ubuntu1, suportando algoritmos criptográficos fracos como hmac-md5 e diffie-hellman-group1-sha1) e o HTTP na porta 80 (Apache 2.2.8, com configurações potencialmente inseguras). Adicionalmente, foram encontrados serviços como Samba 3.0.20, MySQL 5.0.51a, PostgreSQL e ISC BIND 9.4.2, todos associados a vulnerabilidades conhecidas. Entre as falhas mais severas, destaca-se a CVE-2008-0166, relacionada com uma implementação defeituosa do OpenSSL em sistemas baseados em Debian, que compromete a robustez das chaves criptográficas e possibilita ataques Man-in-the-Middle. Outras vulnerabilidades críticas incluem o Samba Badlock, que exige atualização para versões iguais ou superiores a 4.2.11, e uma falha de negação de serviço no ISC BIND, afetando versões anteriores a 9.11.22. Comparativamente às varreduras realizadas na questão B2, o Nessus enriqueceu os resultados do Nmap (opções -sV, -A, vulners) ao associar identificadores CVE específicos e detalhar configurações inseguras, como os algoritmos criptográficos frágeis no SSH. Por exemplo, enquanto o Nmap identificou a versão do OpenSSH, o Nessus correlacionou-a com a CVE-2008-0166 e especificou os algoritmos vulneráveis suportados. Em relação à questão B5, as portas 513 e 6000, bloqueadas por regras de firewall e classificadas como “filtered” em varreduras TCP ou “closed” em UDP, não foram reportadas como vulneráveis pelo Nessus, refletindo uma superfície de ataque reduzida. Já na questão B6, o Nessus destacou-se em relação ao Nikto por abranger a totalidade da pilha de serviços, indo além do foco exclusivo em HTTP, o que proporcionou uma avaliação mais abrangente das vulnerabilidades presentes.

4.2.7.2 B7.2

No que concerne às notificações do Suricata, verificou-se que alertas como “ET SCAN Possible Nmap User-Agent Observed” e “SURICATA Applayer Detect protocol only one direction” não encontram correspondência direta com as vulnerabilidades identificadas pelo Nessus. Esta discrepância decorre das finalidades distintas das ferramentas: o Suricata concentra-se na deteção de padrões anómalos de tráfego, como atividades de varredura do Nmap, que refletem ações do auditor e não fragilidades intrínsecas ao sistema alvo. Por exemplo, o alerta “ET SCAN” está associado ao uso do Nmap durante o teste, não representando uma vulnerabilidade explorável no Metasploitable 2. De forma semelhante, notificações como “SMTP invalid reply” emergem de interações atípicas do Nmap com serviços, sem indicarem falhas de segurança concretas. Em contrapartida, o Nessus prioriza a identificação de vulnerabilidades específicas, como CVEs e configurações inseguras, desconsiderando artefactos resultantes do processo de varredura. Além disso, o Suricata pode registrar tentativas de conexão em portas filtradas, como 513 e 6000, enquanto o Nessus não as avalia devido à ausência de resposta, evidenciando a diferença entre análise de tráfego e deteção de falhas exploráveis. Assim, a integração dos resultados do Nessus com as notificações do Suricata sublinha a complementaridade das abordagens, com o Nessus a destacar a urgência de medidas corretivas, como a atualização do OpenSSL, a desativação de algoritmos criptográficos obsoletos no SSH e a aplicação de correções nos serviços expostos, reforçando a necessidade de uma estratégia de mitigação robusta.

5 Referências

OSINT Framework

<https://www.frutasdocavado.com>

<https://uebyou.pt>

<https://www.empregoxl.com/emprego/435319/web-designer-developer-medida-estagio-ativar-em-braga/>

<https://nmap.org>

<https://www.tenable.com/>