



2025

MAPEAMENTO DE REDE CORPORATIVA



CURSO FORMAÇÃO CIBERSEC

Aluno:
Claudio Mendonça

Professor:
Jose Menezes

Instrutores:
Gilson Andrade
João Pedro Belo



Sumário

Sumário Executivo	3
Objetivo	4
Escopo	5
Metodologia	6
Etapas do Reconhecimento das Redes	6
Descoberta de Hosts	12
Scan de Portas	18
Extras úteis	46
Arp	46
Organização dos Resultados	50
Resumo da Organização:	51
Inventário Final - Tabela Descritiva	51
Diagrama	53
Diagnóstico	54
Recomendações	57
Segurança	58
Monitoramento	60
Documentação	61
Auditoria	62
Extras	63
Plano de Ação (modelo 80/20)	64
Conclusão	67
Referência Bibliográficas	68
Anexos	69

Sumário Executivo

Este projeto foi desenvolvido com o objetivo de realizar uma análise detalhada da infraestrutura de rede, identificando os dispositivos conectados, os serviços em operação e possíveis riscos à segurança. A abordagem adotada permitiu mapear a rede de forma abrangente, garantindo uma visão clara de sua estrutura e funcionamento.

Inicialmente, foi realizado um levantamento dos equipamentos conectados à rede e testada a comunicação entre eles. Em seguida, foram analisados os serviços ativos, como servidores de arquivos, bancos de dados e páginas web, além de verificar possíveis vulnerabilidades, como entradas desprotegidas que poderiam ser exploradas.

As informações coletadas foram organizadas de forma estruturada para facilitar a interpretação e análise. Com base nos resultados, foi elaborado um plano de ação com recomendações práticas para aprimorar a segurança e eficiência da rede, incluindo medidas como restrição de acessos desnecessários, fortalecimento de autenticação e monitoramento contínuo.

Em resumo, o projeto forneceu um diagnóstico completo da rede, destacando pontos fortes e áreas que requerem atenção, e apresentou soluções para garantir maior proteção e desempenho da infraestrutura.

Objetivo

Este projeto faz parte do Módulo 1 da Trilha de Formação em Cybersecurity e tem como objetivo simular uma rede corporativa segmentada utilizando Docker. O ambiente criado representa uma empresa fictícia com diferentes sub-redes, estações de trabalho, servidores e dispositivos pessoais.

O desafio proposto é assumir o papel de um analista de segurança e realizar o mapeamento completo dos ativos e sub-redes disponíveis. A partir disso, deve identificar máquinas acessíveis, determinar os propósitos das sub-redes, e elaborar um inventário técnico contendo informações como IPs, MAC, nome do equipamento seu grupo de trabalho e outros detalhes.

Por fim, o projeto culmina na criação de um relatório técnico com diagnóstico, recomendações e um plano de ação baseado na regra 80/20, permitindo o desenvolvimento de habilidades práticas em reconhecimento de rede e análise de exposição.

Escopo

No projeto, foi identificado um arquivo oculto chamado 'ANOTACAO-ULTIMO-SCAN.TXT' no equipamento analyst (equipamento do analista), contendo um roteiro detalhado de comandos utilizados para mapear redes e ativos. Este roteiro manual inclui etapas como reconhecimento de redes, teste de conectividade, descoberta de hosts, análise de portas e serviços, além de organização e backup dos resultados.

Etapas do Reconhecimento das Redes

- Identificar as interfaces de rede e endereços IP disponíveis.
- Salvar os resultados em arquivos para referência futura.

Descoberta de Hosts

- Utilizar Nmap para realizar um ping scan e identificar hosts ativos em cada sub-rede.
- Organizar os resultados em arquivos separados.

Scan de Portas

- Utilizar Rustscan para identificar portas abertas nos hosts descobertos.

Análise de Serviços

- Executar scripts específicos do Nmap para analisar serviços como FTP, MySQL, LDAP, SMB e HTTP.
- Salvar os resultados em arquivos organizados.

Extras

- Utilizar ferramentas como arp para mapear endereços IP e dispositivos na rede.

Organização dos Resultados

- Criar diretórios para organizar os arquivos gerados e facilitar o acesso.

Metodologia

A metodologia utilizada neste projeto segue um processo estruturado para mapear redes e ativos, identificar serviços e vulnerabilidades, e organizar os resultados para análise.

Etapas do Reconhecimento das Redes

Início com o reconhecimento da rede corporativo, vou aplicar uma metodologia de scan profissional em duas fases para mapear a infraestrutura de uma empresa fictícia, usando as ferramentas certas para cada tarefa. Com o objetivo de reconhecimento inicial, descobrindo em quais redes estou conectado.

Utilizando como base uma documentação do último analista do 'ANOTACAO-ULTIMO-SCAN.TXT':

Iniciei com os comandos `ip a` e `ip a | grep inet` para pegar as informações da rede:

```
(root@788e6630118f) - [ /home/analyst ]
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host proto kernel_lo
       valid_lft forever preferred_lft forever
2: eth0@if28: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
   link/ether 5e:38:c3:8d:02:90 brd ff:ff:ff:ff:ff:ff link-netnsid 0
   inet 10.10.50.6/24 brd 10.10.50.255 scope global eth0
       valid_lft forever preferred_lft forever
3: eth1@if29: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
   link/ether 7a:e8:a3:c8:c4:5f brd ff:ff:ff:ff:ff:ff link-netnsid 0
   inet 10.10.30.2/24 brd 10.10.30.255 scope global eth1
       valid_lft forever preferred_lft forever
4: eth2@if30: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
   link/ether be:f8:9b:7d:60:6d brd ff:ff:ff:ff:ff:ff link-netnsid 0
   inet 10.10.10.2/24 brd 10.10.10.255 scope global eth2
       valid_lft forever preferred_lft forever
```

Figura 1: Print 01 - Comando `ip a`.

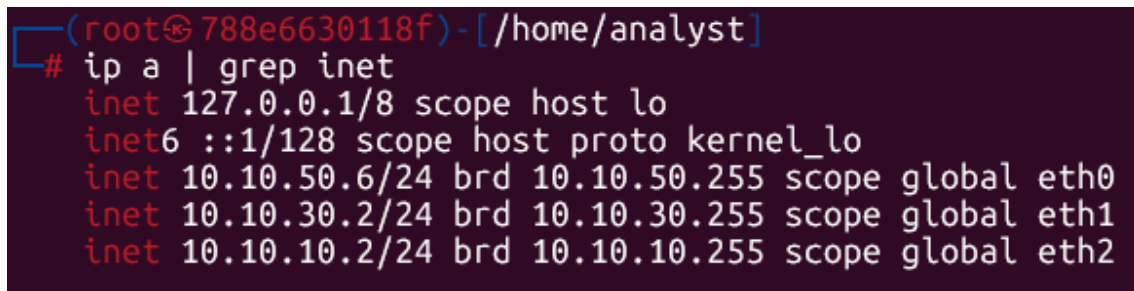
Aqui está uma tabela de descrição da rede baseada nas informações fornecidas do equipamento utilizado pelo analista (analyst):

Interface	Tipo	Estado	Endereço IPv4	Máscara	Endereço IPv6	MAC Address	MTU
lo	Loopback	UP	127.0.0.1	/8	::1/128	00:00:00:00:00:00	65536
eth0@if28	Ethernet	UP	10.10.50.6	/24	-	5e:38:c3:8d:02:90	1500
eth1@if29	Ethernet	UP	10.10.30.2	/24	-	7a:e8:a3:c8:c4:5f	1500
eth2@if30	Ethernet	UP	10.10.10.2	/24	-	be:f8:9b:7d:60:6d	1500

Detalhes:

- **Interface:** Nome da interface de rede.
- **Tipo:** Tipo de conexão (Loopback ou Ethernet).
- **Estado:** Estado atual da interface (UP indica ativa).
- **Endereço IPv4:** Endereço IPv4 atribuído à interface.
- **Máscara:** Máscara de sub-rede associada ao IPv4.
- **Endereço IPv6:** Endereço IPv6 atribuído (se aplicável).
- **MAC Address:** Endereço físico da interface.
- **MTU:** Unidade máxima de transmissão (Maximum Transmission Unit).

O comando ``ip a | grep inet`` é usado para listar os endereços IP atribuídos às interfaces de rede no sistema.



```
(root@788e6630118f) - [/home/analyst]
# ip a | grep inet
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host proto kernel_lo
inet 10.10.50.6/24 brd 10.10.50.255 scope global eth0
inet 10.10.30.2/24 brd 10.10.30.255 scope global eth1
inet 10.10.10.2/24 brd 10.10.10.255 scope global eth2
```

Figura 2: Print 02 - Comando ``ip a | grep inet``.

Aqui está uma explicação detalhada:

Comando:

- **ip a:** Mostra informações detalhadas sobre todas as interfaces de rede, incluindo endereços IP, estado das interfaces, MTU, etc.
- **| grep inet:** Filtra a saída para mostrar apenas as linhas que contêm "inet" (endereços IPv4) ou "inet6" (endereços IPv6).

Saída Explicada:

1. **inet 127.0.0.1/8 scope host lo**

- **127.0.0.1/8:** Endereço IPv4 de loopback (localhost), usado para comunicação interna no próprio dispositivo.
- **scope host:** Indica que o endereço é local ao host.
- **lo:** Interface de loopback.

2. **inet6 ::1/128 scope host proto kernel_lo**

- **::1/128**: Endereço IPv6 de loopback, equivalente ao 127.0.0.1 no IPv4.
- **scope host**: Indica que o endereço é local ao host.
- **proto kernel_lo**: Configurado pelo kernel para a interface de loopback.

3. **inet 10.10.50.6/24 brd 10.10.50.255 scope global eth0**

- **10.10.50.6/24**: Endereço IPv4 atribuído à interface eth0 com máscara de sub-rede /24 (255.255.255.0).
- **brd 10.10.50.255**: Endereço de broadcast para a sub-rede.
- **scope global**: Indica que o endereço é acessível globalmente na rede.

4. **inet 10.10.30.2/24 brd 10.10.30.255 scope global eth1**

- **10.10.30.2/24**: Endereço IPv4 atribuído à interface eth1 com máscara de sub-rede /24.
- **brd 10.10.30.255**: Endereço de broadcast para a sub-rede.
- **scope global**: Indica que o endereço é acessível globalmente na rede.

5. **inet 10.10.10.2/24 brd 10.10.10.255 scope global eth2**

- **10.10.10.2/24**: Endereço IPv4 atribuído à interface eth2 com máscara de sub-rede /24.
- **brd 10.10.10.255**: Endereço de broadcast para a sub-rede.
- **scope global**: Indica que o endereço é acessível globalmente na rede.

O comando exibe os endereços IP (IPv4 e IPv6) configurados nas interfaces de rede do sistema, junto com informações como máscara de sub-rede, escopo (local ou global) e endereço de broadcast.

Testar se tem conectividade com as redes:

ping -c 3 10.10.10.1 # corp_net

ping -c 3 10.10.30.1 # infra_net

ping -c 3 10.10.50.1 # guest_net

OBS: Ajustei os IPs no roteiro do arquivo “ANOTACAO-ULTIMO-SCAN.TXT”, que apresentava uma troca entre os IPs das redes guest_net e infra_net. Após uma análise mais detalhada, identifiquei que os laptops possuíam características de dispositivos visitantes, utilizando IPs dinâmicos atribuídos via DHCP.


```
(root@c2d8bea99029)~[/home/analyst]
# ping -c 3 10.10.10.1 # corp_net
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=0.109 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=0.064 ms

--- 10.10.10.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.050/0.074/0.109/0.025 ms

(root@c2d8bea99029)~[/home/analyst]
# ping -c 3 10.10.30.1 # infra_net
PING 10.10.30.1 (10.10.30.1) 56(84) bytes of data.
64 bytes from 10.10.30.1: icmp_seq=1 ttl=64 time=0.096 ms
64 bytes from 10.10.30.1: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 10.10.30.1: icmp_seq=3 ttl=64 time=0.051 ms

--- 10.10.30.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2049ms
rtt min/avg/max/mdev = 0.051/0.066/0.096/0.021 ms

(root@c2d8bea99029)~[/home/analyst]
# ping -c 3 10.10.50.1 # guest_net
PING 10.10.50.1 (10.10.50.1) 56(84) bytes of data.
64 bytes from 10.10.50.1: icmp_seq=1 ttl=64 time=0.077 ms
64 bytes from 10.10.50.1: icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from 10.10.50.1: icmp_seq=3 ttl=64 time=0.049 ms

--- 10.10.50.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2068ms
rtt min/avg/max/mdev = 0.047/0.057/0.077/0.013 ms
```

Figura 3: Print 03 - Comando `ping -c 3 10.10.10.1 # corp_net` | `ping -c 3 10.10.30.1 # infra_net` | `ping -c 3 10.10.50.1 # guest_net`.

O comando ping -c 3 foi usado para testar a conectividade com três diferentes endereços IP, cada um representando uma rede específica. Aqui está a explicação detalhada:

Comando:

- **ping -c 3 <IP>**: Envia 3 pacotes ICMP (Internet Control Message Protocol) para o endereço IP especificado e exibe os resultados.
 - **-c 3**: Define o número de pacotes ICMP a serem enviados (neste caso, 3).
 - **<IP>**: O endereço IP do destino.

Saída Explicada:

1. Rede: corp_net (10.10.10.1)

```Resultado

PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.

64 bytes from 10.10.10.1: icmp\_seq=1 ttl=64 time=0.109 ms

64 bytes from 10.10.10.1: icmp\_seq=2 ttl=64 time=0.050 ms

64 bytes from 10.10.10.1: icmp\_seq=3 ttl=64 time=0.064 ms

```

- **Resposta ICMP:** O host 10.10.10.1 respondeu a todos os 3 pacotes.
- **ttl=64:** Indica o "Time to Live" (tempo de vida) do pacote, que é decrementado a cada roteador atravessado.
- **time=0.050 ms:** Tempo de ida e volta (round-trip time) do pacote em milissegundos.

Estatísticas:

3 packets transmitted, 3 received, 0% packet loss, time 2040ms

rtt min/avg/max/mdev = 0.050/0.074/0.109/0.025 ms

- **Pacotes transmitidos/recebidos:** Todos os 3 pacotes foram enviados e recebidos com sucesso (0% de perda).
- **rtt min/avg/max/mdev:** Estatísticas do tempo de resposta:
 - **min:** Tempo mínimo (0.047 ms).
 - **avg:** Tempo médio (0.051 ms).
 - **max:** Tempo máximo (0.058 ms).
 - **mdev:** Desvio padrão (0.004 ms).

2. Rede: infra_net (10.10.30.1)

```Resultado

PING 10.10.30.1 (10.10.30.1) 56(84) bytes of data.

64 bytes from 10.10.30.1: icmp\_seq=1 ttl=64 time=0.096 ms

64 bytes from 10.10.30.1: icmp\_seq=2 ttl=64 time=0.051 ms

64 bytes from 10.10.30.1: icmp\_seq=3 ttl=64 time=0.051 ms

```

- **Resposta ICMP:** O host 10.10.30.1 respondeu a todos os 3 pacotes.
- **time=0.055 ms:** Tempo de ida e volta dos pacotes.

Estatísticas:

3 packets transmitted, 3 received, 0% packet loss, time 2049ms

rtt min/avg/max/mdev = 0.051/0.066/0.096/0.021ms

- **Pacotes transmitidos/recebidos:** Todos os pacotes foram enviados e recebidos com sucesso (0% de perda).
- **rtt:**
 - **min:** 0.051 ms.
 - **avg:** 0.066 ms.
 - **max:** 0.096 ms.
 - **mdev:** 0.021 ms.

3. Rede: guest_net (10.10.50.1)

```Resultado

PING 10.10.50.1 (10.10.50.1) 56(84) bytes of data.

64 bytes from 10.10.50.1: icmp\_seq=1 ttl=64 time=0.077 ms

64 bytes from 10.10.50.1: icmp\_seq=2 ttl=64 time=0.047 ms

64 bytes from 10.10.50.1: icmp\_seq=3 ttl=64 time=0.049 ms

```

- **Resposta ICMP:** O host 10.10.50.1 respondeu a todos os 3 pacotes.
- **time=0.050 ms:** Tempo de ida e volta dos pacotes.

Estatísticas:

3 packets transmitted, 3 received, 0% packet loss, time 2068ms

rtt min/avg/max/mdev = 0.047/0.057/0.077/0.013 ms

- **Pacotes transmitidos/recebidos:** Todos os pacotes foram enviados e recebidos com sucesso (0% de perda).
- **rtt:**
 - **min:** 0.047 ms.
 - **avg:** 0.057 ms.
 - **max:** 0.077 ms.
 - **mdev:** 0.013 ms.

Em resumo ao iniciando com o reconhecimento da rede corporativa, apliquei uma metodologia de scan profissional em duas fases para mapear a infraestrutura de uma empresa fictícia, utilizando ferramentas adequadas para cada tarefa. Baseando-me na documentação do último analista, utilizei os comandos `ip a` e `ip a | grep inet` para coletar informações detalhadas das interfaces de rede, como endereços IP, máscaras de sub-rede e estado das conexões. Após organizar os dados em uma tabela descritiva, testei a conectividade com as redes `corp_net`, `guest_net` e `infra_net` usando o comando `ping -c 3` para cada endereço IP correspondente. Os resultados mostraram que todos os hosts responderam sem perda de pacotes, com tempos de resposta muito baixos, indicando proximidade na rede e conectividade estável.

Descoberta de Hosts

Vou usar o `nmap` para identificar rapidamente quais hosts (IPs) estão ativos em cada segmento de rede.

Descobrir os hosts com Nmap ping scan:

`` `Comandos - Corp Network - Sub-rede: 10.10.10.0/24

```
nmap -sn -T4 10.10.10.0/24 -oG - | grep "Up"
```

```
nmap -sn -T4 10.10.10.0/24 -oG - | awk '/Up$/{print $2}' | tee corp_net_ips.txt
```

```
nmap -sn -T4 10.10.10.0/24 -oG - | awk '/Up$/{print $2, $3}' | tee corp_net_ips_hosts.txt
```

`` `

```
(root@788e6630118f) [/home/analyst]
# nmap -sn -T4 10.10.10.0/24 -oG - | grep "Up"
Host: 10.10.10.1 ( ) Status: Up
Host: 10.10.10.10 (WS_001.projeto_final_opcao_1_corp_net) Status: Up
Host: 10.10.10.101 (WS_002.projeto_final_opcao_1_corp_net) Status: Up
Host: 10.10.10.127 (WS_003.projeto_final_opcao_1_corp_net) Status: Up
Host: 10.10.10.222 (WS_004.projeto_final_opcao_1_corp_net) Status: Up
Host: 10.10.10.2 (788e6630118f) Status: Up

(root@788e6630118f) [/home/analyst]
# nmap -sn -T4 10.10.10.0/24 -oG - | awk '/Up$/ {print $2}' | tee corp_net_ips.txt
10.10.10.1
10.10.10.10
10.10.10.101
10.10.10.127
10.10.10.222
10.10.10.2

(root@788e6630118f) [/home/analyst]
# nmap -sn -T4 10.10.10.0/24 -oG - | awk '/Up$/ {print $2, $3}' | tee corp_net_ips_hosts.txt
10.10.10.1 ( )
10.10.10.10 (WS_001.projeto_final_opcao_1_corp_net)
10.10.10.101 (WS_002.projeto_final_opcao_1_corp_net)
10.10.10.127 (WS_003.projeto_final_opcao_1_corp_net)
10.10.10.222 (WS_004.projeto_final_opcao_1_corp_net)
10.10.10.2 (788e6630118f)
```

Figura 4: Print 04 - Comandos - Corp Network - Sub-rede: 10.10.10.0/24.

``` Comandos – Infra Network - Sub-rede: 10.10.30.0/24

nmap -sn -T4 10.10.30.0/24 -oG - | grep "Up"

nmap -sn -T4 10.10.30.0/24 -oG - | awk '/Up\$/ {print \$2}' | tee infra\_net\_ips.txt

nmap -sn -T4 10.10.30.0/24 -oG - | awk '/Up\$/ {print \$2, \$3}' | tee infra\_net\_ips\_hosts.txt

```

```
(root@788e6630118f) [/home/analyst]
# nmap -sn -T4 10.10.30.0/24 -oG - | grep "Up"
Host: 10.10.30.1 ( ) Status: Up
Host: 10.10.30.10 (ftp-server.projeto_final_opcao_1_infra_net) Status: Up
Host: 10.10.30.11 (mysql-server.projeto_final_opcao_1_infra_net) Status: Up
Host: 10.10.30.15 (samba-server.projeto_final_opcao_1_infra_net) Status: Up
Host: 10.10.30.17 (openldap.projeto_final_opcao_1_infra_net) Status: Up
Host: 10.10.30.117 (zabbix-server.projeto_final_opcao_1_infra_net) Status: Up
Host: 10.10.30.227 (legacy-server.projeto_final_opcao_1_infra_net) Status: Up
Host: 10.10.30.2 (788e6630118f) Status: Up

(root@788e6630118f) [/home/analyst]
# nmap -sn -T4 10.10.30.0/24 -oG - | awk '/Up$/ {print $2}' | tee infra_net_ips.txt
10.10.30.1
10.10.30.10
10.10.30.11
10.10.30.15
10.10.30.17
10.10.30.117
10.10.30.227
10.10.30.2

(root@788e6630118f) [/home/analyst]
# nmap -sn -T4 10.10.30.0/24 -oG - | awk '/Up$/ {print $2, $3}' | tee infra_net_ips_hosts.txt
10.10.30.1 ( )
10.10.30.10 (ftp-server.projeto_final_opcao_1_infra_net)
10.10.30.11 (mysql-server.projeto_final_opcao_1_infra_net)
10.10.30.15 (samba-server.projeto_final_opcao_1_infra_net)
10.10.30.17 (openldap.projeto_final_opcao_1_infra_net)
10.10.30.117 (zabbix-server.projeto_final_opcao_1_infra_net)
10.10.30.227 (legacy-server.projeto_final_opcao_1_infra_net)
10.10.30.2 (788e6630118f)
```

Figura 5: Print 05 - Comandos – Infra Network - Sub-rede: 10.10.30.0/24

`` Comandos - Guest Network - Sub-rede: 10.10.50.0/24

```
nmap -sn -T4 10.10.50.0/24 -oG - | grep "Up"
```

```
nmap -sn -T4 10.10.50.0/24 -oG - | awk '/Up$/ {print $2}' | tee guest_net_ips.txt
```

```
nmap -sn -T4 10.10.50.0/24 -oG - | awk '/Up$/ {print $2, $3}' | tee guest_net_ips_hosts.txt
```

...

```
(root@788e6630118f) - [/home/analyst]
# nmap -sn -T4 10.10.50.0/24 -oG - | grep "Up"
Host: 10.10.50.1 () Status: Up
Host: 10.10.50.2 (laptop-vastro.projeto_final_opcao_1_guest_net) Status: Up
Host: 10.10.50.3 (laptop-luiz.projeto_final_opcao_1_guest_net) Status: Up
Host: 10.10.50.4 (macbook-aline.projeto_final_opcao_1_guest_net) Status: Up
Host: 10.10.50.5 (notebook-carlos.projeto_final_opcao_1_guest_net) Status: Up
Host: 10.10.50.6 (788e6630118f) Status: Up

(root@788e6630118f) - [/home/analyst]
# nmap -sn -T4 10.10.50.0/24 -oG - | awk '/Up$/ {print $2}' | tee guest_net_ips.txt
10.10.50.1
10.10.50.2
10.10.50.3
10.10.50.4
10.10.50.5
10.10.50.6

(root@788e6630118f) - [/home/analyst]
# nmap -sn -T4 10.10.50.0/24 -oG - | awk '/Up$/ {print $2, $3}' | tee guest_net_ips_hosts.txt
10.10.50.1 ()
10.10.50.2 (laptop-vastro.projeto_final_opcao_1_guest_net)
10.10.50.3 (laptop-luiz.projeto_final_opcao_1_guest_net)
10.10.50.4 (macbook-aline.projeto_final_opcao_1_guest_net)
10.10.50.5 (notebook-carlos.projeto_final_opcao_1_guest_net)
10.10.50.6 (788e6630118f)
```

Figura 6: Print 06 - Comandos - Guest Network - Sub-rede: 10.10.50.0/24.

O primeiro comando lista os hosts ativos na sub-rede diretamente no terminal.

```
`nmap -sn -T4 {ENDEREÇO-BASE}/{MÁSCARA-SUB-REDE} -oG - | grep "Up"`
```

O segundo comando salva apenas os endereços IP dos hosts ativos em um arquivo.

```
`nmap -sn -T4 {ENDEREÇO-BASE}/{MÁSCARA-SUB-REDE} -oG - | awk '/Up$/ {print $2}' | {NOME}_net_ips.txt`
```

O terceiro comando salva os endereços IP e os nomes de host dos hosts ativos em um arquivo.

```
`nmap -sn -T4 {ENDEREÇO-BASE}/{MÁSCARA-SUB-REDE} -oG - | awk '/Up$/ {print $2, $3}' | {NOME}_net_ips_hosts.txt`
```

Esses comandos são úteis para mapear rapidamente os dispositivos ativos em uma rede e organizar as informações em arquivos para análise posterior.

1. Comando Base:

```
`nmap -sn -T4 10.10.10.0/24 -oG - | grep "Up" `
```

- **nmap -sn**: Realiza um "ping scan" para identificar hosts ativos sem realizar uma varredura de portas.
- **-T4**: Define a velocidade da varredura como "agressiva" (mais rápida).
- **10.10.10.0/24**: Especifica o intervalo de IPs a ser escaneado (sub-rede /24, ou seja, 256 endereços).
- **-oG -**: Gera a saída no formato "grepable" e a envia para a saída padrão (stdout).
- **| grep "Up"**: Filtra apenas os hosts que estão ativos (com status "Up").

Saída: Lista os hosts ativos na sub-rede, incluindo seus endereços IP e nomes de host (se disponíveis).

2. Extraíndo Apenas os IPs:

```
nmap -sn -T4 10.10.10.0/24 -oG - | awk '/Up$/ {print $2}' | tee corp_net_ips.txt
```

- **awk '/Up\$/ {print \$2}'**: Filtra as linhas com "Up" no final e imprime apenas o segundo campo (o endereço IP).
- **tee corp_net_ips.txt**: Salva a lista de IPs ativos no arquivo corp_net_ips.txt e também exibe no terminal.

Saída: Apenas os endereços IP dos hosts ativos.

3. Extraíndo IPs e Nomes de Host:

```
nmap -sn -T4 10.10.10.0/24 -  
oG - | awk '/Up$/ {print $2, $3}' | tee corp_net_ips_hosts.txt
```

- **awk '/Up\$/ {print \$2, \$3}'**: Filtra as linhas com "Up" no final e imprime o segundo campo (IP) e o terceiro campo (nome do host, se disponível).
- **tee corp_net_ips_hosts.txt**: Salva a lista de IPs e nomes de host no arquivo corp_net_ips_hosts.txt e também exibe no terminal.

Saída: Lista de IPs e seus respectivos nomes de host (ou () se o nome não estiver disponível).

4. Repetição para Outras Sub-redes:

Os mesmos comandos foram aplicados para as sub-redes 10.10.30.0/24 (infra_net) e 10.10.50.0/24 (guest_net), gerando arquivos separados para cada rede:

- **IPs ativos:** Salvos em infra_net_ips.txt e guest_net_ips.txt.
- **IPs e nomes de host:** Salvos em infra_net_ips_hosts.txt e guest_net_ips_hosts.txt.

Depois executei o comando `ls` para ver se gravou os arquivos.

```
(root@788e6630118f) - [/home/analyst]
# ls
corp_net_ips.txt      guest_net_ips.txt      infra_net_ips.txt
corp_net_ips_hosts.txt  guest_net_ips_hosts.txt  infra_net_ips_hosts.txt
```

Figura 7: Print 07 - Comando `ls`.

Aqui está uma tabela descritiva com os resultados dos comandos nmap para as três sub-redes:

Sub-rede	Endereço IP	Nome do Host
corp_net	10.10.10.1	Switch: Corp Network
	10.10.10.10	WS_001.projeto_final_opcao_1_corp_net
	10.10.10.101	WS_002.projeto_final_opcao_1_corp_net
	10.10.10.127	WS_003.projeto_final_opcao_1_corp_net
	10.10.10.222	WS_004.projeto_final_opcao_1_corp_net
	10.10.10.2	Equipamento local (analyst)
infra_net	10.10.30.1	Switch: Infra Network
	10.10.30.10	ftp-server.projeto_final_opcao_1_infra_net
	10.10.30.11	mysql-server.projeto_final_opcao_1_infra_net
	10.10.30.15	samba-server.projeto_final_opcao_1_infra_net
	10.10.30.17	openldap.projeto_final_opcao_1_infra_net
	10.10.30.117	zabbix-server.projeto_final_opcao_1_infra_net
	10.10.30.227	legacy-server.projeto_final_opcao_1_infra_net
	10.10.30.2	Equipamento local (analyst)
guest_net	10.10.50.1	Switch: Guest Network
	10.10.50.2	laptop-vastro.projeto_final_opcao_1_guest_net
	10.10.50.3	laptop-luiz.projeto_final_opcao_1_guest_net
	10.10.50.4	macbook-aline.projeto_final_opcao_1_guest_net
	10.10.50.5	notebook-carlos.projeto_final_opcao_1_guest_net
	10.10.50.6	Equipamento local (analyst)

Detalhes da Tabela:

- **Sub-rede:** Identifica a rede escaneada (corp_net, infra_net, guest_net).
- **Endereço IP:** Lista os IPs dos hosts ativos na sub-rede.
- **Nome do Host:** Nome do dispositivo associado ao IP (se disponível).
- **Status:** Indica se o host está ativo (Up).

Essa tabela organiza os resultados de forma clara e facilita a análise das redes escaneadas.

Resumo Geral:

Os comandos utilizam o nmap para identificar hosts ativos em três sub-redes diferentes (corp_net, infra_net, guest_net). A saída é processada com grep e awk para extrair informações específicas (IPs e nomes de host) e salvar os resultados em arquivos organizados. Isso facilita a análise e documentação da infraestrutura de rede.

Scan de Portas

Após identificar os hosts ativos, utilizarei o Rustscan para realizar um escaneamento rápido e eficiente das portas abertas em cada máquina, permitindo uma análise detalhada dos serviços disponíveis.

Scan rápido com Rustscan para pegar as portas abertas

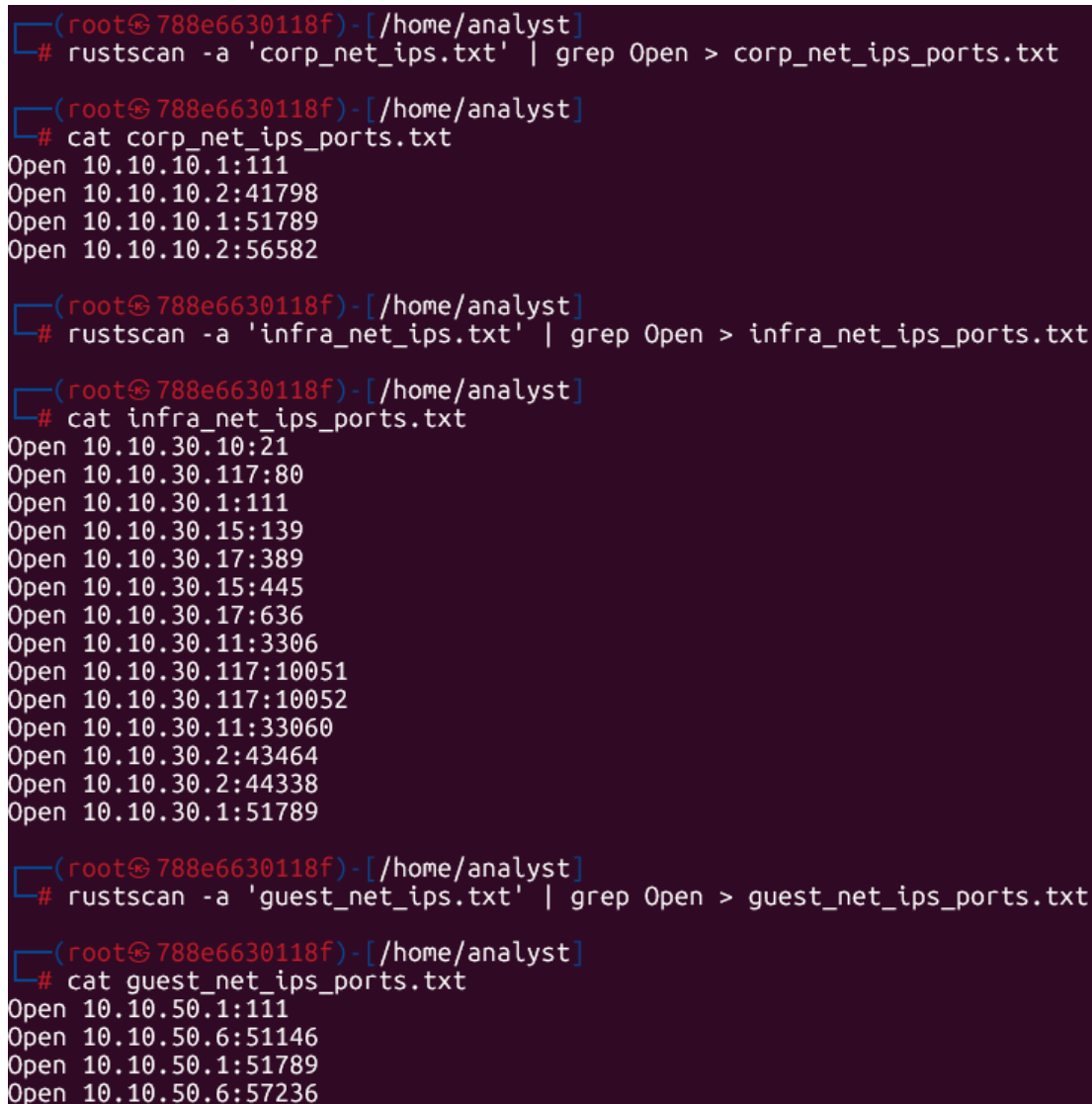
`` Comandos – Rustscan

```
rustscan -a 'corp_net_ips.txt' | grep Open > corp_net_ips_ports.txt
```

```
rustscan -a 'infra_net_ips.txt' | grep Open > infra_net_ips_ports.txt
```

```
rustscan -a 'guest_net_ips.txt' | grep Open > guest_net_ips_ports.txt
```

``



```
(root@788e6630118f) - [/home/analyst]
# rustscan -a 'corp_net_ips.txt' | grep Open > corp_net_ips_ports.txt

(root@788e6630118f) - [/home/analyst]
# cat corp_net_ips_ports.txt
Open 10.10.10.1:111
Open 10.10.10.2:41798
Open 10.10.10.1:51789
Open 10.10.10.2:56582

(root@788e6630118f) - [/home/analyst]
# rustscan -a 'infra_net_ips.txt' | grep Open > infra_net_ips_ports.txt

(root@788e6630118f) - [/home/analyst]
# cat infra_net_ips_ports.txt
Open 10.10.30.10:21
Open 10.10.30.117:80
Open 10.10.30.1:111
Open 10.10.30.15:139
Open 10.10.30.17:389
Open 10.10.30.15:445
Open 10.10.30.17:636
Open 10.10.30.11:3306
Open 10.10.30.117:10051
Open 10.10.30.117:10052
Open 10.10.30.11:33060
Open 10.10.30.2:43464
Open 10.10.30.2:44338
Open 10.10.30.1:51789

(root@788e6630118f) - [/home/analyst]
# rustscan -a 'guest_net_ips.txt' | grep Open > guest_net_ips_ports.txt

(root@788e6630118f) - [/home/analyst]
# cat guest_net_ips_ports.txt
Open 10.10.50.1:111
Open 10.10.50.6:51146
Open 10.10.50.1:51789
Open 10.10.50.6:57236
```

Figura 8: Print 08 - Comandos – Rustscan.

Saída Explicada:

1. **rustscan -a 'corp_net_ips.txt' | grep Open > corp_net_ips_ports.txt**
 - **rustscan -a 'corp_net_ips.txt':** Executa o rustscan para escanear as portas dos IPs listados no arquivo corp_net_ips.txt.
 - **| grep Open:** Filtra a saída para mostrar apenas as linhas que contêm "Open", ou seja, as portas abertas.
 - **> corp_net_ips_ports.txt:** Redireciona a saída filtrada para o arquivo corp_net_ips_ports.txt.
2. **cat corp_net_ips_ports.txt**
 - Exibe o conteúdo do arquivo corp_net_ips_ports.txt, que contém os IPs e as portas abertas identificadas na rede corp_net.
3. **rustscan -a 'infra_net_ips.txt' | grep Open > infra_net_ips_ports.txt**
 - Realiza o mesmo processo descrito acima, mas para os IPs listados no arquivo infra_net_ips.txt, salvando os resultados no arquivo infra_net_ips_ports.txt.
4. **cat infra_net_ips_ports.txt**
 - Exibe o conteúdo do arquivo infra_net_ips_ports.txt, que contém os IPs e as portas abertas identificadas na rede infra_net.
5. **rustscan -a 'guest_net_ips.txt' | grep Open > guest_net_ips_ports.txt**
 - Realiza o mesmo processo descrito acima, mas para os IPs listados no arquivo guest_net_ips.txt, salvando os resultados no arquivo guest_net_ips_ports.txt.
6. **cat guest_net_ips_ports.txt**
 - Exibe o conteúdo do arquivo guest_net_ips_ports.txt, que contém os IPs e as portas abertas identificadas na rede guest_net.

Tabela Descritiva dos Resultados:

Rede	Endereço IP	Portas Abertas	Descrição
corp_net	10.10.10.1	111, 51789	RPC e porta aleatória
	10.10.10.2	41798, 56582	Portas aleatórias
infra_net	10.10.30.10	21	FTP
	10.10.30.117	80, 10051, 10052	HTTP, Zabbix
	10.10.30.1	111, 51789	RPC e porta aleatória
	10.10.30.15	139, 445	SMB
	10.10.30.17	389, 636	LDAP e LDAPS
	10.10.30.11	3306, 33060	MySQL
	10.10.30.2	43464, 44338	Portas aleatórias
guest_net	10.10.50.1	111, 51789	RPC e porta aleatória
	10.10.50.6	51146, 57236	Portas aleatórias

Observações:

- **Portas Abertas:** As portas listadas indicam serviços ou aplicações que estão acessíveis na rede.
- **Descrição:** Algumas portas conhecidas foram identificadas (ex.: 21 para FTP, 80 para HTTP, 139/445 para SMB, etc.), enquanto outras são portas aleatórias que podem estar associadas a serviços específicos ou temporários.
- **Riscos Potenciais:** Portas abertas podem representar vulnerabilidades se os serviços não estiverem devidamente configurados ou protegidos. É importante investigar os serviços associados a essas portas e verificar se há necessidade de mantê-las abertas.

Por fim, vou analisar os resultados para entender a topologia e os serviços expostos.

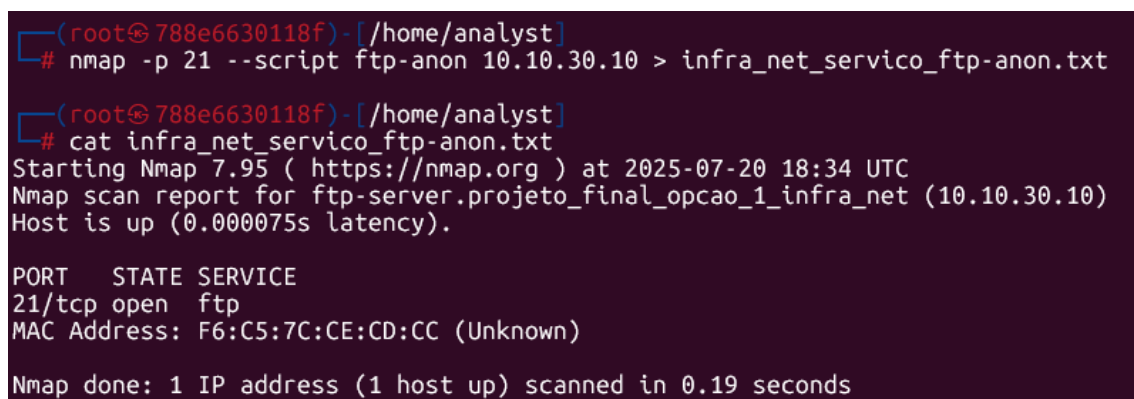
Analisar os serviços específicos

FTP

É um protocolo utilizado para transferir arquivos entre computadores em uma rede, permitindo operações como envio, recebimento e gerenciamento de dados. Ele funciona em um modelo cliente-servidor, onde o cliente acessa arquivos armazenados no servidor. Apesar de ser uma solução prática e amplamente adotada, o FTP padrão transmite informações sem criptografia, o que pode representar riscos de segurança. Por isso, é recomendável utilizar versões mais seguras, como FTPS ou SFTP, que garantem maior proteção durante a transferência de dados.

Comando:

```
` nmap -p 21 --script ftp-anon 10.10.30.10 > infra_net_servico_ftp-anon.txt `
```



```
(root@788e6630118f) - [/home/analyst]
# nmap -p 21 --script ftp-anon 10.10.30.10 > infra_net_servico_ftp-anon.txt

(root@788e6630118f) - [/home/analyst]
# cat infra_net_servico_ftp-anon.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-20 18:34 UTC
Nmap scan report for ftp-server.projeto_final_opcao_1_infra_net (10.10.30.10)
Host is up (0.000075s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
MAC Address: F6:C5:7C:CE:CD:CC (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Figura 9: Print 09 - Comandos `` nmap -p 21 --script ftp-anon 10.10.30.10 > infra_net_servico_ftp-anon.txt ``.

Explicação do Comando:

Partes do Comando:

1. **nmap**: Ferramenta de varredura de rede usada para descobrir hosts e serviços.
2. **-p 21**: Especifica que o Nmap deve escanear apenas a porta 21 (FTP).
3. **--script ftp-anon**: Utiliza o script ftp-anon do Nmap para verificar se o servidor FTP permite acesso anônimo. O script tenta autenticar no servidor FTP sem credenciais (usuário "anonymous").
4. **10.10.30.10**: Endereço IP do host alvo (neste caso, o servidor FTP).
5. **> infra_net_servico_ftp-anon.txt**: Redireciona a saída do comando para o arquivo infra_net_servico_ftp-anon.txt.

Saída do Arquivo `infra_net_servico_ftp-anon.txt`:

...

Starting Nmap 7.95 (<https://nmap.org>) at 2025-07-20 18:34 UTC

Nmap scan report for ftp-server.projeto_final_opcao_1_infra_net (10.10.30.10)

Host is up (0.000075s latency).

PORT STATE SERVICE

21/tcp open ftp

MAC Address: F6:C5:7C:CE:CD:CC (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds

...

Análise da Saída:

1. **Host is up:** O host 10.10.30.10 está ativo e respondeu à varredura.
2. **PORT STATE SERVICE:**
 - **21/tcp open ftp:** A porta 21 (protocolo FTP) está aberta e o serviço FTP está ativo.
3. **MAC Address:** O endereço MAC do dispositivo foi identificado como F6:C5:7C:CE:CD:CC, mas o fabricante não foi reconhecido.
4. **Tempo de Execução:** A varredura foi concluída em 0.19 segundos.

Interpretação:

- O servidor FTP no IP 10.10.30.10 está ativo e com a porta 21 aberta.
- O script ftp-anon foi executado, mas a saída não indica explicitamente se o acesso anônimo foi permitido ou negado. Isso pode significar que o servidor não permite acesso anônimo ou que o script não conseguiu determinar isso.

Próximos Passos:

1. **Verificar Acesso Anônimo Manualmente:**
 - Use um cliente FTP para tentar acessar o servidor com o usuário anonymous: ``ftp 10.10.30.10``
 - Se o acesso for permitido, isso pode representar um risco de segurança.
 -

2. Analisar Configuração do Servidor FTP:

- Verifique se o servidor está configurado para permitir acesso anônimo e, se sim, avalie os arquivos disponíveis.

3. Documentar no Relatório:

- Inclua no relatório técnico:
 - O IP do servidor (10.10.30.10).
 - O serviço identificado (FTP).
 - Se o acesso anônimo foi permitido ou não.
 - Riscos associados ao serviço FTP (ex.: exposição de arquivos sensíveis).

4. Tabela Descritiva:

IP	Porta	Serviço	Estado	MAC Address	Acesso Anônimo	Risco
10.10.30.10	21	FTP	Aberto	F6:C5:7C:CE:CD:CC	Não identificado (Erro mkdb)	Possível exposição de arquivos via FTP

Verificar Acesso Anônimo Manualmente - Instalação do cliente FTP:

Instalei um cliente FTP para teste utilizando o comando ``apt update && apt install ftp -y``.

```
(root@2be5edf97264) - [/home/analyst]
# apt update && apt install ftp -y
Hit:1 http://http.kali.org/kali kali-rolling InRelease
7 packages can be upgraded. Run 'apt list --upgradable' to see them.
Installing:
ftp

Installing dependencies:
tftp
```

Figura 10: Print 10 - Comando ``apt update && apt install ftp -y``.

- **apt update:** Atualiza a lista de pacotes disponíveis no sistema.
- **apt install ftp -y:** Instala o cliente FTP no sistema sem solicitar confirmação.

Saída:

- O cliente FTP foi instalado com sucesso, junto com a dependência tftp.

Comando:

Após a instalação, tentei acessar o servidor FTP com o comando `ftp 10.10.30.10`. No entanto, o acesso falhou devido à falta de configuração no servidor, apresentando o erro: "Unable to read the indexed puredb file (or old format detected) - Try pure-pw mkdb". Isso indica que o banco de dados de usuários do servidor FTP não foi corretamente configurado.

```
(root@2be5edf97264) - [ /home/analyst ]
# ftp 10.10.30.10
Connected to 10.10.30.10.
220----- Welcome to Pure-FTPd [privsep] [TLS] -----
220-You are user number 1 of 5 allowed.
220-Local time is now 20:41. Server port: 21.
220-This is a private system - No anonymous login
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
Name (10.10.30.10:root):
331 User root OK. Password required
Password:
421 Unable to read the indexed puredb file (or old format detected) - Try pure-pw mkdb
ftp: Login failed
ftp>
```

Figura 11: Print 11 - Comando `ftp 10.10.30.10`.

Saída:

- O servidor FTP responde com uma mensagem de boas-vindas.
- O servidor não permite logins anônimos (No anonymous login).
- O usuário tenta fazer login como root, mas a autenticação falha devido ao erro: 421 Unable to read the indexed puredb file (or old format detected) - Try pure-pw mkdb

Esse erro indica que o servidor FTP está configurado para usar um banco de dados de usuários (puredb), mas o arquivo necessário está ausente ou corrompido.

Tabela Descritiva:

Comando	Descrição	Resultado
<code>nmap -p 21 --script ftp-anon 10.10.30.10</code>	Escaneia a porta 21 do servidor FTP e verifica se o acesso anônimo é permitido.	Porta 21 aberta, mas sem informações sobre acesso anônimo.
<code>apt update && apt install ftp -y</code>	Instala o cliente FTP no sistema.	Cliente FTP instalado com sucesso.
<code>ftp 10.10.30.10</code>	Conecta ao servidor FTP no IP 10.10.30.10.	Conexão estabelecida, mas login falhou devido a erro no banco de dados do servidor (puredb).

Conclusão:

- O servidor FTP está ativo, mas não permite logins anônimos.

MySQL

É um sistema de gerenciamento de banco de dados relacional amplamente utilizado em aplicações corporativas e web. Ele permite armazenar, organizar e acessar grandes volumes de dados de forma eficiente, utilizando a linguagem SQL (Structured Query Language) para manipulação e consulta. Reconhecido por sua robustez, escalabilidade e desempenho, o MySQL é uma escolha popular para gerenciar informações em sistemas de e-commerce, plataformas de conteúdo e aplicações empresariais. Apesar de sua eficiência, é essencial garantir boas práticas de segurança, como restrição de acessos, autenticação robusta e uso de conexões criptografadas, para proteger os dados armazenados.

Comando:

```
`nmap -p 3306 --script mysql-info 10.10.30.11 > infra_net_servico_mysql-info.txt`
```

```
(root@788e6630118f)~[/home/analyst]
# nmap -p 3306 --script mysql-info 10.10.30.11 > infra_net_servico_mysql-info.txt

(root@788e6630118f)~[/home/analyst]
# cat infra_net_servico_mysql-info.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-20 19:26 UTC
Nmap scan report for mysql-server.projeto_final_opcao_1_infra_net (10.10.30.11)
Host is up (0.000069s latency).

PORT      STATE SERVICE
3306/tcp  open  mysql
| mysql-info:
|   Protocol: 10
|   Version: 8.0.42
|   Thread ID: 11
|   Capabilities flags: 65535
|   Some Capabilities: Support41Auth, DontAllowDatabaseTableColumn, Speaks41Protocol0
ld, SupportsTransactions, FoundRows, SupportsCompression, LongColumnFlag, LongPasswor
d, ODBCClient, ConnectWithDatabase, IgnoreSigpipes, SwitchToSSLAfterHandshake, Speaks
41ProtocolNew, InteractiveClient, IgnoreSpaceBeforeParenthesis, SupportsLoadDataLocal
, SupportsMultipleStatements, SupportsMultipleResults, SupportsAuthPlugins
|   Status: Autocommit
|   Salt: \x0F*pIV\x1FM\x7F\x1A\x7\x0F\x1A3\x0CF7]\x13
|   Auth Plugin Name: caching_sha2_password
|_  MAC Address: 96:AA:26:A2:CB:9A (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
```

Figura 12: Print 12 - Comando `nmap -p 3306 --script mysql-info 10.10.30.11 > infra_net_servico_mysql-info.txt`.

Explicação do Comando:

Partes do Comando:

1. **nmap**: Ferramenta para varredura de redes.
2. **-p 3306**: Especifica que o Nmap deve escanear apenas a porta 3306, que é a porta padrão do serviço MySQL.
3. **--script mysql-info**: Usa o script mysql-info do Nmap para coletar informações detalhadas sobre o serviço MySQL em execução na porta 3306.

4. **10.10.30.11:** Endereço IP do host alvo (neste caso, o servidor MySQL).
5. **> infra_net_servico_mysql-info.txt:** Redireciona a saída do comando para o arquivo infra_net_servico_mysql-info.txt.

Saída do Arquivo infra_net_servico_mysql-info.txt:

...

Starting Nmap 7.95 (<https://nmap.org>) at 2025-07-20 19:26 UTC

Nmap scan report for mysql-server.projeto_final_opcao_1_infra_net (10.10.30.11)

Host is up (0.000069s latency).

PORT STATE SERVICE

3306/tcp open mysql

| mysql-info:

| Protocol: 10

| Version: 8.0.42

| Thread ID: 11

| Capabilities flags: 65535

| Some Capabilities: Support41Auth, DontAllowDatabaseTableColumn, Speaks41
ProtocolOld, SupportsTransactions, FoundRows, SupportsCompression, LongCol
umnFlag, LongPassword, ODBCClient, ConnectWithDatabase, IgnoreSigpipes, Sw
itchToSSLAfterHandshake, Speaks41ProtocolNew, InteractiveClient, IgnoreSpace
BeforeParenthesis, SupportsLoadDataLocal, SupportsMultipleStatments, Support
sMultipleResults, SupportsAuthPlugins

| Status: Autocommit

| Salt: \x0F*pIV\x1FM\x7F\x1AIX7\x0F\x1A3\x0CF7]\x13

|_ Auth Plugin Name: caching_sha2_password

MAC Address: 96:AA:26:A2:CB:9A (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds

...

Análise da Saída:

1. Host Information:

- **Host is up:** O host 10.10.30.11 está ativo e respondeu à varredura.
- **MAC Address:** O endereço MAC do dispositivo foi identificado como 96:AA:26:A2:CB:9A, mas o fabricante não foi reconhecido.

2. Porta e Serviço:

- **3306/tcp open mysql:** A porta 3306 está aberta e o serviço MySQL está ativo.

3. Informações do MySQL (mysql-info):

- **Protocol: 10:** Versão do protocolo MySQL.
- **Version: 8.0.42:** Versão do servidor MySQL em execução.
- **Thread ID: 11:** ID da thread atual do servidor MySQL.
- **Capabilities flags:** Flags que indicam as capacidades do servidor MySQL. Exemplos:
 - **SupportsTransactions:** Suporte a transações.
 - **SupportsCompression:** Suporte a compressão.
 - **SupportsMultipleStatements:** Suporte a múltiplas instruções em uma única consulta.
 - **SwitchToSSLAfterHandshake:** Suporte a troca para SSL após o handshake.
- **Status: Autocommit:** O servidor está configurado para o modo de autocommit.
- **Salt:** Salt usado para autenticação.
- **Auth Plugin Name: caching_sha2_password:** Plugin de autenticação usado pelo servidor MySQL.

4. Tempo de Execução:

- A varredura foi concluída em 0.16 segundos.

Interpretação:

- O servidor MySQL no IP 10.10.30.11 está ativo e acessível na porta 3306.
- O script mysql-info revelou informações detalhadas sobre a versão do MySQL, o protocolo usado, as capacidades do servidor e o método de autenticação.
- O método de autenticação caching_sha2_password é o padrão em versões recentes do MySQL, oferecendo maior segurança.

Tabela Descritiva:

IP	Porta	Serviço	Versão	Protocolo	Autenticação	Status	MAC Address
10.10.30.11	3306	MySQL	8.0.42	10	caching_sha2_password	Autocommit	96:AA:26:A2:CB:9A

Próximos Passos:

1. Verificar Segurança:

- Certifique-se de que o servidor MySQL está configurado para aceitar conexões apenas de hosts autorizados.
- Avalie se o método de autenticação e as permissões de usuários estão adequados.

2. Documentar no Relatório:

- Inclua as informações coletadas no inventário técnico e no relatório de diagnóstico.
- Identifique possíveis riscos associados ao serviço MySQL exposto na rede.

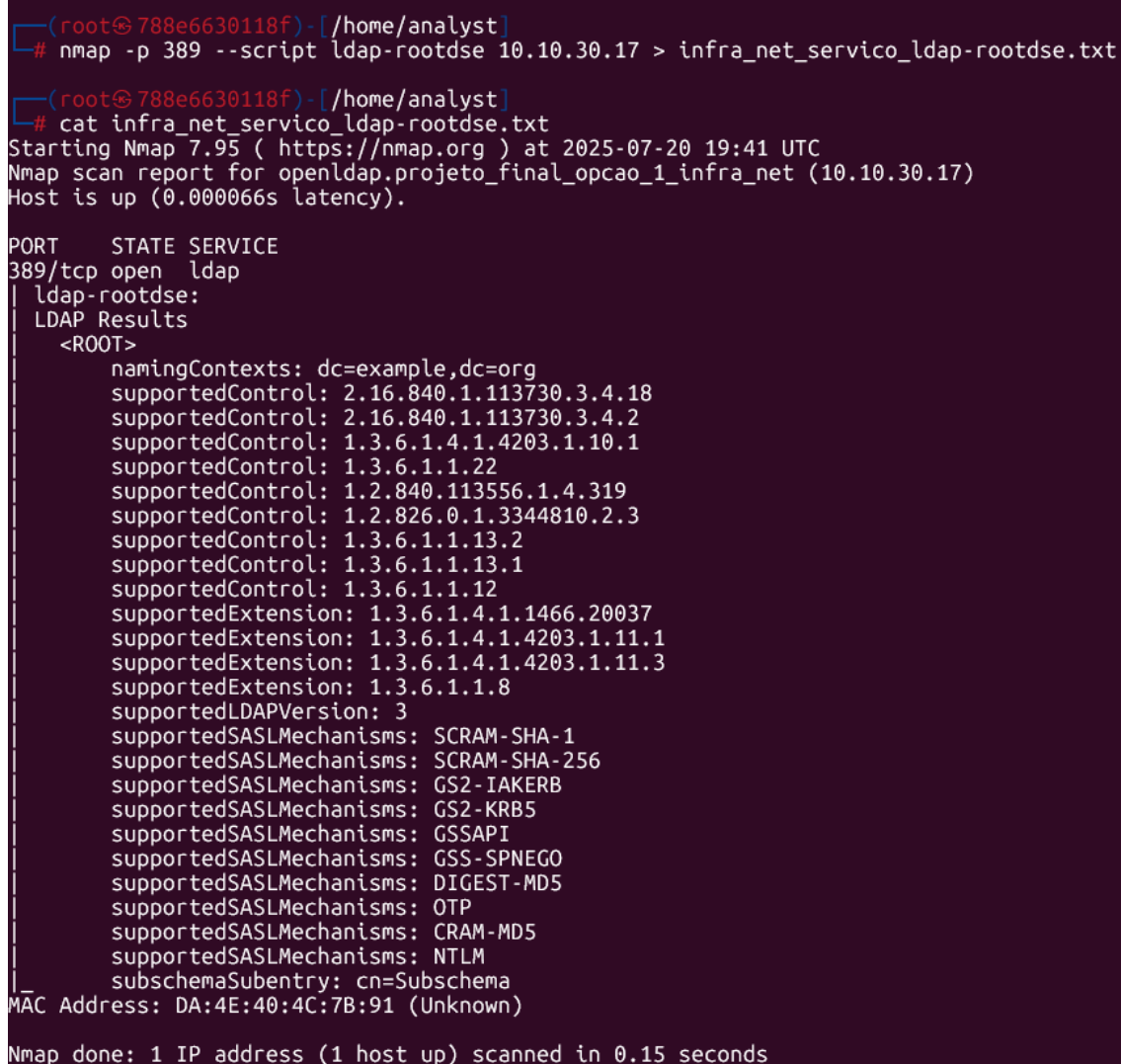
OBS: A partir da versão 8.0 do MySQL, o método de autenticação padrão foi alterado para `caching_sha2_password`, que oferece maior segurança em comparação ao antigo `mysql_native_password`. No entanto, o script `mysql-brute` do Nmap pode não ser compatível com esse novo método de autenticação, resultando em falhas ao tentar realizar ataques de força bruta contra servidores configurados com `caching_sha2_password`.

LDAP

O LDAP (Lightweight Directory Access Protocol) é um protocolo utilizado para acessar e gerenciar serviços de diretórios, permitindo a organização e consulta de informações como usuários, grupos e dispositivos em uma rede. Amplamente adotado em ambientes corporativos, o LDAP facilita a centralização de autenticação e controle de acesso, sendo essencial para sistemas que requerem gerenciamento eficiente de identidades. Sua estrutura hierárquica e flexível torna o LDAP ideal para integrar diferentes aplicações e serviços, mas é fundamental implementar medidas de segurança, como autenticação robusta e criptografia, para proteger os dados sensíveis armazenados.

Comando:

```
`nmap -p 389 --script ldap-rootdse 10.10.30.17 > infra_net_servico_ldap-rootdse.txt`
```



```
(root@788e6630118f) - [/home/analyst]
# nmap -p 389 --script ldap-rootdse 10.10.30.17 > infra_net_servico_ldap-rootdse.txt

(root@788e6630118f) - [/home/analyst]
# cat infra_net_servico_ldap-rootdse.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-20 19:41 UTC
Nmap scan report for openldap.projeto_final_opcao_1_infra_net (10.10.30.17)
Host is up (0.000066s latency).

PORT      STATE SERVICE
389/tcp   open  ldap
| ldap-rootdse:
| LDAP Results
| <ROOT>
|   namingContexts: dc=example,dc=org
|   supportedControl: 2.16.840.1.113730.3.4.18
|   supportedControl: 2.16.840.1.113730.3.4.2
|   supportedControl: 1.3.6.1.4.1.4203.1.10.1
|   supportedControl: 1.3.6.1.1.22
|   supportedControl: 1.2.840.113556.1.4.319
|   supportedControl: 1.2.826.0.1.3344810.2.3
|   supportedControl: 1.3.6.1.1.13.2
|   supportedControl: 1.3.6.1.1.13.1
|   supportedControl: 1.3.6.1.1.12
|   supportedExtension: 1.3.6.1.4.1.1466.20037
|   supportedExtension: 1.3.6.1.4.1.4203.1.11.1
|   supportedExtension: 1.3.6.1.4.1.4203.1.11.3
|   supportedExtension: 1.3.6.1.1.8
|   supportedLDAPVersion: 3
|   supportedSASLMechanisms: SCRAM-SHA-1
|   supportedSASLMechanisms: SCRAM-SHA-256
|   supportedSASLMechanisms: GS2-IAKERB
|   supportedSASLMechanisms: GS2-KRB5
|   supportedSASLMechanisms: GSSAPI
|   supportedSASLMechanisms: GSS-SPNEGO
|   supportedSASLMechanisms: DIGEST-MD5
|   supportedSASLMechanisms: OTP
|   supportedSASLMechanisms: CRAM-MD5
|   supportedSASLMechanisms: NTLM
|   subschemaSubentry: cn=Subschema
|_ MAC Address: DA:4E:40:4C:7B:91 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
```

Figura 13: Print 13 – Comando `nmap -p 389 --script ldap-rootdse 10.10.30.17 > infra_net_servico_ldap-rootdse.txt`.

Explicação do Comando:

Partes do Comando:

1. **nmap**: Ferramenta para varredura de redes.
2. **-p 389**: Especifica que o Nmap deve escanear apenas a porta 389, que é a porta padrão para o serviço LDAP (Lightweight Directory Access Protocol).
3. **--script ldap-rootdse**: Usa o script ldap-rootdse do Nmap para consultar informações do diretório LDAP. Este script coleta informações do **Root DSE (Directory Service Entry)**, que contém dados sobre as capacidades e configurações do servidor LDAP.
4. **10.10.30.17**: Endereço IP do servidor LDAP alvo.
5. **> infra_net_servico_ldap-rootdse.txt**: Redireciona a saída do comando para o arquivo infra_net_servico_ldap-rootdse.txt.

Saída do Arquivo infra_net_servico_ldap-rootdse.txt:

...

Starting Nmap 7.95 (<https://nmap.org>) at 2025-07-20 19:41 UTC

Nmap scan report for openldap.projeto_final_opcao_1_infra_net (10.10.30.17)

Host is up (0.000066s latency).

PORT STATE SERVICE

389/tcp open ldap

| ldap-rootdse:

| LDAP Results

| <ROOT>

| namingContexts: dc=example,dc=org

| supportedControl: 2.16.840.1.113730.3.4.18

| supportedControl: 2.16.840.1.113730.3.4.2

| supportedControl: 1.3.6.1.4.1.4203.1.10.1

| supportedControl: 1.3.6.1.1.22

| supportedControl: 1.2.840.113556.1.4.319

| supportedControl: 1.2.826.0.1.3344810.2.3

| supportedControl: 1.3.6.1.1.13.2

```
| supportedControl: 1.3.6.1.1.13.1
| supportedControl: 1.3.6.1.1.12
| supportedExtension: 1.3.6.1.4.1.1466.20037
| supportedExtension: 1.3.6.1.4.1.4203.1.11.1
| supportedExtension: 1.3.6.1.4.1.4203.1.11.3
| supportedExtension: 1.3.6.1.1.8
| supportedLDAPVersion: 3
| supportedSASLMechanisms: SCRAM-SHA-1
| supportedSASLMechanisms: SCRAM-SHA-256
| supportedSASLMechanisms: GS2-IKRB
| supportedSASLMechanisms: GS2-KRB5
| supportedSASLMechanisms: GSSAPI
| supportedSASLMechanisms: GSS-SPNEGO
| supportedSASLMechanisms: DIGEST-MD5
| supportedSASLMechanisms: OTP
| supportedSASLMechanisms: CRAM-MD5
| supportedSASLMechanisms: NTLM
|_ subschemaSubentry: cn=Subschema
MAC Address: DA:4E:40:4C:7B:91 (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
...
```

Análise da Saída:

1. Host Information:

- **Host is up:** O host 10.10.30.17 está ativo e respondeu à varredura.
- **MAC Address:** O endereço MAC do dispositivo foi identificado como DA:4E:40:4C:7B:91, mas o fabricante não foi reconhecido.

2. Porta e Serviço:

- **389/tcp open ldap:** A porta 389 está aberta e o serviço LDAP está ativo.

3. Informações do LDAP (ldap-rootdse):

- **namingContexts: dc=example,dc=org:** Indica o contexto de nomenclatura principal do diretório LDAP. Neste caso, o domínio é example.org.
- **supportedControl:** Lista os controles LDAP suportados pelo servidor. Cada controle é identificado por um OID (Object Identifier).
- **supportedExtension:** Lista as extensões LDAP suportadas pelo servidor.
- **supportedLDAPVersion: 3:** Indica que o servidor suporta a versão 3 do protocolo LDAP.
- **supportedSASLMechanisms:** Lista os mecanismos de autenticação SASL (Simple Authentication and Security Layer) suportados pelo servidor. Exemplos:
 - **SCRAM-SHA-1 e SCRAM-SHA-256:** Mecanismos de autenticação seguros baseados em hash.
 - **GSSAPI e GSS-SPNEGO:** Usados para autenticação Kerberos.
 - **NTLM:** Mecanismo de autenticação da Microsoft.
- **subschemaSubentry: cn=Subschema:** Indica a entrada de subschema, que contém informações sobre os esquemas LDAP suportados.

4. Tempo de Execução:

- A varredura foi concluída em 0.15 segundos.

Interpretação:

- O servidor LDAP no IP 10.10.30.17 está ativo e acessível na porta 389.
- O script ldap-rootdse revelou informações detalhadas sobre o diretório LDAP, incluindo o domínio principal (dc=example,dc=org), os controles e extensões suportados, a versão do protocolo LDAP e os mecanismos de autenticação disponíveis.
- Essas informações podem ser úteis para entender a configuração do servidor LDAP e identificar possíveis vulnerabilidades ou riscos.

Tabela Descritiva:

IP	Porta	Serviço	Domínio LDAP	Versão LDAP	Autenticação	MAC Address
10.10.30.17	389	LDAP	dc=example,dc=org	3	SCRAM-SHA-1, GSSAPI, NTLM, etc.	DA:4E:40:4C:7B:91

Próximos Passos:**1. Verificar Segurança:**

- Certifique-se de que o servidor LDAP está configurado para aceitar conexões apenas de hosts autorizados.
- Avalie se os mecanismos de autenticação e permissões estão adequados.

2. Documentar no Relatório:

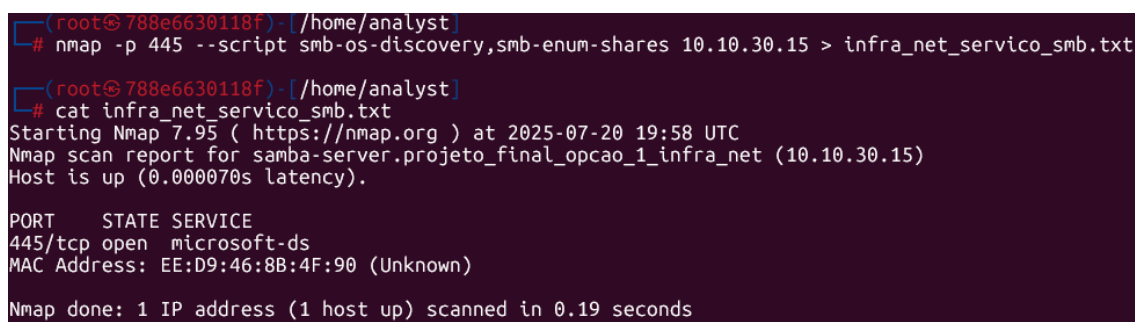
- Inclua as informações coletadas no inventário técnico e no relatório de diagnóstico.
- Identifique possíveis riscos associados ao serviço LDAP exposto na rede

SMB

O SMB (Server Message Block) é um protocolo utilizado para compartilhamento de arquivos, impressoras e outros recursos em redes locais. Ele permite que dispositivos conectados à rede acessem e utilizem recursos de forma colaborativa, sendo amplamente empregado em ambientes corporativos e domésticos. O SMB facilita a integração entre sistemas operacionais, como Windows e Linux, garantindo eficiência na troca de dados. No entanto, devido à sua exposição em redes, é essencial configurar o SMB com boas práticas de segurança, como restrição de acessos, autenticação robusta e desativação de versões antigas do protocolo, para evitar vulnerabilidades e proteger os recursos compartilhados.

Comando:

```
` nmap -p 445 --script smb-os-discovery,smb-enum-shares 10.10.30.15 > infra_net_servico_smb.txt`
```



```
(root@788e6630118f) - [ /home/analyst ]
# nmap -p 445 --script smb-os-discovery,smb-enum-shares 10.10.30.15 > infra_net_servico_smb.txt

(root@788e6630118f) - [ /home/analyst ]
# cat infra_net_servico_smb.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-20 19:58 UTC
Nmap scan report for samba-server.projeto_final_opcao_1_infra_net (10.10.30.15)
Host is up (0.000070s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds
MAC Address: EE:D9:46:8B:4F:90 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Figura 14: Print 14 – Comando `` nmap -p 445 --script smb-os-discovery,smb-enum-shares 10.10.30.15 > infra_net_servico_smb.txt``.

Explicação do Comando:

Partes do Comando:

1. **nmap**: Ferramenta para varredura de redes.
2. **-p 445**: Especifica que o Nmap deve escanear apenas a porta 445, que é usada pelo protocolo SMB (Server Message Block).
3. **--script smb-os-discovery,smb-enum-shares**:
 - **smb-os-discovery**: Script que tenta identificar o sistema operacional do host através do protocolo SMB.
 - **smb-enum-shares**: Script que enumera os compartilhamentos SMB disponíveis no host.
4. **10.10.30.15**: Endereço IP do host alvo (neste caso, o servidor SMB).
5. **> infra_net_servico_smb.txt**: Redireciona a saída do comando para o arquivo `infra_net_servico_smb.txt`.

Saída do Arquivo `infra_net_servico_smb.txt`:

...

Starting Nmap 7.95 (<https://nmap.org>) at 2025-07-20 19:58 UTC

Nmap scan report for samba-server.projeto_final_opcao_1_infra_net (10.10.30.15)

Host is up (0.000070s latency).

PORT STATE SERVICE

445/tcp open microsoft-ds

MAC Address: EE:D9:46:8B:4F:90 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds

...

Análise da Saída:

1. Host Information:

- **Host is up:** O host 10.10.30.15 está ativo e respondeu à varredura.
- **MAC Address:** O endereço MAC do dispositivo foi identificado como EE:D9:46:8B:4F:90, mas o fabricante não foi reconhecido.

2. Porta e Serviço:

- **445/tcp open microsoft-ds:** A porta 445 está aberta e o serviço identificado é o microsoft-ds, que é usado para compartilhamento de arquivos e impressoras via SMB.

3. Scripts Executados:

- Apesar de os scripts `smb-os-discovery` e `smb-enum-shares` terem sido especificados, a saída não mostra informações adicionais sobre o sistema operacional ou os compartilhamentos SMB. Isso pode indicar que:
 - O host não respondeu às consultas SMB.
 - O acesso ao serviço SMB está restrito ou protegido por autenticação.
 - O script não conseguiu obter informações adicionais.

4. Tempo de Execução:

- A varredura foi concluída em 0.19 segundos.

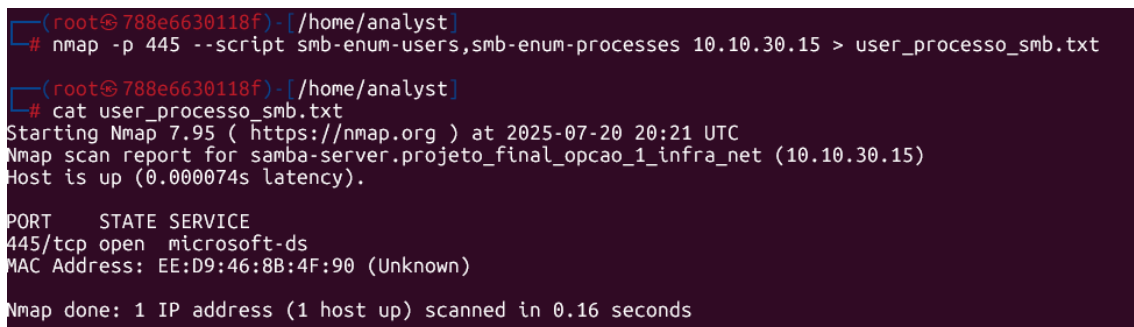
Interpretação:

- O servidor SMB no IP 10.10.30.15 está ativo e acessível na porta 445.
- O serviço SMB está configurado, mas os scripts não retornaram informações detalhadas sobre o sistema operacional ou os compartilhamentos disponíveis.
- Isso pode indicar que o servidor SMB está configurado com restrições de acesso ou que os scripts não conseguiram explorar o serviço adequadamente.

1. Executar Scripts Adicionais:

- Tente outros scripts SMB do Nmap para coletar mais informações:

```
`nmap -p 445 --script smb-enum-users,smb-enum-processes 10.10.30.15`
```



```
(root@788e6630118f) [/home/analyst]
# nmap -p 445 --script smb-enum-users,smb-enum-processes 10.10.30.15 > user_processo_smb.txt

(root@788e6630118f) [/home/analyst]
# cat user_processo_smb.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-20 20:21 UTC
Nmap scan report for samba-server.projeto_final_opcao_1_infra_net (10.10.30.15)
Host is up (0.000074s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: EE:D9:46:8B:4F:90 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
```

Figura 15: Print 15 – Comando `nmap -p 445 --script smb-enum-users,smb-enum-processes 10.10.30.15`.

Explicação do Comando:

Partes do Comando:

1. **nmap**: Ferramenta para varredura de redes.
2. **-p 445**: Especifica que o Nmap deve escanear apenas a porta 445, que é usada pelo protocolo SMB (Server Message Block).
3. **--script smb-enum-users,smb-enum-processes**:
 - **smb-enum-users**: Script que tenta enumerar os usuários SMB configurados no servidor.
 - **smb-enum-processes**: Script que tenta listar os processos em execução no servidor SMB.
4. **10.10.30.15**: Endereço IP do host alvo (neste caso, o servidor SMB).
5. **> user_processo_smb.txt**: Redireciona a saída do comando para o arquivo user_processo_smb.txt.

Saída do Comando:

...

Starting Nmap 7.95 (<https://nmap.org>) at 2025-07-20 20:21 UTC

Nmap scan report for samba-server.projeto_final_opcao_1_infra_net (10.10.30.15)

Host is up (0.000074s latency).

PORT STATE SERVICE

445/tcp open microsoft-ds

MAC Address: EE:D9:46:8B:4F:90 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds

...

Análise da Saída:

1. Host Information:

- **Host is up:** O host 10.10.30.15 está ativo e respondeu à varredura.
- **MAC Address:** O endereço MAC do dispositivo foi identificado como EE:D9:46:8B:4F:90, mas o fabricante não foi reconhecido.

2. Porta e Serviço:

- **445/tcp open microsoft-ds:** A porta 445 está aberta e o serviço identificado é microsoft-ds, que é usado para compartilhamento de arquivos e impressoras via SMB.

3. Scripts Executados:

- Apesar de os scripts smb-enum-users e smb-enum-processes terem sido especificados, a saída não mostra informações adicionais sobre usuários ou processos SMB. Isso pode indicar:
 - Restrições de acesso no servidor SMB.
 - Falha na execução dos scripts devido à falta de permissões ou configurações no servidor.
 - O servidor SMB pode estar configurado para não expor essas informações.

4. Tempo de Execução:

- A varredura foi concluída em 0.16 segundos.

Interpretação:

- O servidor SMB no IP 10.10.30.15 está ativo e acessível na porta 445.
- O serviço SMB está configurado, mas os scripts não retornaram informações detalhadas sobre usuários ou processos SMB.
- Isso pode indicar que o servidor SMB está configurado com restrições de acesso ou que os scripts não conseguiram explorar o serviço adequadamente.

2. Verificar Configuração do Servidor SMB:

- Se você tiver acesso ao servidor, verifique as permissões e configurações de compartilhamento para garantir que estão seguras.

3. Documentar no Relatório:

- Inclua as informações coletadas no inventário técnico e no relatório de diagnóstico.
- Identifique possíveis riscos associados ao serviço SMB exposto na rede.

Tabela Descritiva:

IP	Porta	Serviço	MAC Address	Informações Adicionais
10.10.30.15	445	microsoft-ds (SMB)	EE:D9:46:8B:4F:90	Nenhuma informação adicional retornada

HTTP (web) 1ª parte - Nginx

O HTTP (Hypertext Transfer Protocol) é o protocolo base para a comunicação na web, permitindo a transferência de informações entre servidores e clientes, como navegadores. Ele é responsável por carregar páginas web, enviar formulários e acessar conteúdos online, sendo essencial para o funcionamento da internet. Apesar de sua eficiência, o HTTP padrão não oferece criptografia, o que pode expor dados sensíveis durante a transmissão. Por isso, é recomendável utilizar o HTTPS, que adiciona uma camada de segurança com criptografia SSL/TLS, garantindo a proteção e a confidencialidade das informações trocadas entre usuários e servidores.

Comando:

```
`curl -I http://10.10.30.117 > infra_net_servico_webserver.txt`
```

```
(root@788e6630118f) - [/home/analyst]
# curl -I http://10.10.30.117 > infra_net_servico_webserver.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left   Speed
0          0          0     0         0      0         0         0  --:--:-- --:--:-- --:--:--    0

(root@788e6630118f) - [/home/analyst]
# cat infra_net_servico_webserver.txt
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 20 Jul 2025 20:25:00 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Keep-Alive: timeout=20
X-Powered-By: PHP/7.3.14
Set-Cookie: PHPSESSID=4475570658e7d76e96e1e2e9d0fba8df; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
```

Figura 16: Print 16 – Comando `curl -I http://10.10.30.117 > infra_net_servico_webserver.txt`.

Explicação do Comando:

Partes do Comando:

1. **curl**: Ferramenta de linha de comando usada para transferir dados de ou para um servidor.
2. **-I**: Solicita apenas os cabeçalhos HTTP da resposta, sem o corpo do conteúdo.
3. **http://10.10.30.117**: URL do servidor web alvo (neste caso, o IP 10.10.30.117).

4. >: Redireciona a saída do comando para o arquivo `infra_net_servico_webserver.txt`.

Saída do Arquivo `infra_net_servico_webserver.txt`:

...

HTTP/1.1 200 OK

Server: nginx

Date: Sun, 20 Jul 2025 20:25:00 GMT

Content-Type: text/html; charset=UTF-8

Connection: keep-alive

Keep-Alive: timeout=20

X-Powered-By: PHP/7.3.14

Set-Cookie: PHPSESSID=4475570658e7d76e96e1e2e9d0fba8df; HttpOnly

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate

Pragma: no-cache

X-Content-Type-Options: nosniff

X-XSS-Protection: 1; mode=block

X-Frame-Options: SAMEORIGIN

...

Análise da Saída:

1. Status HTTP:

- **HTTP/1.1 200 OK:** Indica que a solicitação foi bem-sucedida e o servidor respondeu normalmente.

2. Servidor Web:

- **Server: nginx:** O servidor web em execução é o **Nginx**.

3. Data e Hora:

- **Date: Sun, 20 Jul 2025 20:25:00 GMT:** Data e hora da resposta do servidor, no formato GMT.

4. Tipo de Conteúdo:

- **Content-Type: text/html; charset=UTF-8:** O conteúdo retornado pelo servidor é HTML com codificação de caracteres UTF-8.

5. Conexão:

- **Connection: keep-alive:** Indica que a conexão será mantida aberta para reutilização.
- **Keep-Alive: timeout=20:** O tempo limite para manter a conexão aberta é de 20 segundos.

6. Tecnologia do Servidor:

- **X-Powered-By: PHP/7.3.14:** O servidor está usando PHP na versão 7.3.14.

7. Cookies:

- **Set-Cookie: PHPSESSID=4475570658e7d76e96e1e2e9d0fba8df; HttpOnly:** O servidor configurou um cookie de sessão chamado PHPSESSID.

8. Cache:

- **Expires: Thu, 19 Nov 1981 08:52:00 GMT:** Indica que o conteúdo expirou e não deve ser armazenado em cache.
- **Cache-Control: no-store, no-cache, must-revalidate:** Reforça que o conteúdo não deve ser armazenado em cache.
- **Pragma: no-cache:** Compatibilidade com navegadores mais antigos para evitar cache.

9. Segurança:

- **X-Content-Type-Options: nosniff:** Evita que navegadores interpretem o conteúdo como um tipo diferente do especificado.
- **X-XSS-Protection: 1; mode=block:** Ativa a proteção contra ataques XSS (Cross-Site Scripting) no navegador.
- **X-Frame-Options: SAMEORIGIN:** Impede que o conteúdo seja carregado em iframes de domínios diferentes, protegendo contra ataques de clickjacking.

Interpretação:

- O servidor web no IP 10.10.30.117 está ativo e executando o Nginx com suporte a PHP 7.3.14.
- O cabeçalho HTTP indica que o servidor está configurado com boas práticas de segurança, como proteção contra XSS, clickjacking e restrições de cache.
- O cookie de sessão (PHPSESSID) sugere que o servidor pode estar executando uma aplicação web que utiliza sessões PHP.

1. Analisar o PHP:

- Verifique se o servidor expõe informações sensíveis, como arquivos de configuração ou logs.

2. Documentar no Relatório:

- Inclua as informações coletadas no inventário técnico e no relatório de diagnóstico.
- Identifique possíveis riscos associados ao servidor web exposto na rede.

Em resumo o comando ``curl -I http://10.10.30.117 > infra_net_servico_webserver.txt`` foi utilizado para capturar os cabeçalhos HTTP do servidor web no IP 10.10.30.117, salvando-os em um arquivo para análise. A resposta indica que o servidor está ativo, executando Nginx com suporte a PHP 7.3.14, e configurado com boas práticas de segurança, como proteção contra XSS, clickjacking e restrições de cache. Além disso, o uso de cookies de sessão sugere a presença de uma aplicação web. Essas informações devem ser documentadas no relatório técnico, destacando possíveis riscos e verificando se há exposição de dados sensíveis.

Explicação do Comando:

Partes do Comando:

1. **curl**: Ferramenta de linha de comando usada para transferir dados de ou para um servidor.
2. **http://10.10.30.117**: URL do servidor web alvo (neste caso, o IP 10.10.30.117).
3. **>**: Redireciona a saída do comando para o arquivo `infra_net_servico_zabbix.txt`.

Saída do Comando:

O comando salva o conteúdo HTML da página inicial do servidor web no arquivo `infra_net_servico_zabbix.txt`. A saída indica que o servidor está executando uma instância do **Zabbix**, uma ferramenta de monitoramento de rede.

Análise da Saída:

1. Título da Página:

- **<title>Zabbix docker: Zabbix</title>**: O título da página indica que o servidor está executando o Zabbix em um contêiner Docker.

2. Formulário de Login:

- A página contém um formulário de login com campos para **Username** e **Password**

3. Links de Ajuda e Suporte:

- A página contém links para a documentação e suporte do Zabbix:
 - **Documentação**: <https://www.zabbix.com/documentation/4.4/>
 - **Suporte**: <https://www.zabbix.com/support>

4. Identidade do Servidor:

- O servidor é identificado como **"Zabbix docker"**, sugerindo que o Zabbix está sendo executado em um contêiner Docker.

5. Tecnologia Utilizada:

- O HTML inclui referências a arquivos CSS e JavaScript, como:
 - **CSS**: `assets/styles/blue-theme.css`
 - **JavaScript**: `js/browsers.js`

6. Segurança:

- A página inclui boas práticas de segurança, como:
 - **X-Frame-Options: SAMEORIGIN:** Protege contra ataques de clickjacking.
 - **X-XSS-Protection: 1; mode=block:** Protege contra ataques de Cross-Site Scripting (XSS).

Interpretação:

- O servidor no IP 10.10.30.117 está executando uma instância do Zabbix, uma ferramenta de monitoramento de rede.
- A interface de login sugere que o servidor está configurado para autenticação de usuários.

Tabela Descritiva:

IP	Serviço	Tecnologia	Descrição
10.10.30.117	Web Server	Zabbix (Docker)	Servidor web executando o Zabbix com uma interface de login acessível.

Extras úteis

Arp

O comando ARP (Address Resolution Protocol) é utilizado para mapear endereços IP a endereços MAC em uma rede local, permitindo a identificação de dispositivos conectados. Ao executar o comando `arp -a`, é possível visualizar a tabela ARP, que lista os dispositivos ativos na rede, seus endereços IP, endereços MAC e a interface de rede utilizada. No contexto deste projeto, o comando foi redirecionado para o arquivo `recon_ip_maps.txt`, armazenando as informações coletadas para análise e documentação. Essa abordagem facilita o reconhecimento da infraestrutura de rede, ajudando a identificar dispositivos e possíveis inconsistências na comunicação entre eles.

Comando:

```
`arp -a > recon_ip_maps.txt`
```

```
(root@788e6630118f) - [/home/analyst]
# arp -a > recon_ip_maps.txt

(root@788e6630118f) - [/home/analyst]
# cat recon_ip_maps.txt
macbook-aline.projeto_final_opcao_1_guest_net (10.10.50.4) at 82:b7:66:11:57:d6 [ether] on eth0
WS_003.projeto_final_opcao_1_corp_net (10.10.10.127) at a6:0c:f2:a2:0f:0d [ether] on eth2
notebook-carlos.projeto_final_opcao_1_guest_net (10.10.50.5) at 52:ca:46:e9:22:d5 [ether] on eth0
openldap.projeto_final_opcao_1_infra_net (10.10.30.17) at da:4e:40:4c:7b:91 [ether] on eth1
mysql-server.projeto_final_opcao_1_infra_net (10.10.30.11) at 96:aa:26:a2:cb:9a [ether] on eth1
? (10.10.50.1) at 52:f1:84:55:7e:70 [ether] on eth0
samba-server.projeto_final_opcao_1_infra_net (10.10.30.15) at ee:d9:46:8b:4f:90 [ether] on eth1
WS_001.projeto_final_opcao_1_corp_net (10.10.10.10) at 82:4a:c4:3d:14:05 [ether] on eth2
? (10.10.10.1) at ea:b2:04:f5:ef:87 [ether] on eth2
zabbix-server.projeto_final_opcao_1_infra_net (10.10.30.117) at 5a:d7:8c:af:8f:f6 [ether] on eth1
ftp-server.projeto_final_opcao_1_infra_net (10.10.30.10) at f6:c5:7c:ce:cd:cc [ether] on eth1
? (10.10.30.1) at 56:ea:69:69:18:54 [ether] on eth1
laptop-vastro.projeto_final_opcao_1_guest_net (10.10.50.2) at 0e:ef:33:a7:c6:fb [ether] on eth0
legacy-server.projeto_final_opcao_1_infra_net (10.10.30.227) at 2a:40:30:5f:40:e8 [ether] on eth1
WS_004.projeto_final_opcao_1_corp_net (10.10.10.222) at 76:3b:07:05:b4:b7 [ether] on eth2
laptop-luiz.projeto_final_opcao_1_guest_net (10.10.50.3) at f2:a4:72:cb:5b:d7 [ether] on eth0
WS_002.projeto_final_opcao_1_corp_net (10.10.10.101) at 02:1a:a6:ae:2a:a7 [ether] on eth2
```

Figura 18: Print 18 – Comando ``arp -a > recon_ip_maps.txt``.

Explicação do Comando:

Partes do Comando:

1. **arp -a:** Exibe a tabela ARP (Address Resolution Protocol), que mapeia endereços IP para endereços MAC na rede local.
 - Mostra os dispositivos conectados à rede, incluindo seus endereços IP, endereços MAC e a interface de rede usada.
2. **>:** Redireciona a saída do comando para o arquivo `recon_ip_maps.txt`.

Saída do Arquivo recon_ip_maps.txt:

A saída contém informações sobre dispositivos na rede, incluindo:

- **Nome do dispositivo** (se resolvido).
- **Endereço IP.**
- **Endereço MAC.**
- **Interface de rede** usada para comunicação.

Tabela Descritiva:

Dispositivo	Endereço IP	Endereço MAC	Interface	Rede
macbook- aline.projeto_final_opcao_1_guest_net	10.10.50.4	82:b7:66:11:57:d6	eth0	guest_net
WS_003.projeto_final_opcao_1_corp_net	10.10.10.127	a6:0c:f2:a2:0f:0d	eth2	corp_net
notebook- carlos.projeto_final_opcao_1_guest_net	10.10.50.5	52:ca:46:e9:22:d5	eth0	guest_net
openldap.projeto_final_opcao_1_infra_net	10.10.30.17	da:4e:40:4c:7b:91	eth1	infra_net
mysql- server.projeto_final_opcao_1_infra_net	10.10.30.11	96:aa:26:a2:cb:9a	eth1	infra_net
Switch: Guest Network - SubNet: 10.10.50.0/24	10.10.50.1	52:f1:84:55:7e:70	eth0	guest_net
samba- server.projeto_final_opcao_1_infra_net	10.10.30.15	ee:d9:46:8b:4f:90	eth1	infra_net
WS_001.projeto_final_opcao_1_corp_net	10.10.10.10	82:4a:c4:3d:14:05	eth2	corp_net
Switch: Corp Network - SubNet: 10.10.10.0/24	10.10.10.1	ea:b2:04:f5:ef:87	eth2	corp_net
zabbix- server.projeto_final_opcao_1_infra_net	10.10.30.117	5a:d7:8c:af:8f:f6	eth1	infra_net
ftp- server.projeto_final_opcao_1_infra_net	10.10.30.10	f6:c5:7c:ce:cd:cc	eth1	infra_net
Switch: Infra Network - SubNet: 10.10.30.0/24	10.10.30.1	56:ea:69:69:18:54	eth1	infra_net
laptop- vastro.projeto_final_opcao_1_guest_net	10.10.50.2	0e:ef:33:a7:c6:fb	eth0	guest_net
legacy- server.projeto_final_opcao_1_infra_net	10.10.30.227	2a:40:30:5f:40:e8	eth1	infra_net
WS_004.projeto_final_opcao_1_corp_net	10.10.10.222	76:3b:07:05:b4:b7	eth2	corp_net
laptop- luiz.projeto_final_opcao_1_guest_net	10.10.50.3	f2:a4:72:cb:5b:d7	eth0	guest_net
WS_002.projeto_final_opcao_1_corp_net	10.10.10.101	02:1a:a6:ae:2a:a7	eth2	corp_net

Detalhes da Tabela:

1. **Dispositivo:** Nome do dispositivo na rede, se resolvido, ou identificador genérico caso o nome.
2. **Endereço IP:** Endereço IP do dispositivo na rede.
3. **Endereço MAC:** Endereço físico do dispositivo (identificador único da interface de rede).
4. **Interface:** Interface de rede local usada para comunicação.
5. **Rede:** Identificação da sub-rede à qual o dispositivo pertence.

Observações:

1. Segmentação de Rede:

- A tabela mostra claramente a separação entre diferentes sub-redes, como corp_net, infra_net e guest_net.
- Essa segmentação é útil para identificar dispositivos pertencentes a diferentes áreas funcionais ou níveis de acesso na rede.

2. Identificação de Switches:

- Os switches são identificados como dispositivos centrais em cada sub-rede, com seus respectivos endereços IP e MAC.
- Eles desempenham um papel crucial na comunicação entre dispositivos dentro da mesma sub-rede.

3. Dispositivos Críticos:

- Servidores importantes, como mysql-server, samba-server, e zabbix-server, estão listados na sub-rede infra_net, indicando que essa rede é destinada a infraestrutura e serviços essenciais.
- Isso pode ser útil para priorizar segurança e monitoramento.

4. Interfaces de Rede:

- A tabela especifica qual interface de rede (eth0, eth1, eth2) cada dispositivo utiliza, o que é importante para diagnósticos e configuração de rede.
- Pode ajudar a identificar possíveis problemas de conectividade ou conflitos de interface.

5. Endereços Resolvidos:

- Alguns dispositivos têm nomes resolvidos (ex.: macbook-aline), enquanto outros aparecem apenas com identificadores genéricos.
- Isso pode indicar que nem todos os dispositivos têm DNS ou mapeamento configurado corretamente.

6. Sub-redes e Máscaras:

- As sub-redes são identificadas com suas faixas de IP (ex.: 10.10.50.0/24), facilitando a análise de alcance e limites de cada rede.
- Isso é útil para planejamento de endereçamento IP e expansão da rede.

7. Dispositivos Legados:

- O dispositivo legacy-server na sub-rede infra_net pode indicar a presença de sistemas antigos que ainda estão em uso.
- Esses dispositivos podem exigir atenção especial em termos de compatibilidade e segurança.

8. Distribuição de Dispositivos:

- A tabela mostra uma distribuição equilibrada de dispositivos entre as sub-redes, indicando uma possível organização lógica da rede.
- Isso pode ser útil para entender o fluxo de dados e carga em cada segmento.

9. Segurança:

- A separação de redes como guest_net e corp_net sugere que há medidas de segurança para isolar dispositivos de visitantes e dispositivos corporativos.
- Isso reduz o risco de acesso não autorizado a recursos internos.

10. Monitoramento e Auditoria:

- A tabela pode ser usada como base para monitoramento contínuo da rede, ajudando a identificar dispositivos desconhecidos ou não autorizados.
- Também pode servir como documentação para auditorias de rede.

Essas observações podem ajudar na análise, manutenção e melhoria da infraestrutura de rede.

Organização dos Resultados

```

```
mkdir -p /home/analyst/recon/{corp_net,guest_net,infra_net}
```

```
mv *corp*.txt /home/analyst/recon/corp_net/
```

```
mv *guest*.txt /home/analyst/recon/guest_net/
```

```
mv *infra*.txt /home/analyst/recon/infra_net/
```

```
mv *recon*.txt /home/analyst/recon/
```

```

Explicação dos Comandos:

1. Criar Diretórios para Organização:

```
`mkdir -p /home/analyst/recon/{corp_net,guest_net,infra_net}`
```

- **mkdir -p:** Cria diretórios, incluindo diretórios pai, se necessário.
- **/home/analyst/recon/{corp_net,guest_net,infra_net}:** Cria a estrutura de diretórios:
 - /home/analyst/recon/corp_net
 - /home/analyst/recon/guest_net
 - /home/analyst/recon/infra_net

2. Mover Arquivos Relacionados à Rede Corporativa:

```
`mv *corp*.txt /home/analyst/recon/corp_net/`
```

- **mv:** Move arquivos.
- ***corp*.txt:** Seleciona todos os arquivos cujo nome contém "corp" e termina com .txt.
- **/home/analyst/recon/corp_net/:** Move os arquivos para o diretório corp_net.

3. Mover Arquivos Relacionados à Rede de Convidados:

```
mv *guest*.txt /home/analyst/recon/guest_net/
```

- Move todos os arquivos cujo nome contém "guest" e termina com .txt para o diretório guest_net.

4. Mover Arquivos Relacionados à Rede de Infraestrutura:

```
mv *infra*.txt /home/analyst/recon/infra_net/
```

- Move todos os arquivos cujo nome contém "infra" e termina com .txt para o diretório infra_net.

5. Mover Arquivos de Reconhecimento Geral:

```
`mv *recon*.txt /home/analyst/recon/`
```

- Move todos os arquivos cujo nome contém "recon" e termina com .txt para o diretório principal /home/analyst/recon/.

Resumo da Organização:

Diretório	Arquivos Movidos
/home/analyst/recon/corp_net/	Arquivos relacionados à rede corporativa (*corp*.txt)
/home/analyst/recon/guest_net/	Arquivos relacionados à rede de convidados (*guest*.txt)
/home/analyst/recon/infra_net/	Arquivos relacionados à rede de infraestrutura (*infra*.txt)
/home/analyst/recon/	Arquivos gerais de reconhecimento (*recon*.txt)

Objetivo: Esses comandos ajudam a organizar os resultados de varreduras e análises de rede em diretórios específicos, mantendo o ambiente de trabalho limpo e estruturado. Isso facilita a navegação e a análise posterior dos dados.

Inventário Final - Tabela Descritiva

Infraestrutura - Rede Infra_net

IP	Hostname	SO Estimado	Portas Abertas	Serviços	Notas
10.10.30.10	ftp-server.projeto_final_opcao_1_infra_net	Não identificado	21	FTP	Serviço FTP ativo, verificar se permite login anônimo.
10.10.30.11	mysql-server.projeto_final_opcao_1_infra_net	Não identificado	3306, 33060	MySQL	MySQL versão 8.0.42, verificar configurações de autenticação e permissões.
10.10.30.15	samba-server.projeto_final_opcao_1_infra_net	Não identificado	139, 445	SMB	Serviço SMB ativo, verificar compartilhamentos e permissões.

10.10.30.17	openldap.projeto_final_opcao_1_infra_net	Não identificado	389, 636	LDAP	Serviço LDAP ativo, verificar configurações de segurança e autenticação.
10.10.30.117	zabbix-server.projeto_final_opcao_1_infra_net	Não identificado	80, 10051, 10052	HTTP (Zabbix), Zabbix Agent	Página de login do Zabbix acessível, verificar segurança de credenciais.
10.10.30.227	legacy-server.projeto_final_opcao_1_infra_net	Não identificado	Não identificado	Não identificado	Necessário investigar mais detalhes.

Infraestrutura - Rede Guest_net

IP	Hostname	SO Estimado	Portas Abertas	Serviços	Notas
10.10.50.1	Não identificado	Não identificado	111, 51789	Não identificado	Necessário investigar mais detalhes.
10.10.50.2	laptop-vastro.projeto_final_opcao_1_guest_net	Não identificado	Não identificado	Não identificado	Necessário investigar mais detalhes.
10.10.50.3	laptop-luiz.projeto_final_opcao_1_guest_net	Não identificado	Não identificado	Não identificado	Necessário investigar mais detalhes.
10.10.50.4	macbook-aline.projeto_final_opcao_1_guest_net	Não identificado	Não identificado	Não identificado	Necessário investigar mais detalhes.
10.10.50.5	notebook-carlos.projeto_final_opcao_1_guest_net	Não identificado	Não identificado	Não identificado	Necessário investigar mais detalhes.
10.10.50.6	Não identificado	Não identificado	51146, 57236	Não identificado	Necessário investigar mais detalhes.

Infraestrutura - Rede Corp_net

IP	Hostname	SO Estimado	Portas Abertas	Serviços	Notas
10.10.10.10	WS_001.projeto_final_opcao_1_corp_net	Não identificado	Não identificado	Não identificado	Necessário investigar mais detalhes.
10.10.10.101	WS_002.projeto_final_opcao_1_corp_net	Não identificado	Não identificado	Não identificado	Necessário investigar mais detalhes.
10.10.10.127	WS_003.projeto_final_opcao_1_corp_net	Não identificado	Não identificado	Não identificado	Necessário investigar mais detalhes.
10.10.10.222	WS_004.projeto_final_opcao_1_corp_net	Não identificado	Não identificado	Não identificado	Necessário investigar mais detalhes.

Diagrama

O diagrama de rede desenvolvido para este projeto oferece uma representação visual clara da infraestrutura, destacando a organização dos dispositivos, sub-redes e conexões entre eles. Ele facilita a compreensão da topologia da rede, permitindo identificar pontos críticos, serviços ativos e fluxos de comunicação. Além de servir como uma ferramenta essencial para análise e planejamento, o diagrama também auxilia na identificação de possíveis vulnerabilidades e na implementação de melhorias, garantindo maior eficiência e segurança na gestão da infraestrutura.

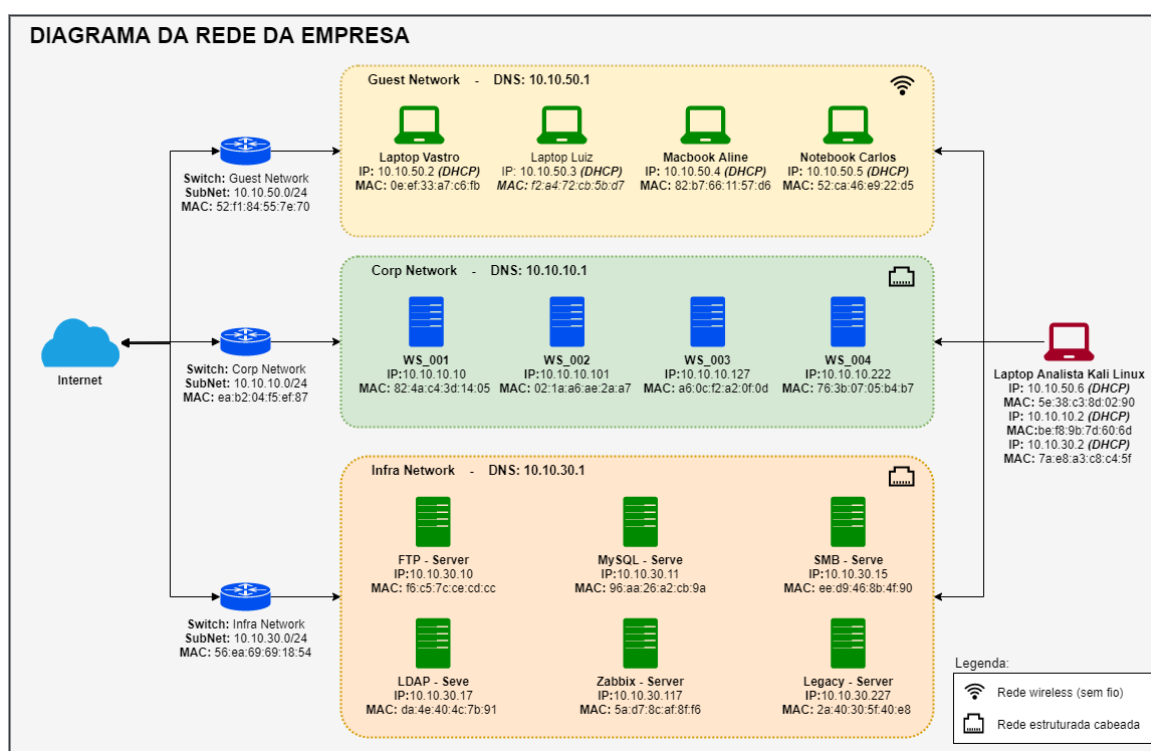


Figura 19: Diagrama 01 – Diagrama da rede da empresa.

Diagnóstico

O diagnóstico realizado neste projeto seguiu uma metodologia estruturada e detalhada, com o objetivo de mapear redes e ativos, identificar serviços e vulnerabilidades, e organizar os resultados para análise. Cada etapa foi cuidadosamente planejada e executada, utilizando ferramentas e técnicas específicas para garantir a coleta de informações relevantes e a identificação de possíveis riscos.

O reconhecimento inicial da rede permitiu identificar interfaces, endereços IP e conectividade entre sub-redes, enquanto a descoberta de hosts ativos, realizada com o Nmap, facilitou a segmentação e documentação dos dispositivos. A varredura de portas com Rustscan garantiu rapidez na identificação de serviços expostos, complementada por uma análise aprofundada com ferramentas como Nmap, curl e ftp, que detalharam configurações e vulnerabilidades. A organização dos resultados em diretórios específicos e a criação de um inventário final estruturado consolidaram as informações, proporcionando uma visão clara e completa da infraestrutura de rede.

Esse diagnóstico detalhado serve como base para a análise de segurança e planejamento de ações corretivas, garantindo uma abordagem eficiente e proativa na gestão da infraestrutura.

A metodologia aplicada neste projeto seguiu um processo estruturado para mapear redes e ativos, identificar serviços e vulnerabilidades, e organizar os resultados para análise. Abaixo está o diagnóstico detalhado:

1. Reconhecimento da Rede

- **Comandos Utilizados:** ip a, ip a | grep inet, ping.

Resultados:

- Identificação das interfaces de rede, endereços IP, máscaras de sub-rede e estado das conexões.
- Teste de conectividade com as redes corp_net, guest_net e infra_net, confirmando estabilidade e proximidade na rede.
- **Diagnóstico:**
 - A coleta inicial foi eficiente, fornecendo informações detalhadas sobre a configuração da rede.
 - A conectividade foi confirmada sem perda de pacotes, indicando uma infraestrutura estável.

2. Descoberta de Hosts

- **Ferramenta Utilizada:** nmap com varredura de ping (-sn).
- **Resultados:**
 - Identificação de hosts ativos em cada sub-rede (corp_net, infra_net, guest_net).
 - Organização dos IPs e nomes de host em arquivos para análise posterior.
- **Diagnóstico:**
 - A metodologia foi eficaz para mapear rapidamente os dispositivos ativos.
 - A segmentação por sub-rede facilitou a análise e documentação.

3. Scan de Portas

- **Ferramenta Utilizada:** Rustscan.
- **Resultados:**
 - Identificação de portas abertas em dispositivos ativos.
 - Detecção de serviços como FTP, MySQL, LDAP, SMB e HTTP.
- **Diagnóstico:**
 - O uso do Rustscan garantiu rapidez e eficiência na varredura de portas.
 - A identificação de serviços expostos é crucial para análise de segurança.

4. Análise de Serviços

- **Ferramentas Utilizadas:** nmap, curl, ftp.
- **Resultados:**
 - Coleta de informações detalhadas sobre serviços como FTP, MySQL, LDAP, SMB e HTTP.
 - Identificação de versões, configurações e possíveis vulnerabilidades.

- **Diagnóstico:**

- A análise foi aprofundada, permitindo identificar riscos específicos, como serviços mal configurados ou expostos.
- A documentação dos resultados foi bem estruturada, facilitando a interpretação.

5. Organização dos Resultados

- **Comandos Utilizados:** mkdir, mv.

- **Resultados:**

- Organização dos arquivos em diretórios específicos para cada sub-rede.

- **Diagnóstico:**

- A organização foi eficiente, garantindo um ambiente de trabalho limpo e estruturado.
- Facilita a navegação e análise posterior dos dados.

6. Inventário Final

- **Resultados:**

- Tabelas descritivas com informações detalhadas sobre dispositivos, serviços e portas abertas.

- **Diagnóstico:**

- O inventário final é completo e bem organizado, permitindo uma visão clara da infraestrutura de rede.
- Identifica dispositivos críticos e possíveis riscos.

Recomendações

As recomendações apresentadas têm como objetivo fortalecer a segurança, monitoramento, documentação e auditoria da infraestrutura de rede, além de implementar medidas complementares para garantir sua resiliência e eficiência. Cada área foi detalhada com práticas específicas, ferramentas e estratégias que visam mitigar riscos, otimizar o desempenho e assegurar a conformidade com regulamentações aplicáveis.

A segurança é abordada com foco na configuração adequada de serviços expostos, restrição de acessos e proteção contra-ataques, utilizando autenticação robusta e protocolos seguros. O monitoramento contínuo, com ferramentas avançadas e alertas configurados, permite uma resposta ágil a incidentes e uma visão proativa da infraestrutura. A documentação estruturada facilita a gestão, recuperação de configurações e auditorias, enquanto as auditorias regulares garantem a conformidade, eficiência e identificação de vulnerabilidades. Por fim, medidas extras, como treinamento de equipe, redundância, segmentação de rede e testes de penetração, complementam a abordagem, promovendo uma proteção robusta e proativa.

Essas recomendações oferecem uma base sólida para implementar práticas eficazes que assegurem a integridade, confidencialidade e disponibilidade dos dados, além de garantir a estabilidade e segurança da infraestrutura de rede.

Segurança

A segurança da infraestrutura de rede é um aspecto crítico para garantir a proteção contra ameaças e vulnerabilidades. Configurar serviços expostos de forma segura, como FTP, MySQL, LDAP, SMB e HTTP, é essencial para mitigar riscos, utilizando autenticação robusta, protocolos seguros e restrições de acesso. Além disso, implementar listas de controle de acesso (ACLs), firewalls e autenticação multifator (MFA) fortalece a defesa contra acessos não autorizados. A proteção contra ataques é complementada por sistemas de prevenção e detecção de intrusão (IPS/IDS), monitoramento de logs e medidas contra força bruta, criando uma abordagem integrada para preservar a integridade, confidencialidade e disponibilidade dos dados.

1. Configuração de Serviços Expostos:

- **FTP:**
 - Desativar o acesso anônimo, caso esteja habilitado.
 - Implementar autenticação baseada em usuários e senhas fortes.
 - Configurar o FTP para usar TLS/SSL para proteger a transmissão de dados.
- **MySQL:**
 - Restringir conexões externas ao banco de dados, permitindo apenas hosts autorizados.
 - Revisar permissões de usuários e garantir que cada usuário tenha acesso apenas ao necessário.
 - Habilitar autenticação com `caching_sha2_password` para maior segurança.
- **LDAP:**
 - Configurar o servidor para aceitar conexões apenas de hosts confiáveis.
 - Implementar autenticação SASL com mecanismos seguros, como GSSAPI ou SCRAM-SHA-256.
 - Restringir consultas anônimas ao diretório.

- **SMB:**
 - Verificar permissões de compartilhamento e restringir acessos desnecessários.
 - Desativar versões antigas do protocolo SMB (ex.: SMBv1).
 - Implementar autenticação baseada em usuários e senhas fortes.
- **HTTP:**
 - Configurar HTTPS com certificados válidos para proteger a comunicação.
 - Revisar cabeçalhos de segurança (ex.: X-Frame-Options, X-XSS-Protection, Content-Security-Policy).
 - Garantir que aplicações web estejam atualizadas e livres de vulnerabilidades conhecidas.

2. Restrição de Acesso:

- Implementar listas de controle de acesso (ACLs) para limitar o tráfego entre sub-redes.
- Configurar firewalls para bloquear portas e serviços não utilizados.
- Utilizar autenticação multifator (MFA) para serviços críticos.

3. Proteção Contra Ataques:

- Configurar sistemas de prevenção/detecção de intrusão (IPS/IDS).
- Implementar proteção contra ataques de força bruta em serviços expostos.
- Monitorar logs de serviços para identificar atividades suspeitas.

A segurança da infraestrutura de rede exige uma abordagem abrangente e integrada para mitigar riscos e proteger contra ameaças. Configurar serviços expostos, como FTP, MySQL, LDAP, SMB e HTTP, com autenticação robusta e protocolos seguros é essencial para reduzir vulnerabilidades. A implementação de restrições de acesso, como ACLs, firewalls e autenticação multifator (MFA), fortalece a defesa contra acessos não autorizados. Além disso, sistemas de prevenção e detecção de intrusão (IPS/IDS), monitoramento de logs e proteção contra ataques de força bruta complementam as medidas de segurança, garantindo a integridade, confidencialidade e disponibilidade dos dados em toda a infraestrutura.

Monitoramento

O monitoramento é uma prática indispensável para garantir a estabilidade, segurança e desempenho da infraestrutura de rede. Ferramentas como Zabbix, Nagios e Prometheus permitem acompanhar dispositivos e serviços em tempo real, enquanto soluções como Wireshark e Zeek ajudam a analisar o tráfego de rede. Para segurança, plataformas como Splunk e ELK Stack oferecem recursos avançados de análise de logs. Além disso, configurar alertas para eventos críticos e garantir notificações em canais apropriados facilita a resposta rápida a incidentes. O monitoramento de integridade, com verificações regulares em servidores e arquivos sensíveis, complementa essa abordagem, permitindo uma visão proativa e detalhada da infraestrutura.

1. Ferramentas de Monitoramento:

- Configurar ferramentas como Zabbix, Nagios ou Prometheus para monitorar dispositivos e serviços.
- Implementar monitoramento de tráfego de rede com ferramentas como Wireshark ou Zeek.
- Utilizar soluções de monitoramento de segurança, como Splunk ou ELK Stack, para análise de logs.

2. Alertas e Notificações:

- Configurar alertas para eventos críticos, como falhas de serviços ou tentativas de acesso não autorizado.
- Garantir que notificações sejam enviadas para canais apropriados (ex.: e-mail, SMS, Slack).

3. Monitoramento de Integridade:

- Implementar verificações regulares de integridade em servidores e dispositivos críticos.
- Monitorar alterações em arquivos sensíveis ou configurações de serviços.

O monitoramento é essencial para assegurar a estabilidade, segurança e desempenho da infraestrutura de rede. Ferramentas como Zabbix, Nagios e Prometheus oferecem supervisão em tempo real de dispositivos e serviços, enquanto Wireshark e Zeek permitem análises detalhadas do tráfego de rede. Soluções como Splunk e ELK Stack aprimoram a segurança com recursos avançados de análise de logs. A configuração de alertas para eventos críticos e notificações em canais apropriados garante uma resposta ágil a incidentes. Além

disso, o monitoramento de integridade, com verificações regulares em servidores e arquivos sensíveis, reforça uma abordagem proativa e detalhada para a gestão da infraestrutura.

Documentação

A documentação é um elemento fundamental para a gestão eficiente da infraestrutura de rede, garantindo organização e suporte em processos críticos. Um inventário atualizado com informações detalhadas sobre dispositivos, serviços e suas dependências facilita a análise e o planejamento. Procedimentos operacionais bem documentados, incluindo guias para configuração, manutenção e resposta a incidentes, asseguram consistência e agilidade em situações adversas. Além disso, o registro de alterações na infraestrutura, aliado ao uso de sistemas de controle de versão como Git, permite rastrear modificações e manter um histórico confiável, essencial para auditorias e recuperação de configurações.

1. Inventário de Rede:

- Atualizar o inventário com informações detalhadas sobre dispositivos, serviços e configurações.
- Incluir informações sobre dependências entre serviços e dispositivos.

2. Procedimentos Operacionais:

- Documentar procedimentos para configuração, manutenção e recuperação de serviços.
- Criar guias para resposta a incidentes de segurança.

3. Histórico de Alterações:

- Manter um registro de alterações realizadas na infraestrutura, incluindo configurações de rede e serviços.
- Utilizar sistemas de controle de versão para gerenciar configurações (ex.: Git).

A documentação é essencial para garantir a gestão eficiente e organizada da infraestrutura de rede. Um inventário atualizado com informações detalhadas sobre dispositivos, serviços e suas dependências facilita o planejamento e a análise. Procedimentos operacionais bem estruturados, incluindo guias para configuração, manutenção e resposta a incidentes, promovem agilidade e consistência em situações críticas. Além disso, o registro de alterações na infraestrutura, aliado ao uso de sistemas de controle de versão como Git, assegura

rastreabilidade e confiabilidade, sendo indispensável para auditorias e recuperação de configurações.

Auditoria

A auditoria é uma prática indispensável para avaliar e fortalecer a segurança, conformidade e desempenho da infraestrutura de rede. Auditorias de segurança, realizadas com ferramentas como Nmap, Nessus ou OpenVAS, ajudam a identificar vulnerabilidades e validar configurações de serviços expostos. Auditorias de conformidade garantem que a infraestrutura esteja alinhada com regulamentações como GDPR, LGPD e PCI-DSS, além de revisar políticas de segurança e acesso. Já as auditorias de performance monitoram serviços críticos, identificam gargalos e testam a capacidade de resposta em cenários de alta carga, assegurando eficiência e estabilidade operacional.

1. Auditorias de Segurança:

- Realizar varreduras regulares com ferramentas como Nmap, Nessus ou OpenVAS para identificar vulnerabilidades.
- Testar configurações de serviços expostos para garantir conformidade com boas práticas de segurança.

2. Auditorias de Conformidade:

- Garantir que a infraestrutura esteja em conformidade com regulamentações aplicáveis (ex.: GDPR, LGPD, PCI-DSS).
- Revisar políticas de segurança e acesso regularmente.

3. Auditorias de Performance:

- Monitorar o desempenho de serviços críticos e identificar gargalos.
- Testar a capacidade de resposta da infraestrutura em cenários de alta carga.

A auditoria é essencial para garantir a segurança, conformidade e desempenho da infraestrutura de rede. Auditorias de segurança identificam vulnerabilidades e validam configurações de serviços expostos, utilizando ferramentas como Nmap, Nessus e OpenVAS. Auditorias de conformidade asseguram alinhamento com regulamentações como GDPR, LGPD e PCI-DSS, além de revisar políticas de acesso e segurança. Por fim, auditorias de performance monitoram serviços críticos, detectam gargalos e testam a capacidade de resposta em cenários de alta carga, promovendo eficiência e estabilidade operacional.

Extras

Os extras representam medidas complementares que fortalecem a segurança e a resiliência da infraestrutura de rede. O treinamento da equipe é essencial para capacitar profissionais na identificação e resposta a incidentes, além de promover boas práticas de segurança e uso de ferramentas de monitoramento. A implementação de redundância para serviços críticos e backups regulares assegura alta disponibilidade e recuperação de dados em caso de falhas. A segmentação de rede, com isolamento de sub-redes e uso de VLANs, reduz o impacto de incidentes e melhora o controle de tráfego. Por fim, testes de penetração realizados internamente ou por especialistas externos ajudam a identificar vulnerabilidades exploráveis, garantindo uma abordagem proativa na proteção da infraestrutura.

1. Treinamento de Equipe:

- Capacitar a equipe para identificar e responder a incidentes de segurança.
- Promover treinamentos sobre boas práticas de segurança e uso de ferramentas de monitoramento.

2. Redundância e Backup:

- Implementar redundância para serviços críticos, garantindo alta disponibilidade.
- Configurar backups regulares e testar a recuperação de dados.

3. Segmentação de Rede:

- Garantir que sub-redes estejam devidamente isoladas para minimizar o impacto de incidentes.
- Implementar VLANs para separar tráfego de diferentes tipos de dispositivos.

4. Testes de Penetração:

- Realizar testes de penetração para identificar vulnerabilidades exploráveis.
- Contratar especialistas externos para realizar avaliações independentes.

As medidas extras são fundamentais para reforçar a segurança e a resiliência da infraestrutura de rede. O treinamento da equipe promove a capacitação para lidar com incidentes e adotar boas práticas de segurança. A redundância para serviços

críticos e backups regulares garantem alta disponibilidade e recuperação de dados em situações adversas. A segmentação de rede, com isolamento de sub-redes e uso de VLANs, minimiza o impacto de incidentes e melhora o controle de tráfego. Por fim, testes de penetração, realizados internamente ou por especialistas externos, ajudam a identificar vulnerabilidades exploráveis, fortalecendo a proteção da infraestrutura de forma proativa.

Plano de Ação (modelo 80/20)

O plano de ação apresentado segue o modelo 80/20, priorizando as ações que geram maior impacto na segurança, estabilidade e eficiência da infraestrutura de rede. Com base no diagnóstico detalhado e nas recomendações propostas, as etapas foram organizadas para garantir que os esforços sejam concentrados nas áreas mais críticas, enquanto mantêm a flexibilidade para ajustes e melhorias contínuas.

1. Segurança

Prioridades (80% do impacto):

- Configurar serviços expostos (FTP, MySQL, LDAP, SMB e HTTP) com autenticação robusta e protocolos seguros.
- Implementar listas de controle de acesso (ACLs) para limitar o tráfego entre sub-redes.
- Configurar firewalls para bloquear portas e serviços não utilizados.
- Adotar autenticação multifator (MFA) para serviços críticos.
- Instalar e configurar sistemas de prevenção/detecção de intrusão (IPS/IDS).

Ações Complementares (20% do impacto):

- Monitorar logs de serviços para identificar atividades suspeitas.
- Realizar revisões periódicas das configurações de segurança.
- Atualizar cabeçalhos de segurança em servidores HTTP (ex.: X-Frame-Options, Content-Security-Policy).

2. Monitoramento

Prioridades (80% do impacto):

- Configurar ferramentas como Zabbix, Nagios ou Prometheus para monitorar dispositivos e serviços em tempo real.
- Implementar monitoramento de tráfego de rede com Wireshark ou Zeek.
- Configurar alertas para eventos críticos, como falhas de serviços ou tentativas de acesso não autorizado.

Ações Complementares (20% do impacto):

- Utilizar soluções de monitoramento de segurança, como Splunk ou ELK Stack, para análise avançada de logs.
- Realizar verificações regulares de integridade em servidores e dispositivos críticos.
- Monitorar alterações em arquivos sensíveis ou configurações de serviços.

3. Documentação

Prioridades (80% do impacto):

- Atualizar o inventário de rede com informações detalhadas sobre dispositivos, serviços e dependências.
- Documentar procedimentos operacionais para configuração, manutenção e recuperação de serviços.
- Criar guias para resposta a incidentes de segurança.

Ações Complementares (20% do impacto):

- Manter um registro de alterações realizadas na infraestrutura.
- Utilizar sistemas de controle de versão, como Git, para gerenciar configurações.
- Revisar e atualizar a documentação regularmente.

4. Auditoria

Prioridades (80% do impacto):

- Realizar varreduras regulares com ferramentas como Nmap, Nessus ou OpenVAS para identificar vulnerabilidades.
- Testar configurações de serviços expostos para garantir conformidade com boas práticas de segurança.
- Monitorar o desempenho de serviços críticos e identificar gargalos.

Ações Complementares (20% do impacto):

- Garantir conformidade com regulamentações aplicáveis (ex.: GDPR, LGPD, PCI-DSS).
- Revisar políticas de segurança e acesso regularmente.
- Testar a capacidade de resposta da infraestrutura em cenários de alta carga.

5. Extras

Prioridades (80% do impacto):

- Capacitar a equipe para identificar e responder a incidentes de segurança.
- Implementar redundância para serviços críticos, garantindo alta disponibilidade.
- Configurar backups regulares e testar a recuperação de dados.

Ações Complementares (20% do impacto):

- Garantir que sub-redes estejam devidamente isoladas para minimizar o impacto de incidentes.
- Realizar testes de penetração para identificar vulnerabilidades exploráveis.
- Contratar especialistas externos para avaliações independentes.

Este plano de ação prioriza as atividades que geram maior impacto na segurança e eficiência da infraestrutura de rede, enquanto mantém ações complementares para garantir melhorias contínuas. A abordagem 80/20 permite concentrar esforços nas áreas mais críticas, assegurando que os recursos sejam utilizados de forma estratégica e eficaz. A implementação dessas etapas fortalecerá a proteção, estabilidade e desempenho da infraestrutura, promovendo uma gestão proativa e resiliente.

Conclusão

Este projeto demonstrou uma abordagem estruturada e detalhada para mapear redes, identificar ativos, serviços e vulnerabilidades, e organizar os resultados para análise. A metodologia aplicada garantiu uma coleta eficiente de informações, utilizando ferramentas avançadas como Nmap, Rustscan, e Zabbix, além de técnicas complementares para análise de serviços e organização de dados.

Os resultados obtidos proporcionaram uma visão clara da infraestrutura de rede, destacando dispositivos críticos, serviços expostos e possíveis vulnerabilidades. A organização dos dados em inventários e tabelas descritivas facilitou a interpretação e documentação, enquanto o diagnóstico detalhado serviu como base para recomendações práticas e um plano de ação estratégico.

Com as recomendações e o plano de ação proposto, é possível fortalecer a segurança, monitoramento e gestão da infraestrutura, garantindo maior estabilidade, eficiência e conformidade com boas práticas e regulamentações. Este projeto não apenas mapeou a rede, mas também estabeleceu uma base sólida para melhorias contínuas e proteção proativa da infraestrutura.

Referência Bibliográfica

Ferramentas e Documentação Técnica:

1. Nmap Documentation. Disponível em: <https://nmap.org/book/>.
2. Wireshark User Guide. Disponível em: <https://www.wireshark.org/docs/>.
3. Zabbix Documentation. Disponível em: <https://www.zabbix.com/documentation>.
4. OpenVAS Documentation. Disponível em: <https://www.openvas.org/documentation.html>.
5. Nessus Documentation. Disponível em: <https://docs.tenable.com/nessus/>.
6. Prometheus Documentation. Disponível em: <https://prometheus.io/docs/>.
7. ELK Stack Documentation. Disponível em: <https://www.elastic.co/guide/index.html>.
8. Splunk Documentation. Disponível em: <https://docs.splunk.com/>.

Normas e Regulamentações:

1. ISO/IEC 27001:2013 - Information Security Management Systems. Disponível em: <https://www.iso.org/standard/54534.html>.
2. GDPR - General Data Protection Regulation. Disponível em: <https://gdpr-info.eu/>.
3. LGPD - Lei Geral de Proteção de Dados (Brasil). Disponível em: <https://www.gov.br/lgpd>.
4. PCI DSS - Payment Card Industry Data Security Standard. Disponível em: <https://www.pcisecuritystandards.org/>.
- 5.

Recursos Educacionais e Tutoriais:

1. OWASP Foundation. Disponível em: <https://owasp.org/>.
2. Cybersecurity & Infrastructure Security Agency (CISA). Disponível em: <https://www.cisa.gov/>.
3. MITRE ATT&CK Framework. Disponível em: <https://attack.mitre.org/>.

Recursos de Ferramentas de Consulta:

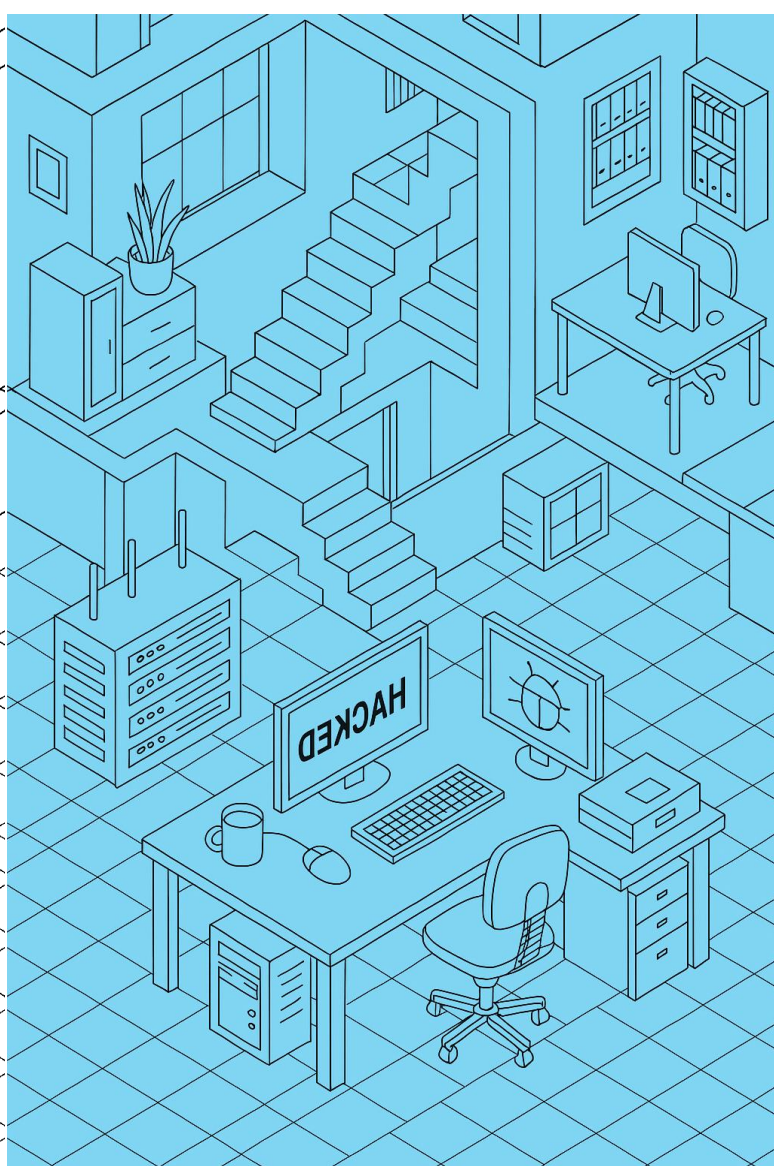
1. ChatGPT - Desenvolvido pela OpenAI, o ChatGPT foi utilizado para consultas rápidas e esclarecimento de dúvidas técnicas, auxiliando na compreensão de conceitos e na elaboração de estratégias para o projeto. Disponível em: <https://openai.com/chatgpt>.
2. GeminiAI - Criado pelo Google DeepMind, o GeminiAI foi empregado para consultas avançadas e suporte em análises complexas, contribuindo para a tomada de decisões no desenvolvimento do projeto. Disponível em: <https://www.deepmind.com>.
3. GitHub Copilot Desenvolvido pela GitHub em parceria com a OpenAI, o Copilot foi utilizado para sugerir trechos de código e soluções práticas, otimizando o tempo de desenvolvimento e garantindo maior eficiência no projeto. Disponível em: <https://github.com/features/copilot>.

Materiais Adicionais:

1. Material do curso Formação Cibersec. Disponível em: <https://escolavainaweb-com.gitbook.io/formacao-cibersec>.
2. Documentação do projeto final. Disponível em: <https://drive.google.com/file/d/1yT8bNuMP29qpE0YII3ZK26DEOiocUsRK/view>.
3. Repositório do projeto final módulo 01. Disponível em: https://github.com/Kensei-CyberSec-Lab/formacao-cybersec/tree/main/modulo1-fundamentos/projeto_final_opcao_1.

Anexos

'ANOTACAO-ULTIMO-SCAN.TXT'	5, 6
corp_net_ips.txt	12, 15, 18, 19
corp_net_ips_hosts.txt	12, 15
corp_net_ips_ports.txt	18, 19
guest_net_ips.txt.....	14, 16, 18, 19
guest_net_ips_hosts.txt.....	16
guest_net_ips_ports.txt	18, 19
infra_net_ips.txt	18, 19
infra_net_ips_hosts.txt	16
infra_net_ips_ports.txt	18, 19
infra_net_servico_ftp-anon.txt	21, 22
infra_net_servico_ldap-rootdse.txt.....	29, 30
infra_net_servico_mysql-info.txt	25, 26
infra_net_servico_smb.txt.....	34, 35, 69
infra_net_servico_webserver.txt.....	39, 40, 42
infra_net_servico_zabbix.txt	43, 44
recon_ip_maps.txt	46, 47
user_processo_smb.txt	36
Figura 1: Print 01 - Comando `ip a`	6
Figura 2: Print 02 - Comando `ip a grep inet`	7
Figura 3: Print 03 - Comando `ping -c 3 10.10.10.1 # corp_net` `ping -c 3 10.10.30.1 # infra_net` `ping -c 3 10.10.50.1 # guest_net`	9
Figura 4: Print 04 - Comandos - Corp Network - Sub-rede: 10.10.10.0/24.	13
Figura 5: Print 05 - Comandos - Infra Network - Sub-rede: 10.10.30.0/24.....	13
Figura 6: Print 06 - Comandos - Guest Network - Sub-rede: 10.10.50.0/24.	14
Figura 7: Print 07 - Comando `ls`	16
Figura 8: Print 08 - Comandos - Rustscan.	18
Figura 9: Print 09 - Comandos `nmap -p 21 --script ftp-anon 10.10.30.10 > infra_net_servico_ftp- anon.txt`	21
Figura 10: Print 10 - Comando `apt update && apt install ftp -y`	23
Figura 11: Print 11 - Comando `ftp 10.10.30.10`	24
Figura 12: Print 12 - Comando `nmap -p 3306 --script mysql-info 10.10.30.11 > infra_net_servico_mysql-info.txt`	25
Figura 13: Print 13 - Comando `nmap -p 389 --script ldap-rootdse 10.10.30.17 > infra_net_servico_ldap-rootdse.txt`	29
Figura 14: Print 14 - Comando `nmap -p 445 --script smb-os-discovery,smb-enum-shares 10.10.30.15> infra_net_servico_smb.txt`	34
Figura 15: Print 15 - Comando `nmap -p 445 --script smb-enum-users,smb-enum-processes 10.10.30.15`	36
Figura 16: Print 16 - Comando `curl -I http://10.10.30.117> infra_net_servico_webserver.txt`	39
Figura 17: Print 17 - Comando `curl http://10.10.30.117> infra_net_servico_zabbix.txt`	43
Figura 18: Print 18 - Comando `arp -a > recon_ip_maps.txt`	46
Figura 19: Diagrama 01 - Diagrama da rede da empresa.	53



kensei.seg.br



vainaweb.com.br

Na Kensei CyberSec Lab, acreditamos que a tecnologia pode ser uma ponte para a transformação social. Por isso, atuamos lado a lado com ONGs, institutos educacionais e projetos comunitários para proteger dados, preservar operações e empoderar pessoas.