



CYBERSECURITY AND INNOVATION FOR
COMPANIES OF THE FUTURE



CIBERSEGURANÇA - MÓDULO I

Alexandre Alves Ferreira

- versão 1 -

Recife, 22 de julho de 2025

SUMÁRIO

1. Sumário Executivo -----	03
2. Objetivos -----	04
2.1 Objetivo principal -----	04
2.2 Objetivos específicos -----	04
3. Escopo -----	05
4. Metodologia -----	05
4.1 ip -----	05
4.2 ping -----	06
4.3 nmap -----	06
4.4 rustscan -----	08
4.4.1 Rede corp_net -----	09
4.4.1.1 IP 10.10.10.1 -----	09
4.4.2 Rede infra_net -----	11
4.4.2.1 IP 10.10.30.1 -----	12
4.4.2.2 IP 10.10.30.10 -----	12
4.4.2.3 IP 10.10.30.11 -----	12
4.4.2.4 IP 10.10.30.15 -----	13
4.4.2.5 IP 10.10.30.17 -----	14
4.4.2.6 IP 10.10.30.117 -----	15
4.4.3 Rede guest_net -----	16
4.4.3.1 IP 10.10.50.1 -----	17
5. Diagrama de redes -----	18
6. Diagnósticos e Recomendações -----	19
6.1 Rede corp_net (10.10.10.0/24) -----	19
6.1.1 IP 10.10.10.1 -----	19
6.2 Rede infra_net (10.10.30.0/24) -----	19
6.2.1 IP 10.10.30.1 -----	19
6.2.2 IP 10.10.30.10 -----	20
6.2.3 IP 10.10.30.11 -----	20
6.2.4 IP 10.10.30.15 -----	20
6.2.5 IP 10.10.30.17 -----	21
6.2.6 IP 10.10.30.117 -----	22
6.3 Rede guest_net (10.10.50.0/24) -----	22
6.3.1 IP 10.10.50.1 -----	22
7. Plano 80/20 -----	23
8. Conclusões -----	24

1. Sumário Executivo

Este relatório apresenta uma análise técnica das redes corp_net, infra_net e guest_net, com foco na identificação de exposições de segurança e elaboração de um plano de ação estratégico baseado no princípio 80/20. A avaliação identificou vulnerabilidades relevantes ligadas a serviços obsoletos, portas abertas desnecessariamente e aplicações desatualizadas, que podem comprometer a segurança do ambiente.

Dentre os principais pontos de atenção, destacam-se: a exposição do serviço rpcbind em múltiplos gateways, o acesso externo ao banco de dados MySQL, a presença de LDAP sem criptografia e com autenticação anônima, além da utilização de versões antigas do Zabbix e do PHP. Também foram encontrados serviços com status unknown, exigindo investigação adicional.

Aplicando o modelo 80/20, constatou-se que a maior parte do risco está concentrada em poucos ativos e serviços. Isso permitiu priorizar as ações que trarão maior impacto com menor esforço. As recomendações imediatas incluem: desativar ou isolar o rpcbind quando não necessário, restringir o acesso ao MySQL e LDAP com uso de criptografia e firewall, atualizar sistemas expostos (Zabbix e PHP), e identificar serviços desconhecidos em portas abertas.

A conclusão geral é que, embora o ambiente apresente vulnerabilidades importantes, a mitigação pode ser conduzida de forma eficaz se houver foco nos itens críticos identificados. O plano de ação estruturado permitirá uma resposta mais ágil e estratégica, reduzindo a superfície de ataque e fortalecendo a postura de segurança da rede.

2. Objetivos

2.1 Objetivo principal

Elaborar uma auditoria técnica da rede corporativa em questão, incluindo seus múltiplos dispositivos e diferentes sub-redes, analisando potenciais riscos e elaborando recomendações para mitigá-los.

2.2 Objetivos específicos

- Levantar a quantidade de hosts existentes;
- Levantar quais serviços estão em execução e quais portas de execução estão abertas;
- Criar um inventário técnico da rede;
- Criar um diagrama de topologia de rede;
- Criar um relatório técnico com diagnósticos e recomendações;

3. Escopo

A análise documentada neste documento se desenvolveu a partir de um ambiente Docker que simula uma rede, segmentada em sub-redes, que, por sua vez, possuem diverços hosts.

5. Metodologia

Foi utilizado um conjunto de ferramentas no terminal Ubuntu, a partir de uma máquina virtual, construída em Docker, que recebeu o nome de Analyst. Os resultados de saída e informações extraídas através dessas ferramentas serão descritas a seguir, nesta seção.

5.1 ip

O comando `ip` é uma ferramenta usada no terminal de sistemas Linux para visualizar e configurar as conexões de rede do computador. Ele substitui o antigo `ifconfig` e faz parte de um conjunto mais moderno chamado *iproute2*. Com o `ip`, é possível verificar informações como endereços IP, ativar ou desativar conexões de rede, criar rotas e ajustar outras configurações importantes. Por ser mais completo e atualizado, o `ip` é hoje a principal ferramenta recomendada para administrar redes em sistemas Linux.

Utilizei o comando "`ip a`" que é uma forma abreviada do comando "`ip address show`", para exibir as informações de endereço IP das interfaces de rede do sistema. Logo após, utilizei o comando "`ip a | grep inet`" para filtrar apenas as linhas de saída que possuísem a palavra "`inet`".



```
root@bba1cc6bdc:~/home/analyst
root@bba1cc6bdc:~/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_l3
        valid_lft forever preferred_lft forever
2: ethw@v1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether c6:48:0e:52:07:17 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.10.30.2/24 brd 10.10.30.255 scope global eth0
        valid_lft forever preferred_lft forever
3: ethw@v2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether a6:1f:ad:59:66:92 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: ethw@v3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether f2:84:7c:2f:78:0f brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.10.30.2/24 brd 10.10.30.255 scope global eth2
        valid_lft forever preferred_lft forever

root@bba1cc6bdc:~/home/analyst
# ip a | grep inet
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host proto kernel_l3
inet 10.10.30.2/24 brd 10.10.30.255 scope global eth0
inet 10.10.30.2/24 brd 10.10.30.255 scope global eth1
inet 10.10.30.2/24 brd 10.10.30.255 scope global eth2
```

Desta forma descobrimos as redes 10.10.50.2/24 (eth0), 10.10.30.2/24 (eth1) e 10.10.10.2/24 (eth 2). Ressaltamos que o ip de rede 127.0.0.1/8 refere-se ao loopback, logo a nós mesmo.

5.2 ping

O comando ping é utilizado para testar a conectividade entre o computador local e outro dispositivo em uma rede, como um servidor ou outro computador. Ele envia pacotes de dados para o destino e mede o tempo que esses pacotes levam para ir e voltar, indicando se o destino está acessível e qual a qualidade da conexão. É uma ferramenta simples, mas muito útil para diagnosticar problemas de rede, como perda de pacotes ou lentidão na comunicação.

Utilizei o comando "ping -c 3" seguido pelo número pelo número de ip da rede para enviar apenas 3 pacotes de teste para o destino especificado.

```
root@kali:~/scrfwtf/~/home/analyst
root@kali:~/scrfwtf/~/home/analyst
# ip a | grep inet
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host proto kernel lo
inet 10.10.10.2/24 brd 10.10.10.255 scope global eth0
inet 10.10.10.2/24 brd 10.10.10.255 scope global eth1
inet 10.10.10.2/24 brd 10.10.10.255 scope global eth2

root@kali:~/scrfwtf/~/home/analyst
# ping -c 3 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=0.065 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=0.100 ms

--- 10.10.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/ndev = 0.065/0.106/0.154/0.036 ms

root@kali:~/scrfwtf/~/home/analyst
# ping -c 3 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=0.182 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=0.104 ms

--- 10.10.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/ndev = 0.033/0.079/0.184/0.033 ms

root@kali:~/scrfwtf/~/home/analyst
# ping -c 3 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=0.009 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=0.039 ms

--- 10.10.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/ndev = 0.009/0.060/0.089/0.021 ms
```

Desta forma, consegui saber que as redes estão ativas pelo tempo de resposta extremamente curto.

5.3 nmap

O comando nmap é uma ferramenta utilizada para varredura e análise de redes, permitindo identificar quais dispositivos estão ativos, quais portas estão abertas e quais serviços estão sendo executados em cada máquina. Ele é amplamente usado em testes de segurança e diagnóstico de redes, ajudando a mapear a estrutura de uma rede e detectar possíveis vulnerabilidades. Com uma linguagem acessível, pode-se dizer que o nmap funciona como um "scanner" que mostra o que está disponível e em funcionamento dentro de uma rede, sendo uma ferramenta essencial para administradores e profissionais da área de TI.

Usei o comando "nmap -sn -T4 10.10.10.0/24 -oG - | grep "Up" (onde o "-sn" indica para não escanear portas, o "-T4" indica a velocidade adequada para esse tipo de escaneamento, o "-oG" gera a saída no formato "grepable" (legível por grep) e "| grep "Up"" para filtrar apenas as linhas de saída que possuísem a palavra "Up"), para escanear a rede 10.10.10.0/24 (corp_net).

```

root@d0a1ecc9ed9c: /home/analyst
Host is up (0.00042s latency).
MAC Address: F2:CC:D3:64:65:22 (Unknown)
Nmap scan report for WS_003.projeto_final_opcao_1_corp_net (10.10.10.127)
Host is up (0.00067s latency).
MAC Address: FA:FB:D1:34:38:E8 (Unknown)
Nmap scan report for WS_004.projeto_final_opcao_1_corp_net (10.10.10.222)
Host is up (0.0030s latency).
MAC Address: 4A:D9:20:8A:1B:10 (Unknown)
Nmap scan report for d0a1ecc9ed9c (10.10.10.2)
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 15.85 seconds

root@d0a1ecc9ed9c: /home/analyst
# nmap -sn -T4 10.10.10.0/24 -o- | 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 (d0a1ecc9ed9c) Status: Up

root@d0a1ecc9ed9c: /home/analyst
# nmap -sn -T4 10.10.10.0/24 -o- | 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@d0a1ecc9ed9c: /home/analyst
# nmap -sn -T4 10.10.10.0/24 -o- | 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 (d0a1ecc9ed9c)

root@d0a1ecc9ed9c: /home/analyst
#

```

Encontrei na rede 10.10.10.0/24 (corp_net) os seguintes ips descritos na tabela.

IP	NOME
10.10.10.1	() - gateway padrão da rede
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	(d0a1ecc9ed9c) - nossa máquina

Utilizei o mesmo comando nmap para a rede 10.10.30.0/24 (infra_net).

```

root@d0a1ecc9ed9c: /home/analyst
root@d0a1ecc9ed9c: /home/analyst
# nmap -sn -T4 10.10.30.0/24 -o- | 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 (d0a1ecc9ed9c) Status: Up

root@d0a1ecc9ed9c: /home/analyst
# nmap -sn -T4 10.10.30.0/24 -o- | awk '/Up$/ {print $2}' | tee corp_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@d0a1ecc9ed9c: /home/analyst
# nmap -sn -T4 10.10.30.0/24 -o- | awk '/Up$/ {print $2, $3}' | tee corp_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 (d0a1ecc9ed9c)

root@d0a1ecc9ed9c: /home/analyst
#

```

Encontrei na rede 10.10.30.0/24 (infra_net) os seguintes ips descritos na tabela.

IP	NOME
10.10.30.1	() - gateway padrão da rede
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)
10.10.30.6	(d0a1ecc9ed9c) - nossa máquina

Utilizando novamente o comando nmap, desta vez para a rede 10.10.50.0/24 (guest_net).

```

root@d0a1ecc9ed9c: /home/analyst
root@d0a1ecc9ed9c:~/home/analyst# nmap -sn -T4 10.10.50.0/24 -oG - | grep "Up"
Host: 10.10.50.1 () Status: Up
Host: 10.10.50.3 (notebook-carlos.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 (laptop-luiz.projeto_final_opcao_1_guest_net) Status: Up
Host: 10.10.50.6 (laptop-vastro.projeto_final_opcao_1_guest_net) Status: Up
Host: 10.10.50.2 (d0a1ecc9ed9c) Status: Up

root@d0a1ecc9ed9c:~/home/analyst# nmap -sn -T4 10.10.50.0/24 -oG - | awk '/Up$/ {print $2}' | tee corp_net_ips.txt
10.10.50.1
10.10.50.3
10.10.50.4
10.10.50.5
10.10.50.6
10.10.50.2

root@d0a1ecc9ed9c:~/home/analyst# nmap -sn -T4 10.10.50.0/24 -oG - | awk '/Up$/ {print $2, $3}' | tee corp_net_ips_hosts.txt
10.10.50.1 ()
10.10.50.3 (notebook-carlos.projeto_final_opcao_1_guest_net)
10.10.50.4 (macbook-aline.projeto_final_opcao_1_guest_net)
10.10.50.5 (laptop-luiz.projeto_final_opcao_1_guest_net)
10.10.50.6 (laptop-vastro.projeto_final_opcao_1_guest_net)
10.10.50.2 (d0a1ecc9ed9c)

root@d0a1ecc9ed9c:~/home/analyst#

```

Encontrei na rede 10.10.50.0/24 (infra_net) os seguintes ips descritos na tabela.

IP	NOME
10.10.50.1	() - gateway padrão da rede
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	(d0a1ecc9ed9c) - nossa máquina

5.4 rustscan

O comando rustscan é uma ferramenta moderna de varredura de portas desenvolvida com foco em desempenho e velocidade. Ele é utilizado para identificar rapidamente quais portas estão abertas em um endereço IP ou rede, funcionando como uma alternativa mais rápida ao tradicional nmap. O rustscan pode ser integrado ao nmap para realizar análises mais detalhadas após a identificação inicial das portas. Em resumo, é uma ferramenta útil para quem precisa mapear dispositivos e serviços ativos em uma rede de forma eficiente e com linguagem acessível para iniciantes e profissionais de TI.

Utilizei o comando rustscan -a <ip da rede> | grep open (onde o “-a” indica que vai ser passado o endereço logo a seguir, “<ip da rede>” é o ip que desejo escanear e “| grep open” filtra apenas as linhas que contém a palavra “open”), para escanear cada ip em cada rede.

5.4.1 Rede corp_net

Iniciando com os ips da rede 10.10.10.0/24 (corp_net), os resultados seguem na imagem e tabela abaixo.

```
root@d0a1ecc9ed9c: /home/analyst
(root@d0a1ecc9ed9c)-[/home/analyst]
# rustscan -a 10.10.10.1 | grep open
Discovered open port 111/tcp on 10.10.10.1
Discovered open port 34017/tcp on 10.10.10.1
111/tcp open rpcbind syn-ack ttl 64
34017/tcp open unknown syn-ack ttl 64
(root@d0a1ecc9ed9c)-[/home/analyst]
```

IP	NOME	PORTAS ABERTAS
10.10.10.1	() - gateway padrão da rede	111/tcp rpcbind 44205/tcp unknown
10.10.10.10	(WS_001.projeto_final_opcao_1_corp_net)	Sem portas abertas
10.10.10.101	(WS_002.projeto_final_opcao_1_corp_net)	Sem portas abertas
10.10.10.127	(WS_003.projeto_final_opcao_1_corp_net)	Sem portas abertas
10.10.10.222	(WS_004.projeto_final_opcao_1_corp_net)	Sem portas abertas
10.10.10.2	(d0a1ecc9ed9c) - nossa máquina	Sem portas abertas

5.4.1.1 IP 10.10.10.1

Investigando a porta tcp/111 rcpcbind, utilizei o comando “nmap -p 111 --script=rpcinfo 10.10.10.1” para investigar quais programas RPC estão disponíveis e quais portas eles usam.

```
root@d0a1ecc9ed9c: /home/analyst
(root@d0a1ecc9ed9c)-[/home/analyst]
# nmap -p 111 --script=rpcinfo 10.10.10.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-19 21:18 UTC
Nmap scan report for 10.10.10.1
Host is up (0.000067s latency).

PORT      STATE SERVICE
111/tcp   open  rpcbind
rpcinfo:
  program version  port/proto  service
  100000  2,3,4      111/tcp    rpcbind
  100000  2,3,4      111/udp    rpcbind
  100000  3,4        111/tcp6   rpcbind
  100000  3,4        111/udp6   rpcbind
  100024  1          34017/tcp  status
  100024  1          40454/udp6 status
  100024  1          42917/tcp6 status
  100024  1          44272/udp  status
MAC Address: A2:7F:08:A9:B2:06 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 13.29 seconds
(root@d0a1ecc9ed9c)-[/home/analyst]
```

Ainda na porta tcp/111 rpcbind, utilizei “nmap -sV -p 111 10.10.10.1” para identificar a versão do serviço RPC.

```
root@dDataced9c:/home/analyst
(root@dDataced9c)-[/home/analyst]
# nmap -sV -p 111 10.10.10.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-19 21:22 UTC
Nmap scan report for 10.10.10.1
Host is up (0.00014s latency).

PORT      STATE SERVICE VERSION
111/tcp   open  rpcbind 2-4 (RPC #100000)
MAC Address: A2:7F:08:A9:82:06 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.63 seconds
(root@dDataced9c)-[/home/analyst]
```

Por último, sobre a porta tcp/111 rpcbind, usei “nmap -p 111 --script=nfs* 10.10.10.1” para investigar se há compartilhamentos NFS abertos.

```
root@dDataced9c:/home/analyst
(root@dDataced9c)-[/home/analyst]
# nmap -p 111 --script=nfs* 10.10.10.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-19 21:26 UTC
Nmap scan report for 10.10.10.1
Host is up (0.00011s latency).

PORT      STATE SERVICE
111/tcp   open  rpcbind
MAC Address: A2:7F:08:A9:82:06 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 13.23 seconds
(root@dDataced9c)-[/home/analyst]
```

Para investigar a porta 44205/tcp unknown, utilizei o comando “nmap -sV -p 44205 10.10.10.1” para investigar qual serviço está rodando nesta porta, já que aparecia como desconhecido.

```
root@dDataced9c:/home/analyst
(root@dDataced9c)-[/home/analyst]
# nmap -sV -p 44205 10.10.10.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-19 21:43 UTC
Nmap scan report for 10.10.10.1
Host is up (0.000098s latency).

PORT      STATE SERVICE VERSION
44205/tcp  closed unknown
MAC Address: A2:7F:08:A9:82:06 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.64 seconds
(root@dDataced9c)-[/home/analyst]
```

Logo após, para a porta 44205/tcp unknown, utilizei “nmap -sC -p 44205 10.10.10.1” ainda para tentar descobrir o serviço que estava utilizando a porta, utilizando os scripts do nmap, mas obtive a mesma saída da imagem acima.

Por fim, para a porta 44205/tcp unknown, utilizei curl <http://10.10.10.1:44205> para investigar se havia algum serviço http rodando na porta mas não houve saída positiva.

5.4.2 Rede infra_net

Passamos a escanear os ips da rede 10.10.30.0/24 (infra_net), os resultados seguem na imagem e tabela abaixo.

```

root@d0a1ecc9ed9c: /home/analyst
--(root@d0a1ecc9ed9c: /home/analyst)
# nmap -sT 10.10.30.1 | grep open
Discovered open port 111/tcp on 10.10.30.1
Discovered open port 44205/tcp on 10.10.30.1
111/tcp open  rpcbind syn-ack ttl 64
44205/tcp open unknown syn-ack ttl 64

--(root@d0a1ecc9ed9c: /home/analyst)
# nmap -sT 10.10.30.10 | grep open
Discovered open port 21/tcp on 10.10.30.10
21/tcp open  ftp syn-ack ttl 64

--(root@d0a1ecc9ed9c: /home/analyst)
# nmap -sT 10.10.30.11 | grep open
Discovered open port 3306/tcp on 10.10.30.11
Discovered open port 33060/tcp on 10.10.30.11
3306/tcp open  mysql syn-ack ttl 64
33060/tcp open mysqlx syn-ack ttl 64

--(root@d0a1ecc9ed9c: /home/analyst)
# nmap -sT 10.10.30.15 | grep open
Discovered open port 445/tcp on 10.10.30.15
Discovered open port 139/tcp on 10.10.30.15
139/tcp open  netbios-ssn syn-ack ttl 64
445/tcp open  microsoft-ds syn-ack ttl 64

--(root@d0a1ecc9ed9c: /home/analyst)
# nmap -sT 10.10.30.17 | grep open
Scanning openldap.projeto_final_opcao_1_infra_net (10.10.30.17) [2 ports]
Discovered open port 389/tcp on 10.10.30.17
Discovered open port 636/tcp on 10.10.30.17
Nmap scan report for openldap.projeto_final_opcao_1_infra_net (10.10.30.17)
389/tcp open  ldap syn-ack ttl 64
636/tcp open  ldaps syn-ack ttl 64

--(root@d0a1ecc9ed9c: /home/analyst)
# nmap -sT 10.10.30.117 | grep open
Discovered open port 80/tcp on 10.10.30.117
Discovered open port 10052/tcp on 10.10.30.117
Discovered open port 10051/tcp on 10.10.30.117
80/tcp open  http syn-ack ttl 64
10051/tcp open zabbix-trapper syn-ack ttl 64

```

IP	NOME	PORTAS ABERTAS
10.10.30.1	() - gateway padrão da rede	111/tcp rpcbind 44205/tcp unknown
10.10.30.10	(ftp-server.projeto_final_opcao_1_infra_net)	21/tcp ftp
10.10.30.11	(mysql-server.projeto_final_opcao_1_infra_net)	3306/tcp mysql 33060/tcp mysqlx
10.10.30.15	(samba-server.projeto_final_opcao_1_infra_net)	139/tcp netbios-ssn 445/tcp microsoft-ds
10.10.30.17	(openldap.projeto_final_opcao_1_infra_net)	389/tcp ldap 636/tcp ldaps
10.10.30.117	(zabbix-server.projeto_final_opcao_1_infra_net)	80/tcp http 10051/tcp zabbix-trapper 10052/tcp unknown
10.10.30.227	(legacy-server.projeto_final_opcao_1_infra_net)	Sem portas abertas
10.10.30.6	(d0a1ecc9ed9c) - nossa máquina	Sem portas abertas

5.4.2.1 IP 10.10.30.1

Para o ip em tela, os resultados das investigações foram iguais do ip 10.10.10.1 da rede corp_net.

5.4.2.2 IP 10.10.30.10

Investigando a porta 21/tcp ftp, utilizei o comando "nmap -p 21 --script ftp-anon 10.10.30.10" para verificar se a porta permite usuário anônimo, mas a saída não sugeriu isto.



```
root@dDetecteDc:/home/analyst
root@dDetecteDc:~# nmap -p 21 --script ftp-anon 10.10.30.10
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-22 20:07 UTC
Nmap scan report for ftp-server.projeto_final_opcao_1_infra_net (10.10.30.10)
Host is up (0.00013s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
MAC Address: 42:D5:EA:4E:89:29 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.51 seconds
root@dDetecteDc:~#
```

5.4.2.3 IP 10.10.30.11

Investigando a porta 3306/tcp mysql, utilizei o comando "nmap -p 3306 --script mysql-info 10.10.30.11" para verificar se a porta estava rodando um serviço mysql. A saída confirmou o serviço na porta.



```
root@dDetecteDc:/home/analyst
root@dDetecteDc:~# nmap -p 3306 --script mysql-info 10.10.30.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-19 22:04 UTC
Nmap scan report for mysql-server.projeto_final_opcao_1_infra_net (10.10.30.11)
Host is up (0.00017s latency).

PORT      STATE SERVICE
3306/tcp  open  mysql
mysql-info:
  Protocol: 10
  Version: 8.0.42
  Thread ID: 11
  Capabilities flags: 65535
  Some Capabilities: IgnoreSigpipes, COCClient, ConnectToDatabase, Support41Auth, SupportsLoadDataLocal, LongPassword, LongColumnFlag, Speaks41ProtocolOld, SupportsTr
  actions, Speaks41ProtocolNew, SwitchToSSLAfterHandshake, DontAllowDatabaseTableColumn, IgnoreSpaceBeforeParenthesis, InteractiveClient, FoundRows, SupportsCompression, Su
  portsMultipleStatements, SupportsMultipleresults, SupportsAuthPlugins
  Status: Autocommit
  Salt: M\w85CHCv15kg1x865+1\w8001\w10*P\w14\
  Auth Plugin Name: caching_sha2_password
  MAC Address: 4E:D1:A7:63:C6:86 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
root@dDetecteDc:~#
```

Utilizei "nmap -p 3306 --script mysql-brute --script-args userdb=users.txt,passdb=senhas.txt 10.10.30.11" para verificar usuários com logins e senhas padrão. O arquivo users.txt (lista de usuários) não foi encontrado, impossibilitando a confirmação.

```
root@dDetective01c:/home/analyst
root@dDetective01c:/home/analyst
nmap -p 3306 --script mysql-brute --script-args userdb=users.txt,passdb=senha.txt 10.10.30.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-19 22:05 UTC
Nmap scan report for mysql-server.projeto_final_opcao_1_infra_net (10.10.30.11)
Host is up (0.000005s latency).

PORT      STATE SERVICE
3306/tcp  open  mysql
|_ mysql-brute: Invalid usernames iterator: Error parsing username list: users.txt: No such file or directory
MAC Address: 4E:D1:A7:83:C6:86 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
root@dDetective01c:/home/analyst
```

Investigando a porta 3306/tcp mysql, utilizei o comando "nmap -p 3306 10.10.30.11" para verificar o estado da porta.

```
root@dDetective01c:/home/analyst
root@dDetective01c:/home/analyst
nmap -p 3306 10.10.30.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-22 20:16 UTC
Nmap scan report for mysql-server.projeto_final_opcao_1_infra_net (10.10.30.11)
Host is up (0.000075s latency).

PORT      STATE SERVICE
3306/tcp  open  mysqlx
MAC Address: 02:4E:9C:27:C8:87 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
root@dDetective01c:/home/analyst
```

5.4.2.4 IP 10.10.30.15

Investigando a porta 445/tcp microsoft-ds, utilizei o comando "nmap -p 445 --script smb-os-discovery,smb-enum-shares 10.10.30.15" para interagir com um servidor SMB (compartilhamento de arquivos do Windows) coletando informações sobre o sistema operacional remoto (nome da máquina, domínio, versão do Windows etc.) e a lista de compartilhamentos SMB (pastas de rede) disponíveis pública ou anonimamente.

```
root@dDetective01c:/home/analyst
root@dDetective01c:/home/analyst
nmap -p 445 --script smb-os-discovery,smb-enum-shares 10.10.30.15
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-19 22:06 UTC
Nmap scan report for samba-server.projeto_final_opcao_1_infra_net (10.10.30.15)
Host is up (0.00022s latency).

PORT      STATE SERVICE
445/tcp  open  microsoft-ds
MAC Address: 22:D7:25:BD:A5:61 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
root@dDetective01c:/home/analyst
```

Investigando a porta 139/tcp netbios-ssn, utilizei o comando "nmap -p139,445 --script smb-protocols 10.10.30.15" para verificar quais versões SMB (v1, v2, v3) estão habilitadas, pois o SMBv1 é inseguro e deve ser desativado.

```
root@d8a1ecb0d8c:/home/analyst
root@d8a1ecb0d8c:~/home/analyst
# nmap -p339,445 --script smb-protocols 10.10.30.15
Starting Nmap 7.95 ( https://nmap.org ) at 2023-07-22 22:36 UTC
Nmap scan report for samba-server.projeto_final_opcao_1_infra_net (10.10.30.15)
Host is up (0.000006s latency).

PORT      STATE SERVICE
339/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: DE:D4:A8:6A:C4:85 (Unknown)

Host script results:
|_ smb-protocols:
|   dialects:
|       2.1.0
|       3.0.0
|       3.0.2
|       3.1.1
|_
Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds

root@d8a1ecb0d8c:~/home/analyst
```

5.4.2.5 IP 10.10.30.17

Investigando a porta 389/tcp ldap, utilizei o comando "nmap -p 389 --script ldap-rootdse 10.10.30.17" para para interrogar o servidor LDAP e coletar informações básicas sobre ele.

```
root@d8a1ecb0d8c:/home/analyst
# nmap -p 389 --script ldap-rootdse 10.10.30.17
Starting Nmap 7.95 ( https://nmap.org ) at 2023-07-22 22:47 UTC
Nmap scan report for openldap.projeto_final_opcao_1_infra_net (10.10.30.17)
Host is up (0.00021s 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: GSS-IAKRB8
|       supportedSASLMechanisms: GSS-KRB5
|       supportedSASLMechanisms: SCRAM-SHA-1
|       supportedSASLMechanisms: SCRAM-SHA-256
|       supportedSASLMechanisms: GSSAPI
|       supportedSASLMechanisms: GSS-SPNEGO
|       supportedSASLMechanisms: DIGEST-MD5
|       supportedSASLMechanisms: OTP
|       supportedSASLMechanisms: NTLM
|       supportedSASLMechanisms: CRAM-MD5
|       subSchemaSubentry: cn=Subschema
|_
MAC Address: BA:85:E3:CB:D9:7D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds

root@d8a1ecb0d8c:~/home/analyst
```

Investigando a porta 636/tcp ldapssl, utilizei o comando "openssl s_client -connect 10.10.30.17:636 -showcerts" para verificar o certificado SSL da porta 636. Os resultados serão demonstrados no capítulo de diagnósticos e recomendações.

Ainda investigando a porta 636/tcp ldapssl, utilizei o comando "nmap -p 636 --script ssl-cert,ssl-enum-ciphers 10.10.30.17 para verificar vulnerabilidades com Nmap NSE (scripts de segurança). A saída demonstrou boas práticas que serão discutidas no capítulo de Diagnósticos e Recomendações.

```
root@kali:~/Documents# cd /home/analyst/
[analyst@kali ~]$ root@d81acc9ed9c:~/home/analyst/
[analyst@kali ~]$ nmap -p 636 -sC --script ssl-cert,ssl-enum-ciphers 10.10.10.17
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-22 22:58 UTC
Nmap scan report for openldap.projets_final_opcao_1_infra_net (10.10.10.17)
Host is up (0.000077s latency).

PORT      STATE SERVICE
636/tcp   open  ldapssl
|_ ssl-enum-ciphers:
|_   TLSv1.2:
|_     ciphers:
|_       TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|_       TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|_       TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|_       TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (secp256r1) - A
|_       TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA256 (secp256r1) - A
|_       TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (secp256r1) - A
|_       TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (secp256r1) - A
|_     compressors:
|_       NULL
|_     cipher preference: client
|_     warnings:
|_       Key exchange (secp256r1) of lower strength than certificate key
|_       least strength: A
MAC Address: 4E:2C:EA:CA:8C:8E (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
[analyst@kali ~]$ root@d81acc9ed9c:~/home/analyst/
```

5.4.2.6 IP 10.10.30.117

Investigando a porta 80/tcp http, utilizei o comando "curl -I http://10.10.30.117" para enviar uma requisição HTTP do tipo HEAD para o servidor no IP 10.10.30.117 e exibir apenas os cabeçalhos (headers) na resposta.

```
root@kali:~/Documents/analyst# curl -I http://10.10.10.10:80
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 19 Jul 2025 22:08:19 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Keep-Alive: timeout=10
X-Powered-By: PHP/7.3.14
Set-Cookie: PHPSESSID=a48b20894a4863489951773f949c1ee4; 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
```

Ainda na porta 80/tcp http, utilizei o comando "curl http://10.10.30.117" para visualizar a página completa, não apenas o cabeçalho.

[illegible]

Investigando a porta 10051/tcp zabbix-trapper, utilizei o comando "nmap -sV -p 10051 --script "default or safe or vuln" 10.10.30.117" para procurar por vulnerabilidades. Os resultados serão demonstrados no capítulo de Diagnósticos e Recomendações.

Investigando a porta 10052/tcp unknown, utilizei o comando "nmap -sV -p 10052 --version-intensity 9 10.10.30.117" para descobrir qual serviço pode estar rodando na porta. É uma detecção mais profunda, usando opções extras para tentar identificar o serviço.

5.4.3 Rede guest_net

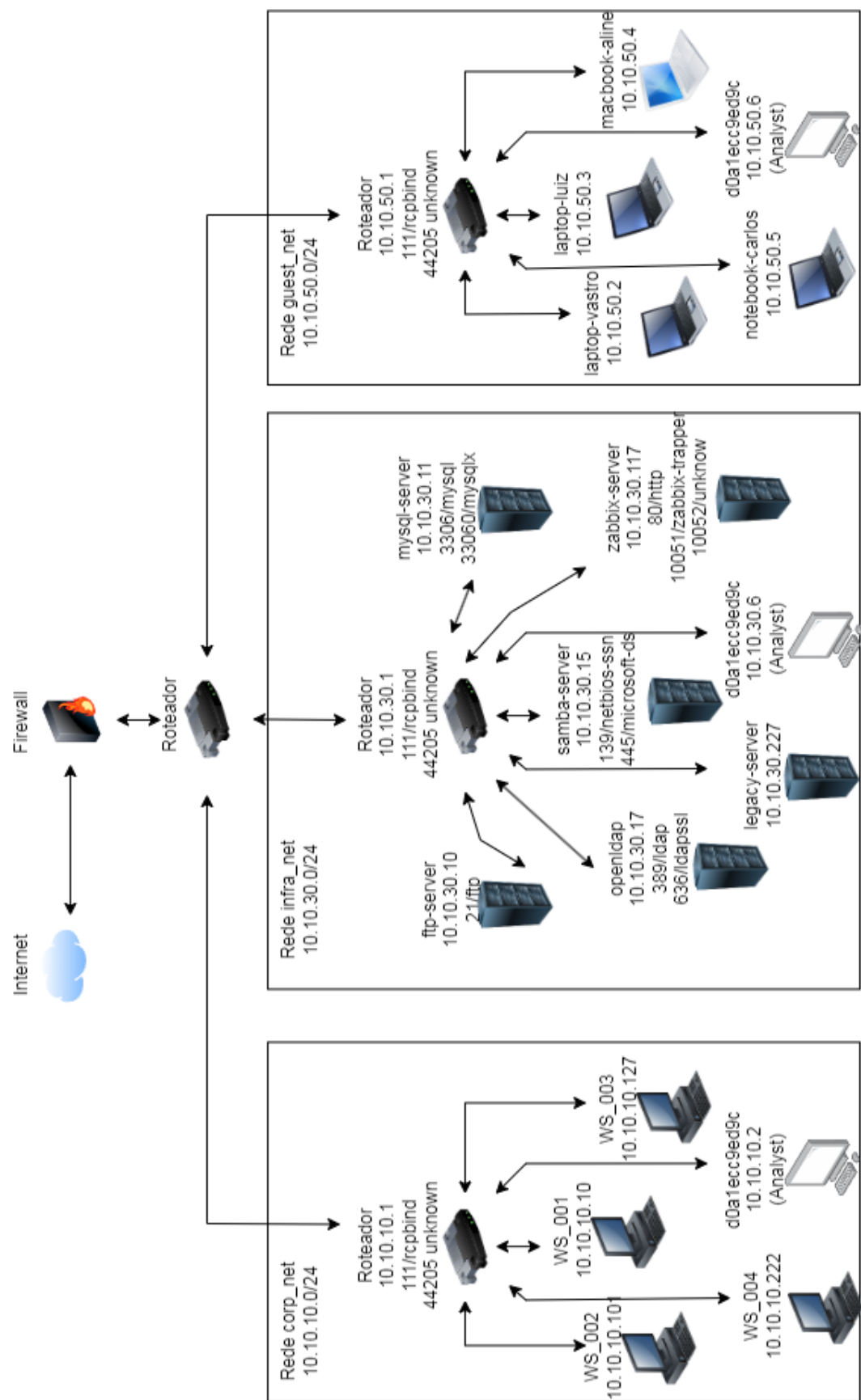
Passamos a escanear os ips da rede 10.10.50.0/24 (guest_net), os resultados seguem na imagem e tabela abaixo.

IP	NOME	PORTAS ABERTAS
10.10.50.1	() - gateway padrão da rede	111/tcp rpcbind 44205/tcp unknown
10.10.50.2	(laptop-vastro.projeto_final_opcao_1_guest_net)	Sem portas abertas
10.10.50.3	(laptop-luiz.projeto_final_opcao_1_guest_net)	Sem portas abertas
10.10.50.4	(macbook-aline.projeto_final_opcao_1_guest_net)	Sem portas abertas
10.10.50.5	(notebook-carlos.projeto_final_opcao_1_guest_net)	Sem portas abertas
10.10.50.6	(d0a1ecc9ed9c) - nossa máquina	Sem portas abertas

5.4.3.1 IP 10.10.50.1

Para o ip em tela, os resultados das investigações foram iguais do ip 10.10.10.1 da rede corp_net.

6. Diagrama de Rede



7. Diagnósticos e Recomendações

7.1 Rede corp_net (10.10.10.0/24)

7.1.1 IP 10.10.10.1

Este ip está relacionado com o gateway (roteador) da rede corp_net e possui as portas 111/rpcbind e 44205/unknown abertas. O serviço rpcbind (também conhecido como portmapper) é usado em sistemas Unix/Linux para mapear chamadas de procedimentos remotos (RPCs) para as portas corretas onde os serviços estão escutando. Um serviço "unknown" (desconhecido) geralmente aparece em varreduras de rede, como as feitas com o Nmap, quando uma porta está aberta, mas o scanner não conseguiu identificar qual serviço está rodando nela.

A versão 2-4 do RPC (Remote Procedure Call), que é o utilizado no sistema, que aparece geralmente como suporte a versões 2, 3 e 4 no serviço rpcbind (porta 111/tcp), não é considerada atual nem segura nos padrões modernos. Essas versões são protocolos antigos, desenvolvidas nos anos 90 e início dos 2000, usados principalmente em sistemas Unix/Linux legados.

O modelo RPC clássico (usando rpcbind) está praticamente obsoleto em sistemas modernos, que preferem alternativas mais seguras (como gRPC ou protocolos RESTful). Somando-se o fato do rpcbind ser de versões antigas, não têm criptografia, não fazem autenticação adequada e são vulneráveis a ataques de spoofing e DoS (negação de serviço).

O rpcbind é conhecido por ter várias vulnerabilidades exploráveis se exposto à internet, podendo ser usado por invasores para descobrir e interagir com outros serviços RPC vulneráveis. Se a corporação não precisa do serviço RPC (por exemplo, não usa sistemas de compartilhamento como NFS ou NIS), é recomendado desabilitar o rpcbind. Se precisar manter por exigência do ambiente, certifique-se de restringir o acesso via firewall (ex: apenas para IPs internos confiáveis), monitorar conexões e atualizar constantemente o sistema.

Quanto a porta 44205/unknown, não foi possível estabelecer com certeza a natureza do serviço desconhecido, exigindo maior investigação posterior.

7.2 Rede infra_net (10.10.30.0/24)

7.2.1 IP 10.10.30.1

Este ip está relacionado com o gateway (roteador) da rede infra_net e possui as portas 111/rpcbind e 44205/unknown abertas. Os diagnósticos e recomendações evidenciados no IP 10.10.10.1 da corp_net, podem ser atribuídas para esse ip.

7.2.2 IP 10.10.30.10

Este IP possui a porta 21/ftp aberta. FTP (File Transfer Protocol) é um protocolo padrão da internet usado para transferência de arquivos entre computadores, geralmente entre um cliente e um servidor.

Tentei fazer conexão como um usuário anônimo, mas não obtive êxito, mostrando que a porta parece estar segura. Sendo assim, por hora não há nenhuma recomendação.

7.2.3 IP 10.10.30.11

Este IP possui as portas 3306/mysql e 33060/mysqlx abertas. O serviço mysql é serviço principal do banco de dados MySQL, ele escuta na porta 3306/tcp e permite que clientes se conectem ao banco para executar comandos SQL. O mysqlx é um plugin opcional introduzido a partir do MySQL 5.7+ (especialmente no MySQL 8) que habilita um novo protocolo de comunicação, chamado MySQLX.

Um servidor MySQL está rodando na porta 3306, que está aberta e acessível externamente e pode representar risco de segurança, especialmente se não houver firewall ou controle de IPs. O servidor respondeu com informações detalhadas (versão, plugin de autenticação, recursos). A versão 8.0.42 é recente e possui várias correções de segurança, mas divulgar a versão exata pode permitir que um atacante busque vulnerabilidades específicas (se houver exploits públicos). Ele usa o plugin de autenticação caching_sha2_password, padrão em versões recentes, mas se não houver SSL/TLS (protocolos de segurança que garantem a criptografia de dados transmitidos entre dois pontos) habilitado, as credenciais ainda podem ser capturadas na rede, especialmente durante o fallback para autenticação sem criptografia. A porta responde à sondagem do Nmap, o que revela que não há camadas de ofuscação ou bloqueio de fingerprint. O servidor suporta muitos recursos, isso pode aumentar a superfície de ataque se o servidor aceitar conexões de qualquer lugar. Está online e possivelmente acessível para testes futuros (como brute force, enumeração, etc). Foi testado para verificar lista de usuários com login e senhas padrões mas não retornou nenhum resultado.

Para a porta 33060/mysqlx, o serviço deve ser protegido por firewall ou autenticação forte. O MySQLX deve estar restrito a IPs confiáveis, preferencialmente acessível apenas localmente (127.0.0.1), a menos que o acesso remoto seja realmente necessário.

7.2.4 IP 10.10.30.15

Este IP possui as portas 139/netbios-ssn e 445/microsoft-ds abertas. Microsoft-ds é o nome do serviço associado à porta TCP 445, usado por sistemas Windows (e compatíveis) para compartilhamento de arquivos e impressoras através do protocolo SMB (Server Message Block). O netbios-ssn é o nome do serviço associado à porta TCP 139, usado para compartilhamento de arquivos e impressoras em redes Windows

antigas, via o protocolo NetBIOS sobre TCP/IP (também chamado de NetBIOS Session Service).

Uma investigação foi feita na porta 445 e verificou-se que ela está aberta, mas nenhuma informação extra foi retornada pelos scripts. Isso pode ter acontecido por que o host pode bloquear informações SMB para acessos anônimos, configuração de firewall ou por que algumas versões modernas do Windows e servidores Linux com Samba configurado corretamente não revelam informações sensíveis anonimamente. Qualquer que seja, esses itens indicam um bom nível de segurança, restando apenas assegurar a não visibilidade da porta para acessos externos.

Quanto a porta 139/netbios-ssn, SMBv1 não está presente, o que é bom, pois é uma versão obsoleta e insegura (vulnerável ao WannaCry e EternalBlue). O servidor suporta SMBv2.1 e SMBv3.x, que são versões mais modernas e seguras. O suporte ao SMB 3.1.1 (a versão mais atual) é um indicador positivo de segurança. Outra vez resta apenas assegurar a não visibilidade da porta para acessos externos.

7.2.5 IP 10.10.30.17

Este IP possui as portas 389/ldap e 636/ldapsl abertas. O LDAP (Lightweight Directory Access Protocol) é uma espécie de "catálogo de rede" que serve para autenticação e organização de dados em ambientes corporativos. Já o LDAPS é a versão segura do protocolo LDAP, que transmite os dados criptografados por meio da porta 636/tcp.

Na porta 389/ldap podemos dizer que ela está aberta e sem criptografia, que é um padrão LDAP, o que denota um risco moderado a alto. Isso acontece porque o LDAP transmite os dados em texto claro, incluindo usuários, senhas e atributos, a menos que seja usado STARTTLS (comando de extensão para iniciar uma sessão criptografada - TLS), o que não está indicado nesta saída. Em redes inseguras (como a internet), isso é um risco sério de interceptação (sniffing), em redes internas, o risco é menor, mas ainda exige atenção. A consulta foi bem-sucedida sem autenticação, o que significa que o servidor LDAP responde a requisições anônimas, deve-se alterar para que apenas usuários autenticados possam consultar diretórios sensíveis. Sugiro desabilitar mecanismos fracos como CRAM-MD5, DIGEST-MD5 e NTLM se não forem necessários. Por fim, restringir IPs com firewall, não expondo o LDAP na internet sem proteção rígida.

Quanto a porta 636/ldapsl, o está TLS 1.2, que é seguro, ativo e funcionando. O Cipher (algoritmos de criptografia) é forte, moderno e seguro. O Certificado está dentro da validade. Algumas informações poderiam ser tratadas mas por ser um ambiente Docker, já eram esperadas. Desta forma não há recomendações para essa porta, devendo esta ser preferível a utilização da porta 389/ldap.

7.2.6 IP 10.10.30.117

Este IP possui as portas 80/http, 10051/zabbix-trapper e 10052/unknown abertas. HTTP é um tipo de serviço de rede que utiliza o protocolo HTTP (HyperText Transfer Protocol) para enviar e receber dados, principalmente entre navegadores (como Chrome ou Firefox) e servidores web. O serviço zabbix-trapper é um componente do Zabbix Server responsável por receber dados enviados ativamente por hosts monitorados. Ele faz parte do sistema de monitoramento Zabbix, que coleta, armazena e analisa dados de performance e disponibilidade de redes, servidores, aplicações etc. Um serviço "unknown" (desconhecido) geralmente aparece em varreduras de rede, como as feitas com o Nmap, quando uma porta está aberta, mas o scanner não conseguiu identificar qual serviço está rodando nela.

Na porta 80/http foi verificado que o servidor está ativo e usando nginx com PHP 7.3.14. O site é uma página de login do Zabbix, que exige autenticação e envia cookies de sessão. A versão do Zabbix é a 4.4, que já é uma versão antiga e fora de suporte, podendo ser ponto fraco de segurança, especialmente se a instância estiver exposta à internet. Algumas boas práticas de segurança estão em uso, mas a versão do PHP é antiga, sendo também um possível ponto de exploração. Sendo assim, sugiro atualizar a versão do Zabbix verificar a possibilidade de reescrita em linguagem ou versão PHP mais atual para evitar riscos.

Na porta 10051/zabbix-trapper, está aberta e rodando com SSL, ou seja, comunicação criptografada, o que é bom para proteger os dados em trânsito. O IP não está listado em blacklists, indicando que provavelmente o host não está envolvido em atividades maliciosas conhecidas. A configuração DNS está correta. De recomendações apenas que o Zabbix seja atualizado, que o acesso à porta seja restrita via firewall, permitindo somente agentes ou servidores autorizados, e que os certificados SSL usados sejam válidos e fortes.

Na porta 10052/unknown, o Nmap tentou várias técnicas de detecção, e recebeu uma resposta com o prefixo "ZBXD". Esse prefixo "ZBXD" indica que a porta está rodando o protocolo Zabbix (o mesmo do servidor na 10051). Logo, o serviço da porta 10052 provavelmente é Zabbix Proxy ou Zabbix Agent passivo, ou algum componente do Zabbix usando o protocolo Zabbix (ZBXD). Desta forma as recomendações ficam sendo as mesmas da porta 10051 acima.

7.3 Rede guest_net (10.10.50.0/24)

7.3.1 IP 10.10.50.1

Este ip está relacionado com o gateway (roteador) da rede guest_net e possui as portas 111/rpcbind e 44205/unknown abertas. Os diagnósticos e recomendações evidenciados no IP 10.10.10.1 da corp_net, podem ser atribuídas para esse ip.

8. Plano 80/20

Um plano de ação 80/20 é uma estratégia baseada no Princípio de Pareto, que afirma que aproximadamente 80% dos resultados vêm de 20% das causas ou esforços. Aplicado à prática, esse tipo de plano tem como objetivo identificar e priorizar as ações mais impactantes em um determinado cenário, concentrando recursos e esforços nas poucas tarefas que trarão a maior parte dos benefícios ou resolverão a maior parte dos problemas.

Ação	Impacto	Dificuldade	Prioridade
Desabilitar ou isolar o serviço rpcbind nos gateways (10.10.10.1, 10.10.30.1, 10.10.50.1)	Alto	Baixo	Alto
Restringir o acesso externo ao MySQL (3306) e MySQLX (33060) via firewall	Alto	Médio	Alto
Ativar SSL/TLS no MySQL e ocultar banners de versão	Alto	Médio	Alto
Desabilitar acesso anônimo no LDAP (porta 389)	Alto	Médio	Alto
Forçar uso de LDAPS (porta 636) ou STARTTLS no LDAP	Alto	Médio	Alto
Atualizar o Zabbix (versão 4.4) e o PHP (versão 7.3.14)	Alto	Alto	Alto
Restringir acesso à porta 10051 (zabbix-trapper) via firewall	Médio	Baixo	Médio
Investigar os serviços unknown nas portas 44205 e 10052	Médio	Alto	Médio
Verificar se as portas SMB (139/445) estão visíveis externamente e aplicar bloqueio	Médio	Baixo	Médio
Monitorar e revisar segurança no servidor HTTP do Zabbix (porta 80)	Médio	Baixo	Médio
Confirmar que o servidor FTP (porta 21) não permite acessos externos não autorizados	Baixo	Baixo	Baixo

9. Conclusões

Com base na análise das redes corp_net, infra_net e guest_net, foi possível identificar diversos pontos de atenção em termos de segurança, especialmente relacionados a serviços legados, exposições desnecessárias e softwares desatualizados. Aplicando o princípio 80/20, observou-se que a maioria dos riscos está concentrada em poucos elementos recorrentes, como a exposição do serviço rpcbind, o acesso remoto irrestrito a bancos de dados MySQL, o uso de LDAP sem criptografia e autenticação adequada, além da presença de aplicações desatualizadas como o Zabbix 4.4 e PHP 7.3.

Esses serviços representam a superfície de ataque mais relevante e, portanto, devem ser tratados com prioridade máxima. As demais descobertas, embora menos críticas, também merecem atenção complementar para assegurar uma postura de segurança mais robusta e resiliente.

Portanto, a conclusão é que a rede analisada apresenta vulnerabilidades reais, mas com foco e ações corretas sobre os pontos mais críticos — representando apenas uma fração do ambiente — é possível reduzir significativamente os riscos operacionais e de segurança. A implementação de um plano de ação estratégico, priorizando os itens identificados pelo método 80/20, é essencial para fortalecer a defesa do ambiente e garantir maior controle sobre a infraestrutura.

ANEXO I - COMANDOS FEITOS NO ÚLTIMO ESCANEAMENTO DA REDE

Primeiro pegar info das redes

```
ip a
```

```
ip a | grep inet
```

```
ip a | grep inet > recon-redes.txt
```

Testar se tem conectividade com as redes

```
ping -c 3 10.10.10.1 # corp_net
```

```
ping -c 3 10.10.30.1 # guest_net
```

```
ping -c 3 10.10.50.1 # infra_net
```

1. descobrir os hosts com Nmap ping scan

```
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
```

```
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
```

```
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
```

2. Scan rápido com Rustscan para pegar as portas abertas

```
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
```

3. Analisar os serviços específicos

FTP

```
nmap -p 21 --script ftp-anon 10.10.30.10
```

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

MySQL

```
nmap -p 3306 --script mysql-info 10.10.30.11
```

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

LDAP

```
nmap -p 389 --script ldap-rootdse 10.10.30.17
```

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

SMB

```
nmap -p 445 --script smb-os-discovery,smb-enum-shares 10.10.30.15
```

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

```
infra_net_servico_smb.txt
```



```
### HTTP (web)
curl -I http://10.10.30.117
curl -I http://10.10.30.117 > infra_net_servico_webserver.txt
curl http://10.10.30.117
curl http://10.10.30.117 > infra_net_servico_zabbix.txt

## Extras úteis
arp -a
arp -a > recon_ip_maps.txt
cat /etc/resolv.conf

## Organizar os resultados (manter tudo limpinho)
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/

## Copiar depois pro host local - tem que sair do docker e rodar da maquina local
docker cp analyst:/home/analyst/recon ./recon-backup
```

ANEXO II - INVENTÁRIO TÉCNICO

Rede/Sub-rede	Endereço IP	Função / Host	Portas/Serviços Identificados	Observações e Riscos	Recomendações
corp_net (10.10.10.0/24)	10.10.10.1	Gateway da rede	111/rpcbind 44205/unknown	rpcbind expõe versões antigas e vulneráveis; serviço 44205 não identificado.	Desabilitar rpcbind se não utilizado; Restringir acesso via firewall; Investigar porta 44205.
	10.10.10.10 (WS_001)	Estação de trabalho	-	-	-
	10.10.10.101 (WS_002)	Estação de trabalho	-	-	-
	10.10.10.127 (WS_003)	Estação de trabalho	-	-	-
	10.10.10.222 (WS_004)	Estação de trabalho	-	-	-
infra_net (10.10.30.0/24)	10.10.30.1	Gateway da rede	111/rpcbind 44205/unknown	Idêntico ao IP 10.10.10.1	Mesmas recomendações do IP 10.10.10.1
	10.10.30.10	Servidor FTP (possível)	21/ftp	FTP ativo, acesso anônimo desativado	Monitorar acesso; Considerar FTPS/SFTP se dados sensíveis forem transmitidos
	10.10.30.11	Servidor MySQL	3306/mysql 33060/mysqlx	Banco MySQL expõe versão e usa autenticação moderna, mas sem TLS visível. MySQLX ativo.	Restringir acesso externo via firewall; Habilitar TLS; Evitar divulgar versão exata.
	10.10.30.15	Servidor de arquivos SMB	139/netbios-ssn 445/microsoft-ds	Suporte a SMB 2.1/3.x; SMBv1 ausente (positivo). Pouca informação	Garantir não exposição à internet; Revisar regras de acesso.

				retornada (positivo).	
	10.10.30.17	Servidor LDAP/LDAPS	389/ldap 636/ldapsl	LDAP sem criptografia e acesso anônimo; LDAPS usa TLS 1.2 com cipher seguro.	Restringir LDAP a IPs confiáveis; Forçar uso do LDAPS; Desabilitar métodos fracos de auth.
	10.10.30.117	Servidor Zabbix (Web/API)	80/http (nginx + Zabbix 4.4) 10051/zabbix-trapper 10052/unknown (Zabbix protocol)	Zabbix e PHP com versões antigas; HTTP ativo; comunicação SSL ativa em 10051. Porta 10052 é Zabbix.	Atualizar Zabbix e PHP; Restringir acesso às portas 10051/10052; Manter certificados atualizados.
	10.10.30.227	Servidor Legacy	-	-	-
guest_net (10.10.50.0/24)	10.10.50.1	Gateway da rede	111/rpcbind 44205/unknown	Idêntico ao IP 10.10.10.1	Mesmas recomendações do IP 10.10.10.1