



Hackers do Bem – Fundamental
Prof. Fábio Carneiro de Castro
15/02/2026

Atividade Prática – Módulo 3

Aulas 1 e 2

Gabriel dos Santos Schmitz

1 Introdução

Este documento apresenta as evidências práticas das atividades do Módulo 3 (Aulas 1 e 2) do Programa Hackers do Bem – Nível Fundamental, por meio dos prints solicitados, demonstrando a correta execução das tarefas propostas.

De acordo com as orientações do curso, este documento consolida, em um único arquivo PDF, os seguintes registros obrigatórios: Atividade 3.1 (passo 7), Atividade 3.2 (passo 6), Atividade 3.3 (passo 21), Atividade 3.4 (passo 6) e Atividade 3.5 (passo 9). Cada evidência é acompanhada de uma breve descrição, com o objetivo de facilitar a análise e validação pelo instrutor.

2 Atividades

Atividade 3.1. Explorando Exploits Conhecidos com Exploit Database no Kali Linux

```
(root㉿kali)-[~/usr/share/exploitdb]
# searchsploit -t windows 10

Exploit Title | Path
Adobe Flash Player < 10.1.53.64 - Action Script Type Confusion (ASLR + DEP Bypass) | windows/remote/17187.txt
Apple iTunes 8.1.1.10 (Windows) - 'itm5/itcp' Remote Buffer Overflow | windows/remote/8934.py
Autodesk Backburner Manager 3 < 2016.0.0.2150 - Null Dereference Denial of Service | windows/dos/41160.py
Avast Anti-Virus < 19.1.2360 - Local Credentials Disclosure | windows/local/46345.py
Avaya IP Office (IPO) < 10.1 - 'SoftConsole' Remote Buffer Overflow (SEH) | windows/remote/43121.txt
Avaya IP Office (IPO) < 10.1 - ActiveX Buffer Overflow | windows/dos/43120.txt
Cisco WebEx Meetings < 33.6.6 / < 33.9.1 - Privilege Escalation | windows/local/46479.txt
DameWare Remote Controller < 12.0.0.520 - Remote Code Execution | windows/remote/43059.py
EA Origin < 10.5.38 - Remote Code Execution | windows/remote/47019.txt
Ethereal 0.10.9 (Windows) - '3G-A11' Remote Buffer Overflow | windows/dos/874.cpp
Fortinet FortiClient 5.2.3 (Windows 10 x64 Creators) - Local Privilege Escalation | windows/x86-64/local/45149.cpp
Fortinet FortiClient 5.2.3 (Windows 10 x64 Post-Anniversary) - Local Privilege Escalation | windows_x86-64/local/41722.c
Fortinet FortiClient 5.2.3 (Windows 10 x64 Pre-Anniversary) - Local Privilege Escalation | windows_x86-64/local/41721.c
Fortinet FortiClient 5.2.3 (Windows 10 x86) - Local Privilege Escalation | windows_x86/local/41705.cpp
FreeBSD 2.x / HP-UX 9/10/11 / Kernel 2.0.3 / Windows NT 4.0/Server 2003 / NetBSD 1 - 'land. | multiple/dos/20810.c
FreeBSD 2.x / HP-UX 9/10/11 / Kernel 2.0.3 / Windows NT 4.0/Server 2003 / NetBSD 1 - 'land. | multiple/dos/20811.cpp
FreeBSD 2.x / HP-UX 9/10/11 / Kernel 2.0.3 / Windows NT 4.0/Server 2003 / NetBSD 1 - 'land. | multiple/dos/20813.c
FreeBSD 2.x / HP-UX 9/10/11 / Kernel 2.0.3 / Windows NT 4.0/Server 2003 / NetBSD 1 - 'land. | windows/dos/20812.c
FreeBSD 2.x / HP-UX 9/10/11 / Kernel 2.0.3 / Windows NT 4.0/Server 2003 / NetBSD 1 - 'land. | windows/dos/20814.c
Google Chrome < 14.0.835.163 - '.pdf' File Handling Memory Corruption | windows/dos/17929.txt
HP HP-UX 10.34 / Microsoft Windows 95/NT 3.5.1 SP1/NT 3.5.1 SP2/NT 3.5.1 SP3/NT 3.5.1 SP4/N | linux/dos/19103.c
Ipswitch WS_FTP Professional < 12.6.0.3 - Local Buffer Overflow (SEH) | windows/dos/43115.py
Jungo DriverWizard WinDriver < 12.4.0 - Kernel Out-of-Bounds Write Privilege Escalation | windows/local/42625.py
Jungo DriverWizard WinDriver < 12.4.0 - Kernel Pool Overflow / Local Privilege Escalation ( | windows/local/42624.py
Jungo DriverWizard WinDriver < 12.4.0 - Kernel Pool Overflow / Local Privilege Escalation ( | windows/local/42665.py
KiTTY Portable 0.65.0.2p (Windows 8.1/10) - Local kitty.ini Overflow | windows/local/39122.py
```

Fig. 1: Utilização do searchsploit para busca de exploits no Exploit Database

Sobre:

Nesta atividade, foi explorada a base de dados do *Exploit Database* presente no Kali Linux, utilizando a ferramenta `searchsploit` para pesquisa de vulnerabilidades conhecidas. O objetivo foi compreender como identificar exploits disponíveis para diferentes softwares e sistemas, com fins exclusivamente acadêmicos.

Inicialmente, foi realizado acesso ao ambiente Kali Linux via RDP e, em seguida, obtidos privilégios de superusuário com o comando `sudo -i`. Posteriormente, foi verificado o diretório padrão da base de dados do ExploitDB em `/usr/share/exploitdb`, onde estão armazenados exploits, shellcodes e arquivos auxiliares.

A ferramenta `searchsploit` foi utilizada para realizar consultas na base local. Foram observadas suas principais funcionalidades, incluindo:

- Busca por termos relacionados a softwares ou vulnerabilidades;
- Filtros por título, versão e identificadores CVE;

- Exibição de resultados em diferentes formatos, como JSON;
- Acesso direto ao caminho dos exploits armazenados localmente.

Em seguida, a base de dados foi atualizada por meio do comando `searchsploit -u`, garantindo que os resultados refletissem as vulnerabilidades mais recentes disponíveis.

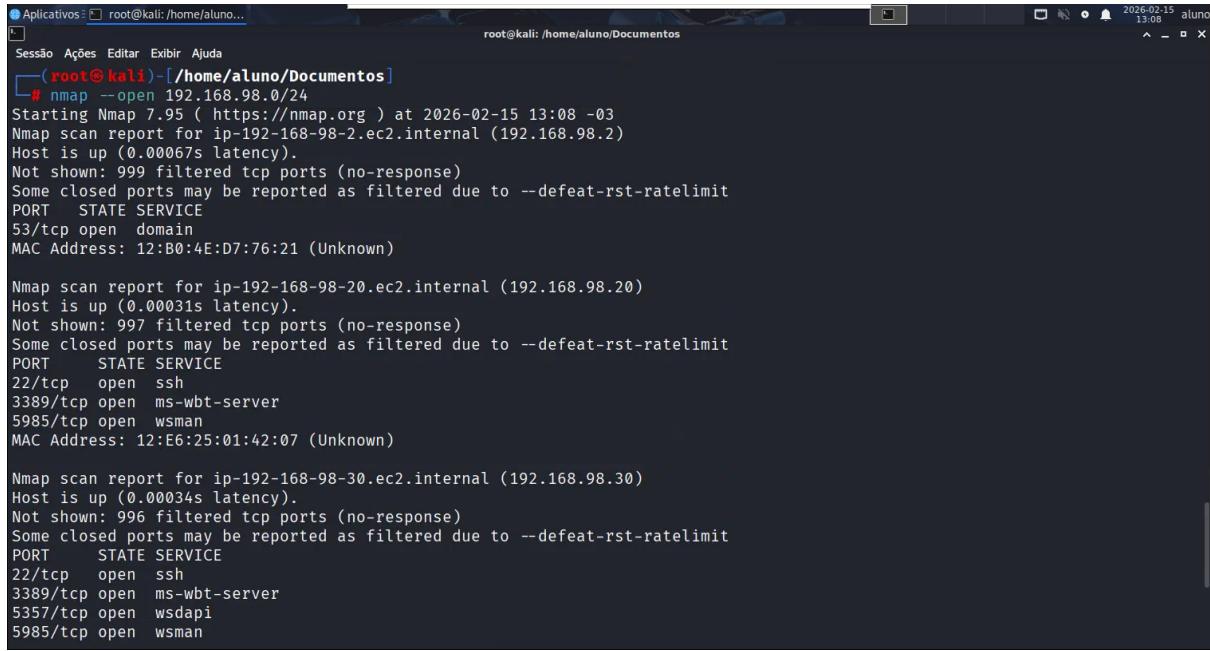
Foram realizadas buscas práticas utilizando diferentes termos. A pesquisa por *OpenSSL* retornou múltiplos exploits relacionados a falhas conhecidas, incluindo vulnerabilidades de negação de serviço, execução remota de código e problemas de criptografia. Também foram apresentados shellcodes e documentos técnicos associados.

Na sequência, foi realizada a busca por *OpenSSH*, onde foram identificados exploits relacionados a enumeração de usuários, execução remota de comandos e escalonamento de privilégios em diferentes versões do serviço.

Por fim, foi executada uma busca específica para *Windows 10*, utilizando o parâmetro `-t` para restringir os resultados ao título dos exploits. Essa consulta demonstrou diversas vulnerabilidades associadas a aplicações e serviços executados no sistema operacional.

Adicionalmente, foi acessado o site oficial do Exploit Database por meio do navegador web, permitindo explorar a base de dados online e complementar a análise realizada localmente.

Atividade 3.2. Explorando Varreduras de Rede com Nmap no Kali Linux



```
root@kali:~/Documentos
# nmap --open 192.168.98.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-15 13:08 -03
Nmap scan report for ip-192-168-98-2.ec2.internal (192.168.98.2)
Host is up (0.00067s latency).
Not shown: 999 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
53/tcp    open  domain
MAC Address: 12:B0:4E:D7:76:21 (Unknown)

Nmap scan report for ip-192-168-98-20.ec2.internal (192.168.98.20)
Host is up (0.00031s latency).
Not shown: 997 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
MAC Address: 12:E6:25:01:42:07 (Unknown)

Nmap scan report for ip-192-168-98-30.ec2.internal (192.168.98.30)
Host is up (0.00034s latency).
Not shown: 996 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
5985/tcp  open  wsman
```

Fig. 2: Varredura de rede com Nmap identificando hosts ativos e serviços disponíveis

Sobre:

Nesta atividade, foi utilizada a ferramenta *Nmap (Network Mapper)*, amplamente empregada em segurança ofensiva e defensiva para descoberta de hosts, serviços e análise de redes. O objetivo foi compreender técnicas básicas de varredura em um ambiente controlado, com fins exclusivamente acadêmicos.

Inicialmente, foi realizado o acesso ao ambiente Kali Linux via RDP e, em seguida, obtidos privilégios administrativos utilizando o comando de elevação de permissões. Com isso, foi possível executar comandos que exigem acesso privilegiado.

Para identificar as interfaces de rede disponíveis, foi utilizado o comando de listagem de interfaces, permitindo verificar o endereço IP associado à interface principal (*eth0*), pertencente à rede 192.168.98.0/24. Essa informação é essencial para definir o escopo da varredura.

Na primeira etapa, foi realizada uma varredura de descoberta de hosts utilizando a opção **-sn**. Esse tipo de varredura, conhecido como *ping scan*, tem como finalidade identificar quais dispositivos estão ativos na rede, sem realizar a análise de portas. O resultado apresentou os hosts que responderam às requisições, confirmando sua disponibilidade.

Em seguida, foi feita a extração dos endereços IP identificados, utilizando o encadeamento de comandos com *pipes*. O fluxo de dados foi filtrado para selecionar apenas os IPs encontrados, os quais foram armazenados em um arquivo de texto denominado *ips.txt*. Essa abordagem facilita o reaproveitamento dos dados em análises posteriores.

Posteriormente, foi executada uma varredura do tipo *TCP ACK Scan*, utilizando a opção **-sA**. Essa técnica é empregada para identificar a presença de mecanismos de filtragem, como firewalls, analisando as respostas aos pacotes enviados. Com isso, é possível classificar as portas como filtradas ou não filtradas, sem necessariamente determinar se estão abertas.

Na sequência, foi realizada a identificação de portas abertas com a opção **--open**. Essa varredura apresenta apenas portas consideradas abertas ou potencialmente acessíveis. Os resultados indicaram que a maior parte dos hosts possuía portas filtradas, sugerindo a existência de controles de segurança na rede. Em um dos hosts, foi observada a porta 53/TCP em estado acessível, associada ao serviço de resolução de nomes (DNS).

Por fim, foi utilizada a opção `--packet-trace`, que permite visualizar detalhadamente os pacotes enviados e recebidos durante a varredura. Essa funcionalidade possibilita analisar o comportamento da comunicação de rede em baixo nível, incluindo cabeçalhos e respostas dos hosts. A análise revelou a existência de serviços ativos, como SSH (porta 22/TCP) e RDP (porta 3389/TCP).

Ao término da atividade, o arquivo gerado foi removido, mantendo o ambiente organizado.

Atividade 3.3. Interceptando tráfego de navegador com o Burp Suite no Kali Linux

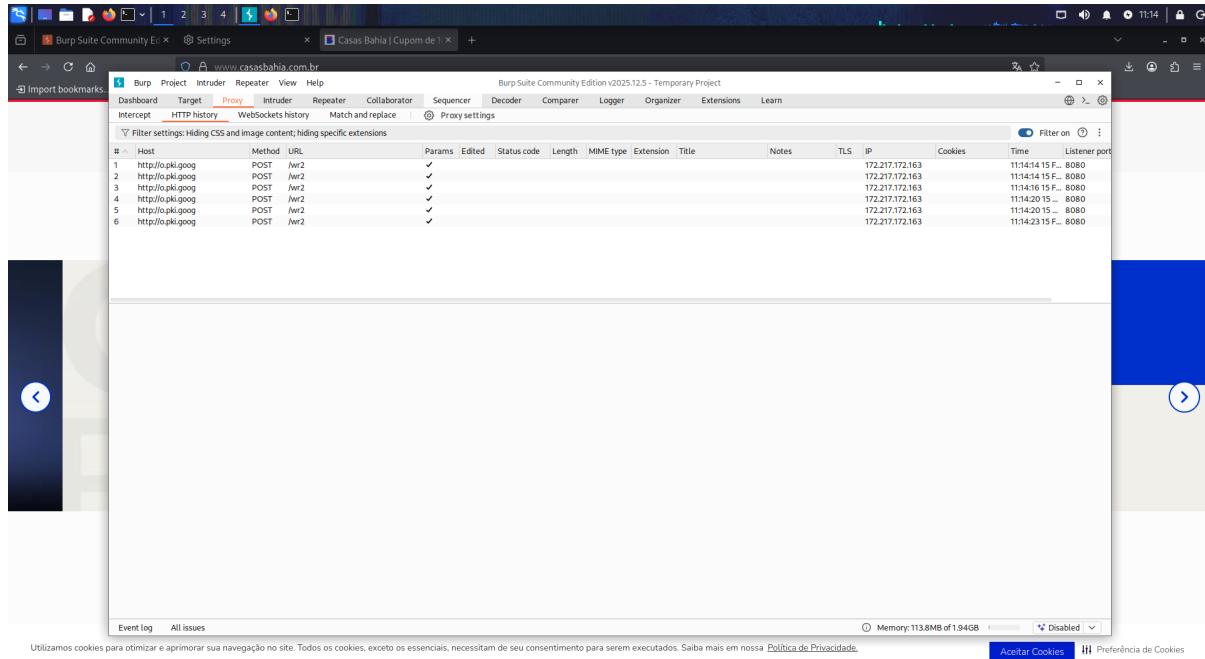


Fig. 3: Interceptação de requisições HTTP utilizando o Burp Suite

Sobre:

Nesta atividade, foi realizada a análise do tráfego de navegação web por meio da ferramenta *Burp Suite*, amplamente utilizada em testes de segurança de aplicações. O objetivo foi compreender o funcionamento de um proxy interceptador, permitindo visualizar e manipular as requisições enviadas entre o navegador e os servidores web, sempre com finalidade exclusivamente acadêmica.

Inicialmente, o ambiente Kali Linux foi acessado remotamente via RDP utilizando as credenciais fornecidas. Em seguida, o *Burp Suite* foi iniciado a partir do terminal com o usuário padrão. Durante a inicialização, foram aceitos os termos de uso da ferramenta e selecionada a opção de projeto temporário em memória, utilizando as configurações padrão.

Após a inicialização, foi acessada a aba *Proxy*, onde se verificou que o interceptador estava inicialmente desativado. Também foi possível observar, nas configurações do proxy, que o serviço estava em execução na interface local 127.0.0.1 na porta 8080.

Na sequência, o navegador Mozilla Firefox foi configurado manualmente para utilizar o proxy local do *Burp Suite*. Para isso, foram ajustadas as configurações de rede, definindo o endereço IP 127.0.0.1 e a porta 8080, além de habilitar o uso do proxy também para conexões HTTPS.

Após a configuração, foi acessado o endereço <http://burp>, permitindo o download do certificado da autoridade certificadora do *Burp Suite*. Esse certificado foi posteriormente importado no navegador, garantindo que conexões HTTPS pudessem ser interceptadas sem apresentar erros de segurança.

Com o ambiente devidamente configurado, a interceptação foi ativada na aba *Proxy*. Em seguida, ao acessar um site HTTPS no navegador, foi possível observar que a requisição foi interceptada pelo *Burp Suite* antes de chegar ao destino. Nesse momento, o navegador permaneceu aguardando, enquanto os detalhes da requisição, como cabeçalhos HTTP, cookies e informações do cliente, eram exibidos na ferramenta.

Também foi analisado o histórico de requisições na aba *HTTP history*, onde todas as comunicações realizadas pelo navegador são registradas. Essa funcionalidade permite acompanhar detalhadamente o fluxo de dados entre cliente e servidor.

Por fim, a requisição interceptada foi liberada manualmente por meio do botão *Forward*, permitindo que o navegador carregasse a página solicitada. Após os testes, o proxy foi desativado nas configurações do navegador e as aplicações foram encerradas.

Atividade 3.4. Escutando requisições com o Netcat no Kali Linux



The screenshot shows a terminal window titled "Aplicativos: Restore Session — Mozilla Firefox". The command entered is "nc -l -p 5555 -v". The output shows the server listening on port 5555 and receiving a connection from a Firefox browser on IP 192.168.98.40. The browser's User-Agent is Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0. The request details show a GET / HTTP/1.1 method, Host header (192.168.98.40:5555), and various Accept headers.

```
aluno@kali: ~
$ nc -l -p 5555 -v
Listening on 0.0.0.0 5555
Connection received on ip-192-168-98-40.ec2.internal 51888
GET / HTTP/1.1
Host: 192.168.98.40:5555
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

Fig. 4: Captura de requisição HTTP utilizando o Netcat em modo escuta

Sobre:

Nesta atividade, foi utilizada a ferramenta **Netcat** (nc) para observar requisições de rede em um ambiente *Kali Linux*. O objetivo foi entender, de forma prática, como conexões TCP podem ser recebidas e analisadas em nível de aplicação, evidenciando o funcionamento básico de comunicações em rede.

Inicialmente, foi realizado o acesso remoto ao sistema via RDP. Em seguida, foram obtidos privilégios administrativos por meio do comando **sudo -i**, permitindo a execução de operações que exigem maior nível de permissão.

Na sequência, foi consultada a ajuda da ferramenta com **nc -help**, permitindo observar os principais parâmetros disponíveis. Entre eles, destacam-se as opções para definição de protocolo, porta, modo de escuta e exibição detalhada das conexões.

Para preparar o ambiente de teste, foi identificado o endereço IP da máquina utilizando o comando **ifconfig**. O endereço obtido foi essencial para permitir que o navegador realizasse a conexão com o serviço em escuta.

Em seguida, o Netcat foi configurado para operar como um servidor, utilizando o comando **nc -l -p 5555 -v**. Nesse contexto:

- **-l** habilita o modo de escuta;
- **-p 5555** define a porta de atendimento;
- **-v** ativa a saída detalhada das conexões.

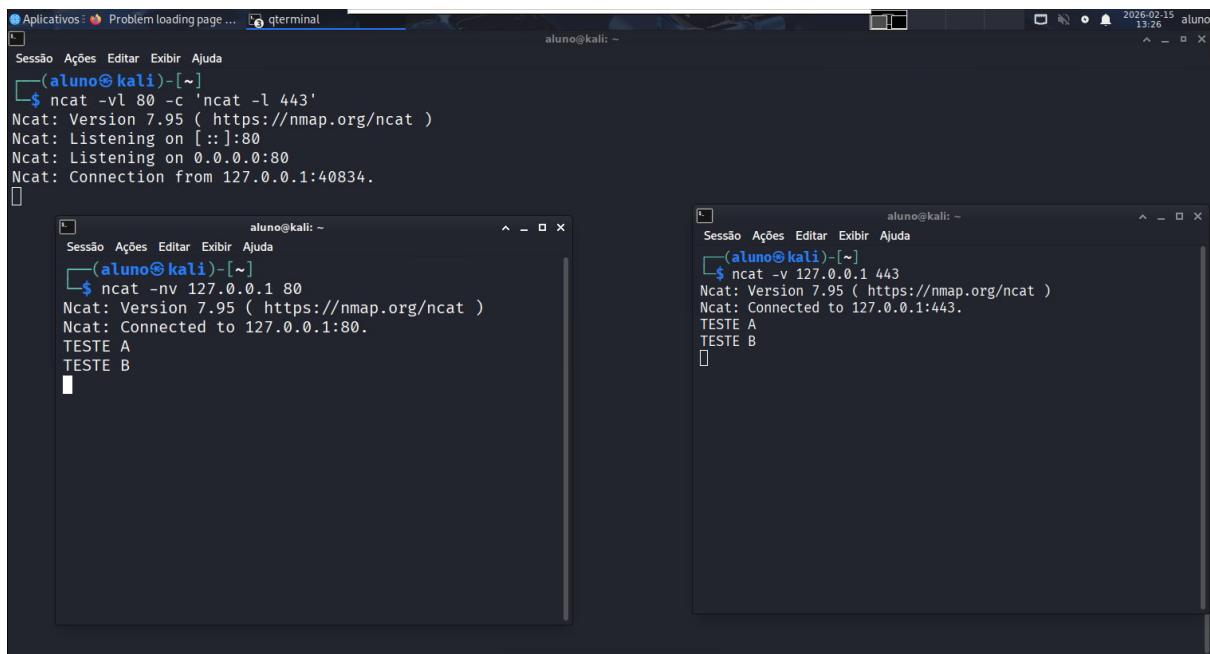
Com o serviço em execução, foi aberto o navegador web e realizada uma requisição HTTP para o endereço correspondente ao IP da máquina na porta configurada. Essa ação resultou no estabelecimento de uma conexão com o Netcat, que passou a exibir, em tempo real, os dados recebidos.

Foi possível observar informações relevantes da requisição HTTP, como o método **GET**, o cabeçalho **Host**, o campo **User-Agent** e outros parâmetros associados às preferências do cliente. Esses dados evidenciam como o protocolo HTTP transmite metadados importantes durante a comunicação entre cliente e servidor.

A análise demonstrou que o Netcat pode ser utilizado como uma ferramenta simples para inspeção de tráfego, permitindo visualizar requisições em texto puro sem a necessidade de softwares mais complexos. Esse tipo de abordagem é útil para fins didáticos, testes de conectividade e compreensão do funcionamento de protocolos de rede.

Por fim, os aplicativos utilizados foram encerrados, concluindo o experimento.

Atividade 3.5. Redirecionando tráfego via portas com o Netcat no Kali Linux



The figure displays three terminal windows from the Kali Linux desktop environment. The top window shows the initial setup where Ncat is listening on port 80 and connecting to port 443. The middle window shows a client Ncat connecting to port 80 from the local host. The bottom window shows the server Ncat receiving data from port 80 and sending it back to port 443, demonstrating bidirectional communication.

```
aluno@kali: ~
$ ncat -vl 80 -c 'ncat -l 443'
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Listening on [::]:80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 127.0.0.1:40834.

aluno@kali: ~
$ ncat -nv 127.0.0.1 80
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Connected to 127.0.0.1:80.
TESTE A
TESTE B

aluno@kali: ~
$ ncat -v 127.0.0.1 443
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Connected to 127.0.0.1:443.
TESTE A
TESTE B
```

Fig. 5: Comunicação entre portas 80 e 443 utilizando redirecionamento com Ncat

Sobre:

Nesta atividade, foi demonstrado o redirecionamento de conexões de rede entre portas distintas utilizando a ferramenta **Ncat**, presente no ambiente *Kali Linux*. O objetivo foi compreender, na prática, como o tráfego pode ser encaminhado entre serviços, permitindo a comunicação indireta entre diferentes pontos.

Inicialmente, foi realizado o acesso ao sistema e a elevação de privilégios com o comando **sudo -i**, possibilitando a execução de serviços em portas privilegiadas, como a porta 80.

Em seguida, foi iniciado um processo de escuta na porta 80 com o comando **ncat -vl 80 -c 'ncat -l 443'**. Nesse cenário, o Ncat foi configurado para receber conexões na porta 80 e, ao identificar uma nova conexão, executar outro processo de escuta na porta 443. Dessa forma, estabeleceu-se um mecanismo de encaminhamento entre essas duas portas.

O modo verboso permitiu acompanhar o estado da aplicação, indicando que o serviço estava ativo tanto em IPv4 quanto em IPv6. Após a inicialização, foi aberto um segundo terminal para realizar uma conexão com o endereço local (**127.0.0.1**) na porta 80, simulando um cliente acessando o serviço.

A conexão foi estabelecida com sucesso, sendo registrada no terminal que estava em escuta, evidenciando a recepção da requisição. Em seguida, foi aberto um terceiro terminal para conectar-se à porta 443 do mesmo host, criando o outro extremo da comunicação.

Com ambas as conexõesativas, foi possível validar o redirecionamento de dados. Mensagens enviadas em um dos terminais foram recebidas no outro, demonstrando que o tráfego estava sendo efetivamente encaminhado entre as portas 80 e 443. Esse comportamento evidencia a capacidade do Ncat de atuar como um intermediário na comunicação, permitindo a troca bidirecional de dados.

A atividade também ilustra conceitos fundamentais de redes, como escuta em portas, estabelecimento de conexões TCP e encaminhamento de tráfego, aspectos frequentemente explorados em cenários de testes de segurança e análise de rede.