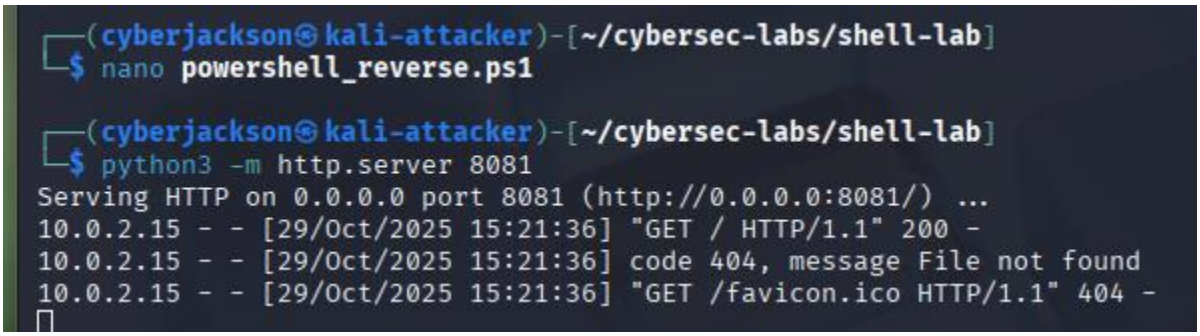
Introduction:

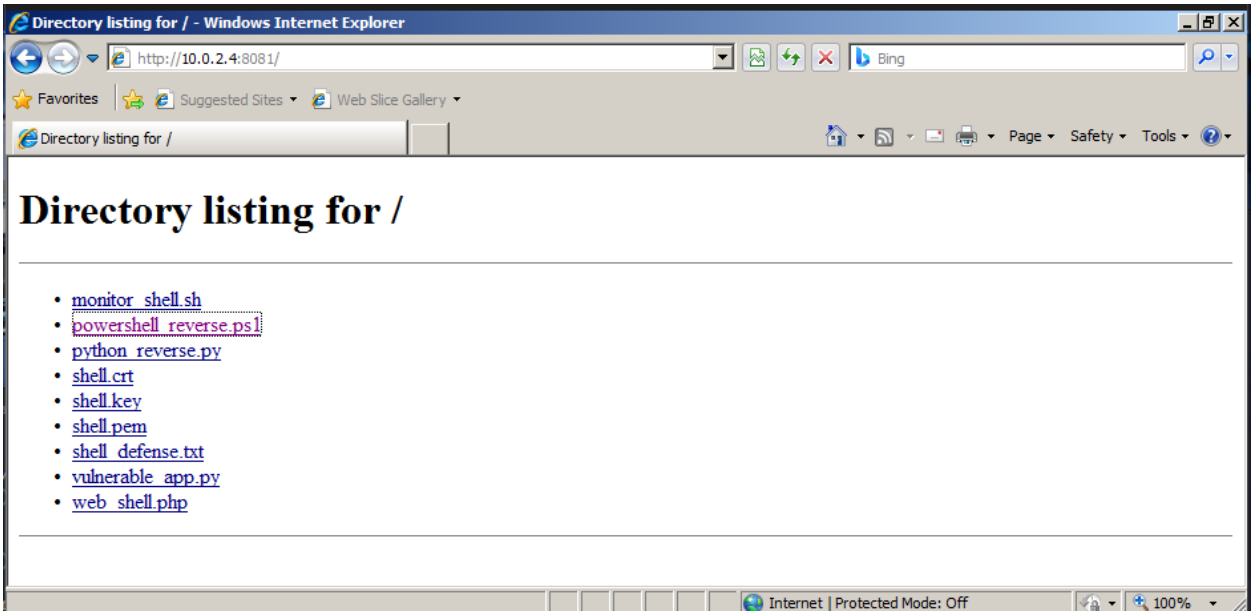
During penetration testing and post-exploitation activities shell access is the backbone of the procedures. Bind shells listen on the taget system for incoming connections, while reverse shells get the victim to call you back by connecting to the attacker's machine. My understanding of bind/reverse shells will be enforced by the activities I do in this lab. Specifically, I will create various types of shell connections, identify and exploit shell vulnerabilities, implement shell stabilization techniques, and apply these concepts to real-world penetration testing scenarios.

Exercise: PowerShell reverse shell

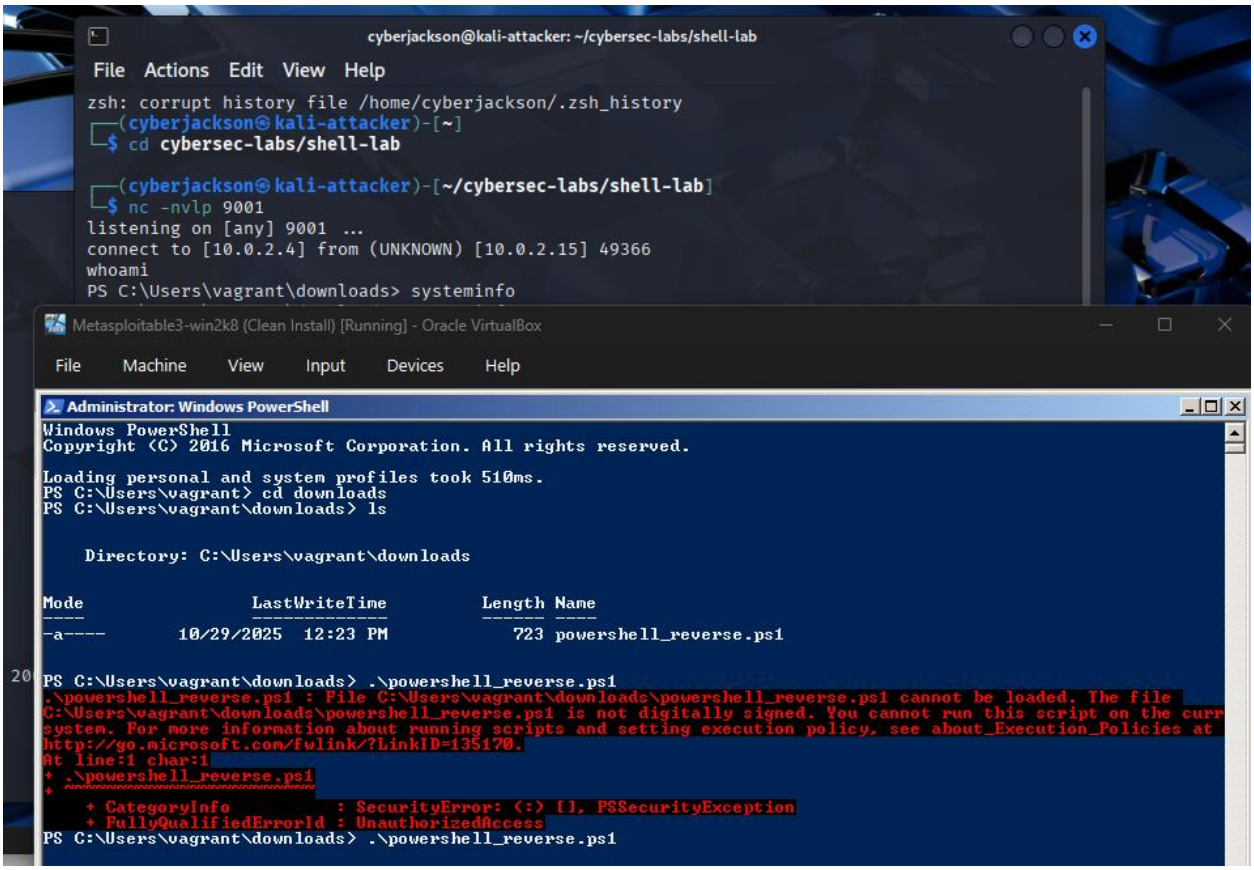
Setting up script and opening a web server



Navigating to webserver on metasploitable3 (Windows 2k8 machine) to download powershell script

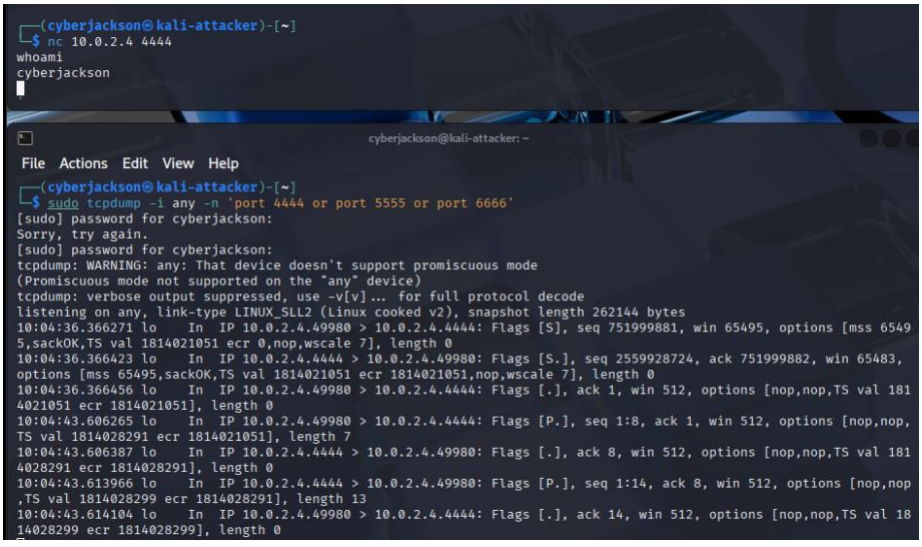


Begin listening and execute script to establish shell connection



Exercise: Shell Detection and Prevention

Monitor network connections (Terminal 3 Monitor below, Attacker terminal above)



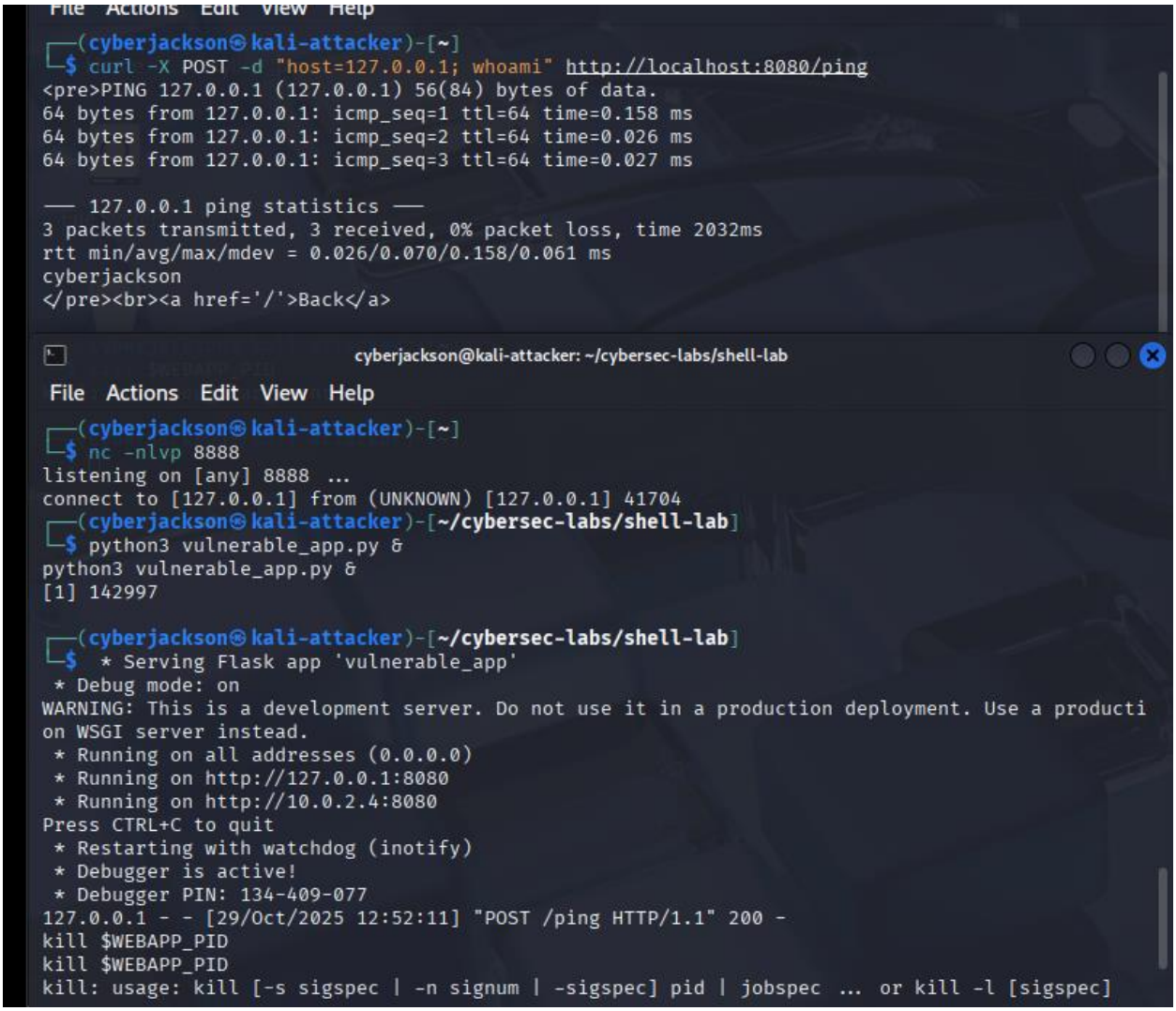
Exercise: Firewall rules to block shells

```
┌──(cyberjackson㉿kali-attacker)-[~/cybersec-labs/shell-lab]
└─$ sudo iptables -A INPUT -p tcp --dport 4444 -j DROP
sudo iptables -A INPUT -p tcp --dport 5555 -j DROP
sudo iptables -A OUTPUT -p tcp --dport 4444 -j DROP

┌──(cyberjackson㉿kali-attacker)-[~/cybersec-labs/shell-lab]
└─$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:4444
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:5555

Chain FORWARD (policy DROP)
target     prot opt source               destination
DOCKER-USER  all  --  0.0.0.0/0            0.0.0.0/0
DOCKER-ISOLATION-STAGE-1  all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLI
DOCKER     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:4444

Chain DOCKER (1 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target     prot opt source               destination
DOCKER-ISOLATION-STAGE-2  all  --  0.0.0.0/0            0.0.0.0/0
RETURN     all  --  0.0.0.0/0            0.0.0.0/0

Chain DOCKER-ISOLATION-STAGE-2 (1 references)
target     prot opt source               destination
DROP       all  --  0.0.0.0/0            0.0.0.0/0
RETURN     all  --  0.0.0.0/0            0.0.0.0/0

Chain DOCKER-USER (1 references)
target     prot opt source               destination
RETURN     all  --  0.0.0.0/0            0.0.0.0/0

┌──(cyberjackson㉿kali-attacker)-[~/cybersec-labs/shell-lab]
└─$
```

d

Exercise: Practical Shell Scenarios

Web application command injection to shell



Conclusion:

After completing the lab I feel much more comfortable and knowledgeable about terminal shells. It is obvious to me now how shell techniques are critical during penetration testing or red teaming to gain initial access and control or simulate APT tactics. For incident response, security assessment, malware analysis understanding attack vectors and investigating how the shells were used are important techniques to the job.

Through this lab I have learned the difference between bind and reverse shells, created various types of shell connections, implemented shell stabilization techniques, used encrypted shells, detected shell-based attacks using system monitoring tools, and applied

defensive measures to prevent shell attacks. All of this comes together to bring me a practical understanding of the shell in penetration testing scenarios.

The most challenging part of the lab was setting up the Powershell reverse shell and the web-based reverse shell. In both cases I had trouble running and compiling the scripts and was tripped up with how the shells should be set up attacker vs. Target.

The most interesting part of the lab for me was implementing shell stabilization techniques. At first it was tricky to me to understand whether the connection was maintained after stabilizing the shell. But, after getting my head around what each of the stabilizing methods were doing and seeing the terminals connection maintained via running test commands. I learned that the stabilization techniques are important for getting more reliability and use out of your shell connections.