Laboratory 3

Software Vulnerabilities (high-level exploitations)

This lab is designed to guide you through building a practical penetration testing environment and simulating real-world attack scenarios. You'll set up a minimal yet functional network architecture that includes both vulnerable and secure systems, mimicking common enterprise layouts such as DMZs and segmented networks. Through a series of tasks, you'll exploit known vulnerabilities, attempt lateral movement, and escalate privileges — ultimately simulating an advanced persistent threat. The goal is to deepen your understanding of offensive security techniques and network exploitation while reinforcing concepts in ethical hacking and system hardening.

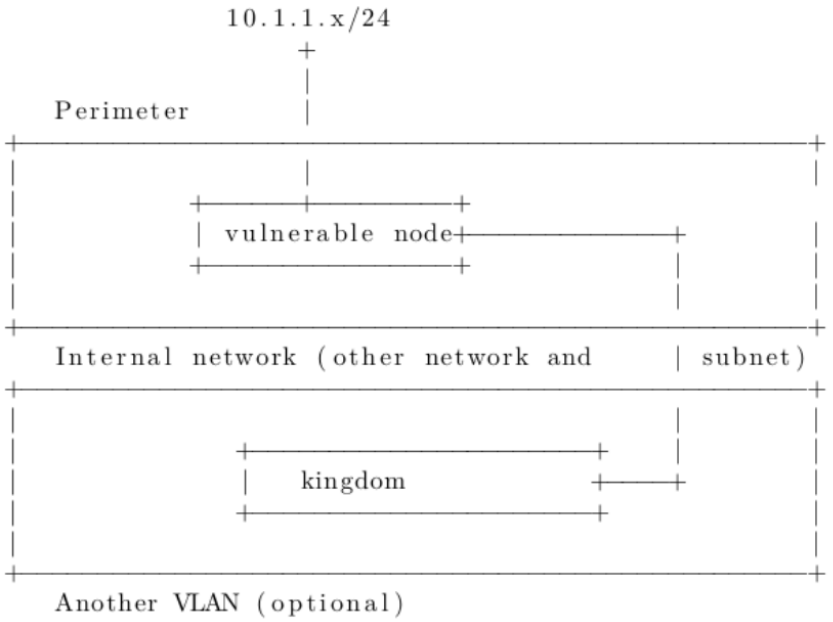
Task 1 - Setup infrastructure for penetration testing

Setup a minimal network architecture containing at least:

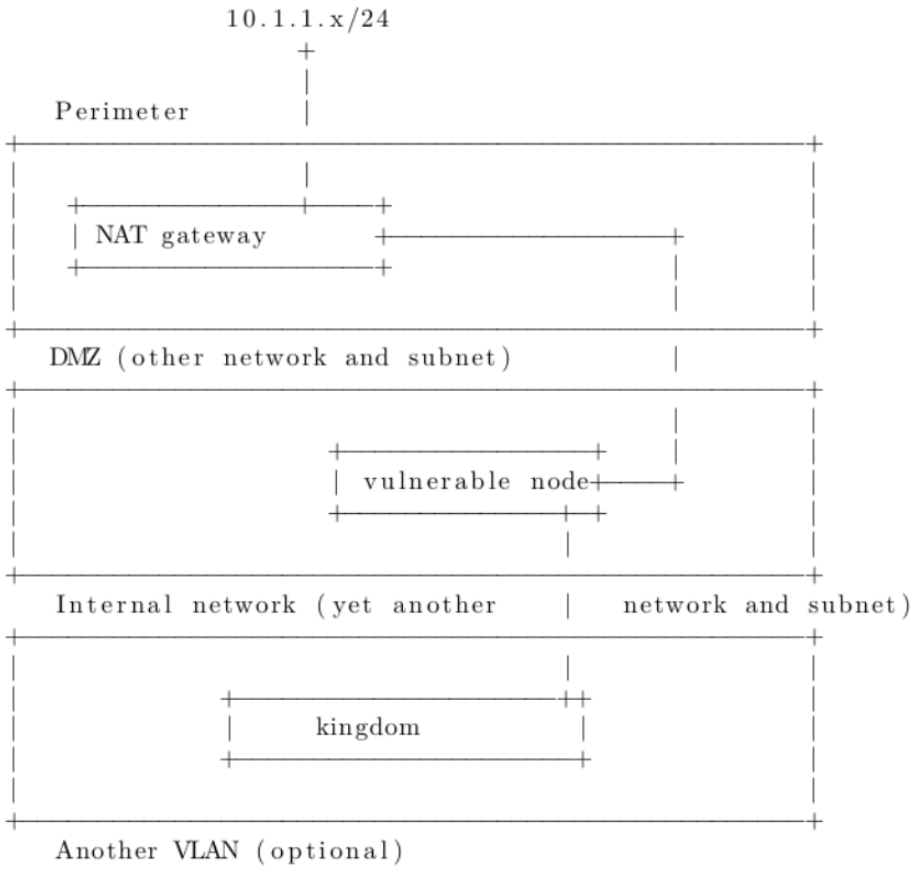
- An attacker host (could be your host OS in the goal to save compute resources)
- A vulnerable node (ideally with an RCE)
- A node that is not vulnerable (kingdom) but that we need to get credentials for (and it's the network link that is vulnerable) or implement any other techniques to steal sensitive data/get
- unauthorized access
- An isolated network for the kingdom or in the DMZ (choose your favourite routing solution)

Some proposed network topologies that could be:

1. **LAN/Perimeter example with only two members**



2. DMZ example with only two members



Task 2 - Make an exploitation

- Try to validate your chosen software vulnerability to attack a vulnerable node.
- Example of old good ones working RCE vulnerabilities (you are free to find others and may be newer):
    - CVE-2015-1635 (IIS)
    - CVE-2017-0144 - EternalBlue
    - CVE-2018-1000861 (Jenkins)
    - CVE-2019-0708 - Bluekeep
    - CVE-2020-7247 (OpenSMTPD)
- Explore the environment and networks as much as possible with the given access to *vulnerable node*.
- To make a PoC, one vulnerability is enough, but you are free to implement more.

Task 3 - Attack a non-vulnerable node (kingdom)

- Get access to the internal network of the kingdom.
- To implement the attack, try to explore with which hacking tools you can implement this (maybe something about SSH MITM?).
- Kingdom-PC can be connected to any different vulnerable node subnet, it doesn't matter.
- To make a PoC with one working hacking tool/technique is enough.

Task 4 - Privilege escalation flow

1. Choose a privilege escalation scenario and set up a vulnerable node. You can use a vulnerable node from previous tasks. If your previously chosen vulnerability can provide you with a privilege escalation attack.
2. Understand the process of chosen vulnerability and describe how it works.
3. Test and validate it. If you do something with privilege escalation in Task 3, just extend the explanations here then.
4. Deliver your vulnerable instance for your partner to attack.

Bonus 1 - More penetration

Demonstrate an additional non-RCE vulnerabilities:

- Network pivoting / route fuzzing
- ARP spoofing
- DNS spoofing
- Pass the hash
- SSL MITM
    - Non-authenticated
    - Private CA
    - CVE-2020-0601: Curveball
- SSH MITM (needs to be persistent and something advanced...)

Bonus 2 - Persistent penetration

Once you compromise a vulnerable node, make sure that you stay there including after a reboot. It can be done using DIY methods (task scheduler & netcat) or using available malware and rootkits. Also, establish a covert channel between the victim network and yours.