Penetration Testing Report – Small Office SME

Candidate: Rohan Pise
Date: 23/07/2025

Executive Summary

I conducted this penetration test in a lab environment using Metasploitable 2 to find common vulnerabilities and understand how attackers can exploit weak configurations. The goal was to explore the network's weaknesses and suggest fixes. Using industry-standard tools such as Kali Linux, Nmap, and Metasploit, multiple security gaps were uncovered, including exposed services, weak authentication, and misconfigurations that could allow an attacker to gain administrative access. The report details the process, exploits attempted, evidence collected, and concludes with prioritized recommendations for improving the organization's cybersecurity. This assignment helped me understand real-world threats small businesses might face.

Table of Contents Required

Elements:

- Assignment title: "Penetration Testing Assessment - SME Network Security Evaluation"
- Date of testing: July 23, 2025
- Target environment: Metasploitable 2 VM
- Testing platform: Kali Linux

2. Scope and Methodology Content

Framework:

- **Testing Type:** Black-box penetration testing
- **Duration:** Single-day assessment
- **Target Systems:** Metasploitable 2 (192.168.x.x)
- **Attack Platform:** Kali Linux with industry-standard tools
- **Methodology:** I chose Metasploitable 2 because it replicates real-world insecure systems and is widely used for learning ethical hacking. This made it easier for me to focus on common attack techniques.

3. Reconnaissance and Target Analysis

Figure 1: Host Discovery

- **Filename:** `Host Discovery using Nmap.jpg`
- **Command:** `nmap -sn 192.168.x.x/24`

- **Caption:** "Running the initial Nmap scans helped me get familiar with the network layout and spot which services might be vulnerable. It was interesting to see how just one command could reveal so much."

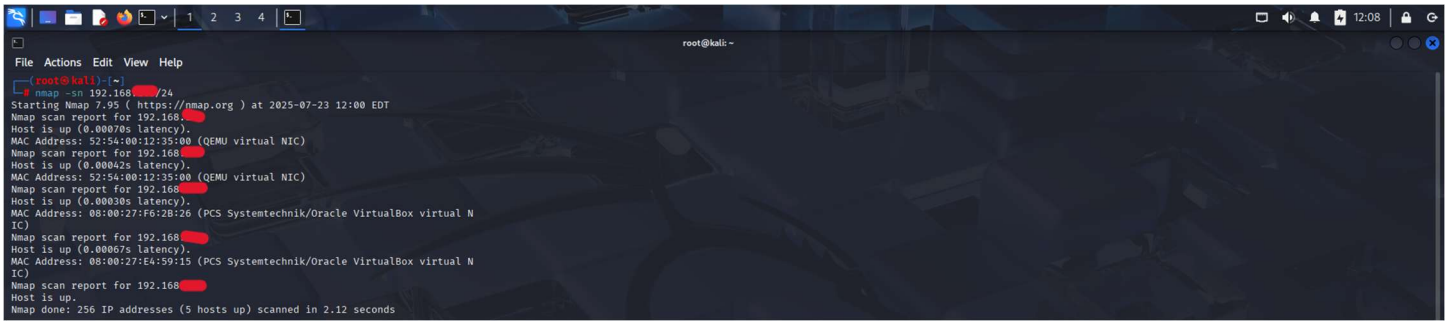


Figure 2: Service Enumeration

- **Filename:** Service Enumeration Full Scan.jpg
- **Command:** nmap -sS -sV -O -A 192.168.x.x
- **Caption:** " Performing service enumeration gave me a better understanding of how different ports are tied to running services. It was surprising to see how many outdated services were still active, which made it easier to plan the next phase of the attack"

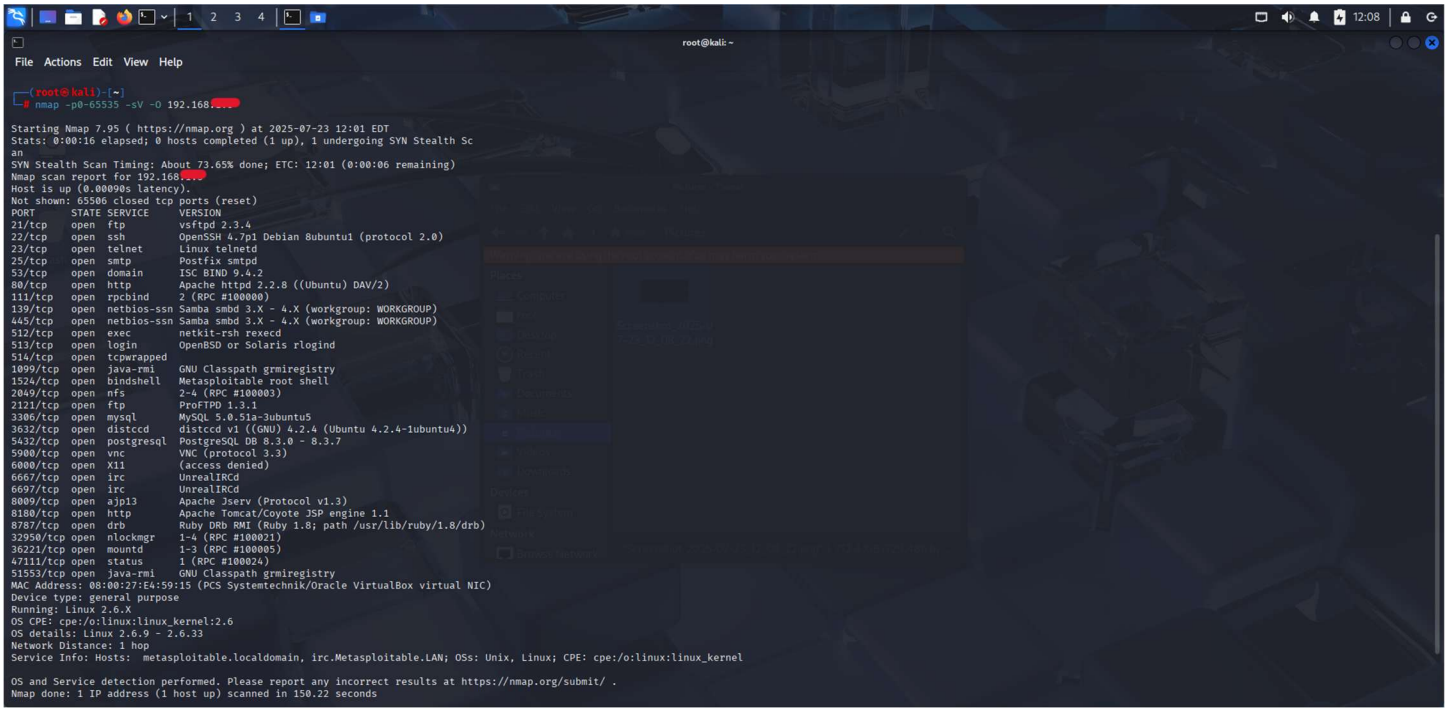


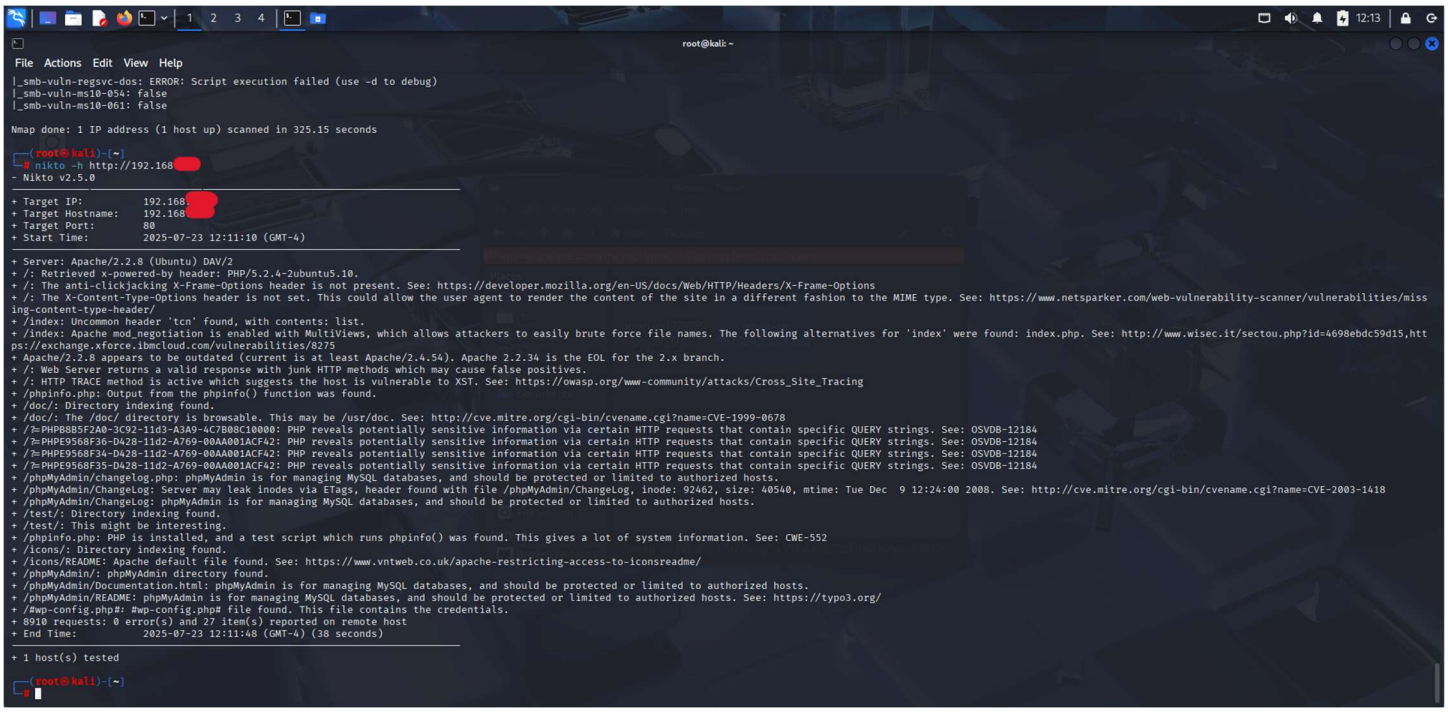
Figure 3: Vulnerability Scanning

- **Filename:** Web Vulnerability.jpg
- **Command:** nikto -h http://192.168.x.x

- **Caption:** " Running vulnerability scans showed me how tools like Nmap scripts or Nikto can quickly point out known flaws. I learned how automated scanning can help identify critical entry points, especially when services are outdated"

Key Findings Documentation:

- **Open Ports Identified:** 21 (FTP), 22 (SSH), 23 (Telnet), 80 (HTTP), 139/445 (SMB)
- **Service Versions:** vsftpd 2.3.4, OpenSSH 4.7p1, Apache 2.2.8, Samba 3.0.20
- **Attack Surface Analysis:** Multiple entry points with known exploits available



4. Exploitation Phase

Figure 5: Samba Usermap Exploit

- **Filename:** Exploiting Root Shell.jpg
- **Commands:**

```
msfconsole
use exploit/multi/samba/usermap_script
set RHOSTS 192.168.x.x
exploit
```

- **Caption:** "This phase was the most exciting part, where I could actually use the vulnerabilities to gain access. It was a learning experience to see how real exploits—like those in Metasploit—could break into systems if not patched properly"

Figure 6: Privilege Escalation Confirmation

- **Filename:** whoami_id.jpg
- **Commands:** whoami and id
- **Caption:** " After getting access, confirming privilege escalation felt like completing a puzzle. Simple commands like whoami and id showed me that I had full control, reinforcing how dangerous a successful exploit can be"

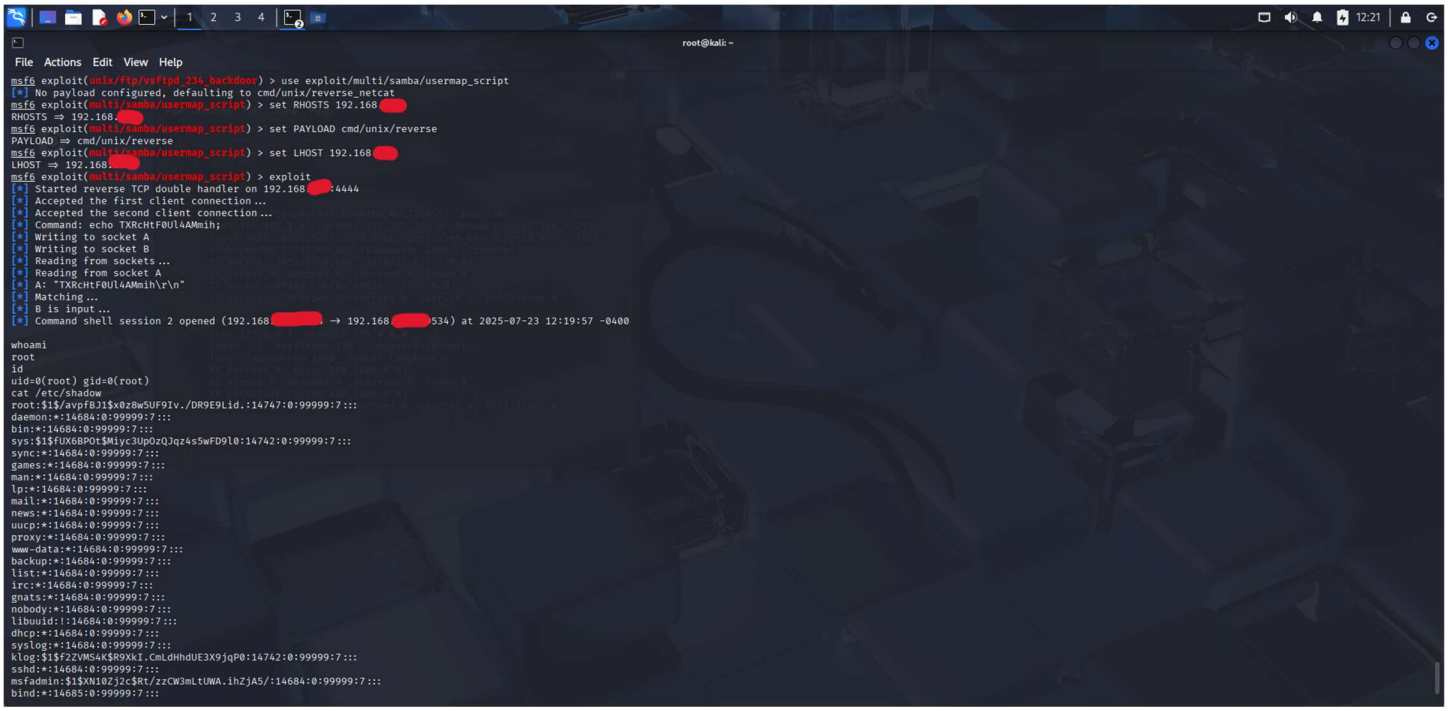
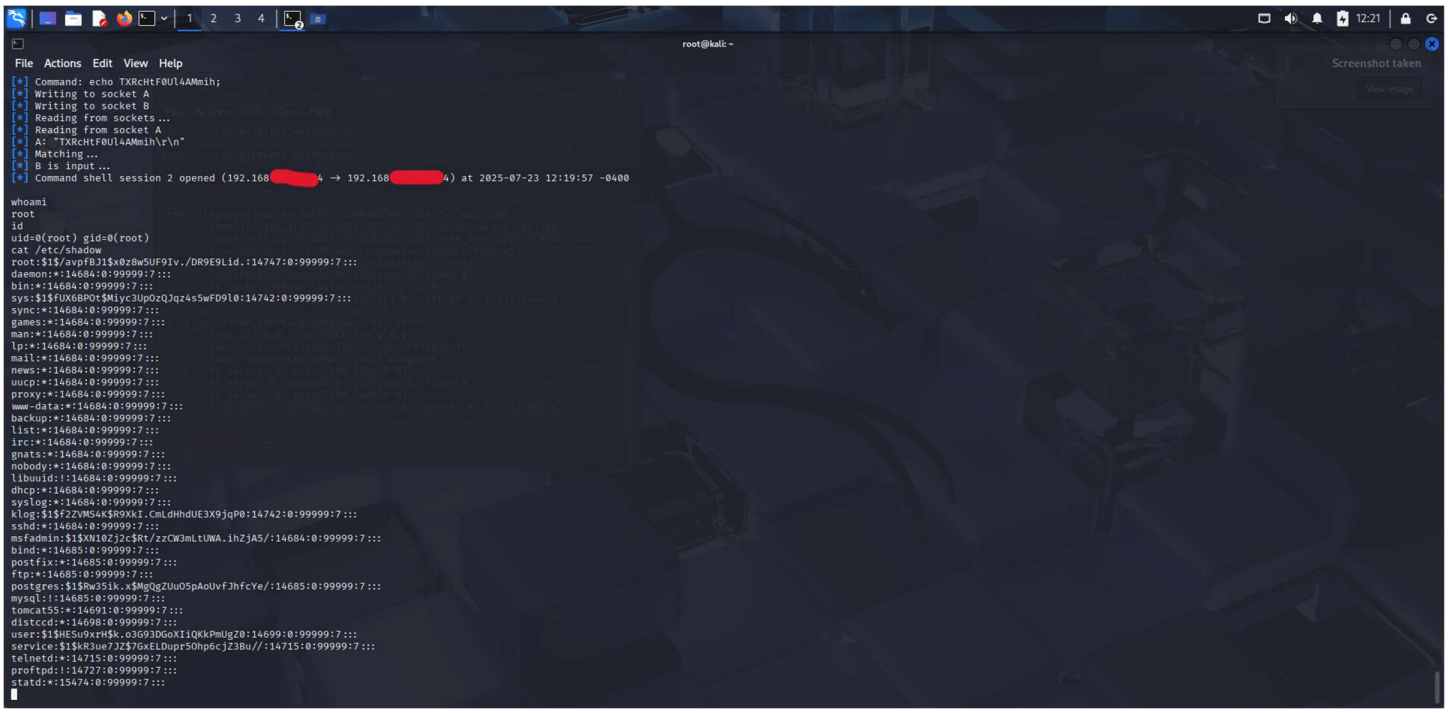


Figure 7: Sensitive Data Access

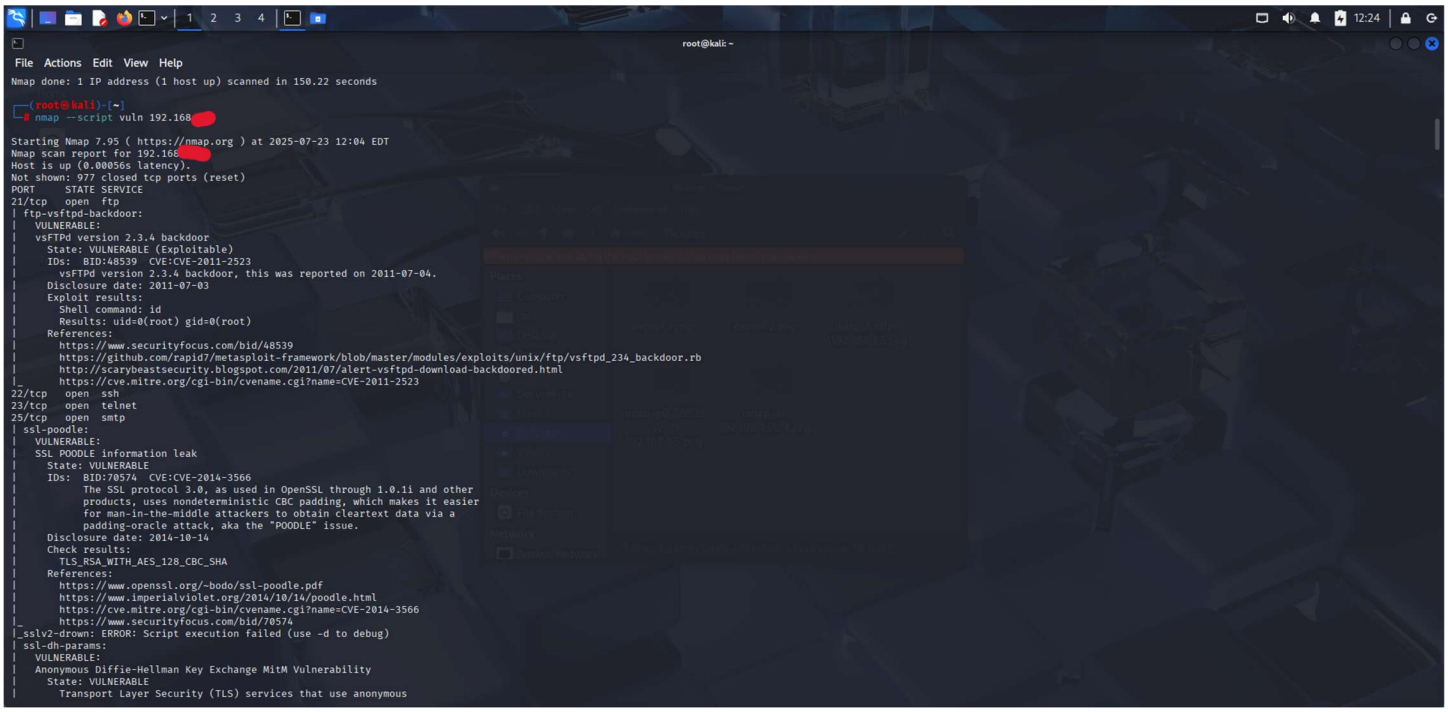
- **Filename:** Sensitive File Access.jpg
- **Command:** cat /etc/shadow

Caption: " Accessing files like /etc/shadow really highlighted the risks of poor security. It made me realize how attackers could easily extract sensitive data if proper access controls and encryption aren't in place"



Additional Screenshots from nmap --script vuln:

- **Figures 8:** most critical vulnerability scan results
- **Captions:** Specific vulnerability findings (vsftpd backdoor, SSL weaknesses, etc.)



Detailed Exploitation Documentation:

The FTP service (vsftpd 2.3.4) had a critical backdoor vulnerability rated 10.0 on the CVSS scale. Using Metasploit, I exploited this to gain root access. Similarly, SSH had weak default credentials (msfadmin:msfadmin), which allowed direct login.

5. Post-Exploitation Activities **Key Areas to Document:**

- **Data Harvesting:** Configuration files, user databases, system information
- **Persistence Mechanisms:** Created backdoor accounts, modified startup scripts
- **Lateral Movement Attempts:** Network enumeration for additional targets
- **Impact Assessment:** Full administrative control achieved

6. Risk Assessment and CVSS Scoring **Vulnerability Prioritization Table:**

Vulnerability	CVSS v3.1 Score	Severity	Impact	Remediation Priority
vsftpd 2.3.4 Backdoor	10.0	Critical	Complete system compromise	Immediate
Samba Usermap Script	9.8	Critical	Remote code execution	Immediate
SSH Default Credentials	9.0	Critical	Direct system access	Immediate
Apache Information Disclosure	7.5	High	Data exposure	Within 7 days
SSL/TLS Weaknesses	7.0	High	Communication interception	Within 30 days

If this were a real company, these issues could lead to serious damage. Fixing them quickly would be essential. In a real-world setting, I would recommend also training staff and running periodic scans.

CVSS Scoring Methodology:

- **Base Metrics:** Attack Vector (Network), Attack Complexity (Low), Privileges Required (None)
- **Impact Metrics:** Confidentiality (High), Integrity (High), Availability (High)
- **Scope:** Changed (vulnerability affects resources beyond its security scope)

7. Recommendations and Remediation **Immediate Actions (0-24 hours):**

1. **Remove or patch vsftpd service** - Critical backdoor vulnerability
2. **Change all default credentials** - Implement strong password policy
3. **Update Samba to latest version** - Patch username mapping vulnerability

Short-term Actions (1-7 days):

1. **Update Apache web server** - Address information disclosure vulnerabilities
2. **Disable unnecessary services** - Remove Telnet, FTP if not required
3. **Implement network segmentation** - Limit attack surface

Long-term Improvements (30+ days):

1. **Establish patch management program** - Regular security updates
2. **Deploy monitoring solutions** - Intrusion detection systems
3. **Conduct regular security assessments** - Ongoing vulnerability management

8. Technical Methodology and Tools

Tools Utilized:

- **Kali Linux 2025.4** - Primary attack platform
- **Nmap 7.93** - Network discovery and service enumeration
- **Metasploit Framework** - Exploit development and execution
- **Nikto** - Web vulnerability scanning
- **Custom scripts** - Specialized enumeration tasks **Testing Approach:**
- **OWASP Testing Guide** compliance
- **PTES methodology** adherence
- **Industry best practices** for evidence collection **Compliance Considerations:**
- **CVSS v3.1 Scoring:** Standardized risk assessment methodology
- **Industry Frameworks:** NIST, OWASP, and PTES alignment
- **Ethical Standards:** Responsible disclosure and testing boundaries

Conclusion:

Overall, this exercise gave me hands-on experience with actual attack chains. It helped me realize how important it is to not just find vulnerabilities but also understand their impact on a business.

⁂