NahamStore

Table of Contents

---

1. Methodology

1. **Information Gathering**: Collected target system details via network scanning tools (e.g., `nmap`).
2. **Enumeration**: Enumerated open ports, services, and directories.
3. **Vulnerability Identification**: Used automated tools and manual inspection to discover common vulnerabilities (e.g., SQL Injection, LFI).
4. **Exploitation**: Executed exploits to gain unauthorized access to sensitive areas of the system.
5. **Post-Exploitation**: Escalated privileges to gain administrative control.
6. **Documentation and Reporting**: Summarized findings and recommended mitigations.

2.Findings

Vulnerability	Description	Impact
Open Port - HTTP (80)	The web server was accessible and running vulnerable web services.	Unauthorized access to web pages
Directory Enumeration	Discovered hidden directories using `gobuster`.	Possible data leakage or admin panel

Vulnerability	Description	Impact
		exposure
SQL Injection (SQLi)	Login form was vulnerable to SQL Injection, allowing bypass authentication.	Access to database and sensitive data
Weak Credentials	Found default credentials in admin panel.	Unauthorized access to admin interface
File Upload Vulnerability	Allowed uploading of web shells.	Remote Code Execution (RCE)

---

3. Exploitation Details

Step 1: Enumeration with Nmap

Ran the following command:

```
bash
Copy code
nmap -sC -sV -oN nmap_scan.txt <target_ip>
```

Results:

- Open Port: 80 (HTTP)
- Service: Apache HTTP Server 2.4.29

Step 2: Directory Enumeration

Used `gobuster` to find hidden directories:

```
bash
Copy code
gobuster dir -u http://<target_ip>/ -w
/usr/share/wordlists/dirb/common.txt
```

Results:

- `/admin/`
- `/uploads/`

Step 3: SQL Injection on Login Form

Bypassed the login authentication using:

```
sql
Copy code
' OR 1=1--
```

This provided access to the admin panel without valid credentials.

Step 4: Uploading a Web Shell

- Navigated to `/uploads/` section.
- Uploaded a PHP web shell (`shell.php`).
- Accessed the shell via:

  ```
  perl
  Copy code
  http://<target_ip>/uploads/shell.php
  ```

- Achieved **Remote Code Execution (RCE)**.

Step 5: Privilege Escalation

- Discovered weak credentials for SSH:
  - **Username:** admin
  - **Password:** admin123
- Logged in via SSH and gained full control of the system.

Task 1: NahamStore

Task 2: Setup

Task 3: Recon

3.1. Jimmy Jones SSN

Answer: *521–61–6392*

Task 4: XSS

4.1. Enter an URL (including parameters) of an endpoint that is vulnerable to XSS

Answer: *http://marketing.nahamstore.thm/?error=*

4.2. What HTTP header can be used to create a Stored XXS

Answer: *User-Agent*

4.3. What HTML tag needs to be escaped on the product page to get the XSS to work?

Answer: *title*

4.4. What JavaScript variable needs to be escaped to get the XSS to work?

Answer: *search*

4.5. What hidden parameter can be found on the shop home page that introduces an XSS vulnerability.

Answer: *q*

4.6. What HTML tag needs to be escaped on the returns page to get the XSS to work?

Answer: *textarea*

4.7. What is the value of the H1 tag of the page that uses the requested URL to create an XSS

Answer: *Page Not Found*

4.8. What other hidden parameter can be found on the shop which can introduce an XSS vulnerability.

Answer: *discount*

Task 5: Open Redirect

5.1. Open Redirect One

Answer: *r*

5.2. Open Redirect Two

Answer: *redirect_url*

Task 6: CSRF

6.1. What URL has no CSRF protection

Answer: *http://nahamstore.thm/account/settings/password*

6.2. What field can be removed to defeat the CSRF protection

Answer: *csrf_protect*

6.3. What simple encoding is used to try and CSRF protect a form

Answer: *base64*

Task 7: IDOR

7.1. First Line of Address

Answer: *160 Broadway*

7.2. Order ID 3 date and time

Answer: *22/02/2021 11:42:13*

Task 8: Local File Inclusion

8.1. LFI Flag

Answer: *{7ef60e74b711f4c3a1fdf5a131ebf863}*

Task 9: SSRF

9.1. Credit Card Number For Jimmy Jones

Answer: *5190216301622131*

Task 10: XXE

10.1. XXE Flag

Answer: *{9f18bd8b9acaada53c4c643744401ea8}*

10.2. Blind XXE Flag

Answer: *{d6b22cb3e37bef32d800105b11107d8f}*

Task 11: RCE

11.1. First RCE flag

Answer: *{b42d2f1ff39874d56132537be62cf9e3}*

11.2. Second RCE flag

Answer: *{93125e2a845a38c3e1531f72c250e676}*

Task 12: SQL Injection

12.1. Flag 1

Answer: *{d890234e20be48ff96a2f9caab0de55c}*

12.2. Flag 2 (blind)

Answer: *{212ec3b036925a38b7167cf9f0243015}*