Penetration Test – Art's Tailor Shop

Benjamin Ruddy

2026-01-14

Contents

Executive Summary

From August 24th, 2022, up to December 12th, 2022, the Pr0b3 Inc. penetration testing firm carried out a penetration test on the network infrastructure of Art's Tailor Shop LLC to evaluate the security posture of the organization as well as the presence of vulnerabilities on their digital systems.

Various aspects of the company's digital security were tested, ranging from exploits related to vulnerable programs and services, to open-source intelligence gathering, to insecure configurations of hardware and software components.

Our testing showed numerous instances of good security practices being followed, including restrictions on DNS zone transferring, usage of secure, encrypted protocols such as HTTPS on web services, robust, centralized domain management in the form of Microsoft Active Directory, and usage of secure wireless communication protocols in the form of WPA2-EAP.

Nevertheless, there remained other parts of the infrastructure that contained issues ranging from innocuous misconfigurations to critical vulnerabilities that threatened the confidentiality, integrity, and availability of company systems. This report aims to not only provide details and risks associated with these findings, but also the steps to take in order to remediate them.

Project Overview

The penetration test can be roughly divided into the following eight stages:

1. Pre-engagement interactions – These include Q&A sessions with Art, scoping meetings, and status updates.
2. Intelligence gathering – This refers to all kinds of reconnaissance that allowed us to learn more about the company infrastructure, e.g. opensourc intelligence (OSINT), social engineering, DNS reconnaissance, etc.
3. Threat modeling – In this stage, we take time to analyze the information we have gathered and brainstorm ways in which we can use what we know to cause unintended or harmful consequences to the infrastructure.
4. Vulnerability Analysis – After modeling potential locations of weaknesses, we physically test the component of the infrastructure for the feasability of a particular vulnerability.
5. Exploitation – Should we be able to confirm a vulnerablity, this stage involves carrying out the technical procedures to abuse it (e.g. run malicious code to grant us a shell on a remote host).
6. Post-exploitation – After exploiting a vulnerability, this stage involves seeing how far we can get and what kind of damage an attacker may be able to cause upon successful exploitation

7. Reporting – i.e. the compilation of all our findings and recommendations into this document.

Note that steps two through six often function as a cycle, in which intelligence gathering, threat modeling, vulnerability analysis, and exploitation are performed multiple times as new network components are targeted, or new parts of the network are discovered.

Goals

First and foremost, the goal of this penetration test is to ensure that the digital systems of Art's Tailor Shop have a security posture that is robust and secure in preparation for their expanding web presence.

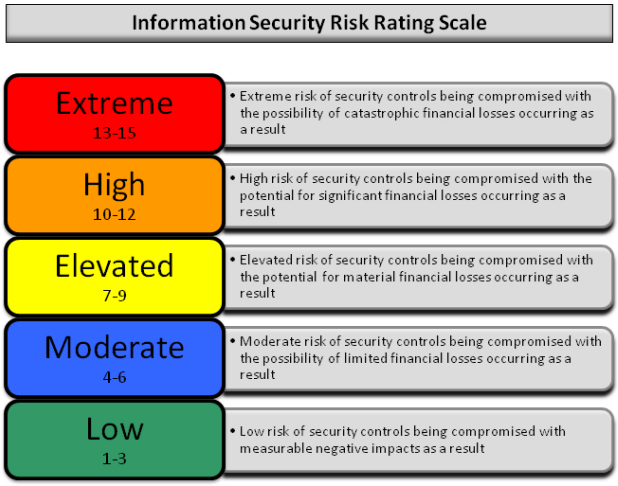
One may get away with a mediocre security posture in an isolated system with relatively little external contact, but a public-facing company infrastructure simply cannot afford to have weak security, as it compromises both the business itself as well as the customers of the business.

By assessing and improving their securing their network infrastructure through a penetration test, Art's Tailor Shop is not only protecting their business assets, but is also taking a greatly important step towards gaining customer trust and loyalty.

Risk Ranking/Profile

Overall Information Security Risk Rating: **12/15**

The above rating is based on the Information Security Risk Rating Scale, shown in the figure below.



The rating above is a cumulative ranking of risk as it refers to the overall digital infrastructure of Art's Tailor Shop. In many cases, vulnerabilities of critical importance were found, ranging from outdated, vulnerable network services to insecure configurations that could lead to administrator privileges on a host or the entire domain.

Despite this, our overall rating of 12 remains in the "High" risk range as opposed to the "Extreme" risk range (13-15) due to the fact that most egregious of vulnerabilities often required local network access (not achievable over the internet), or required prior exploitation such as social engineering or network sniffing to carry out. For example, although an outdated Windows SMB version allowed for total administrative control over the `ARTSTAILOR` domain, the machine could not be accessed over the public internet, and a a result, is not as prone to exploitation as it would be if the machine was public-facing.

Thus, the combination of medium-to-high risk vulnerabilities for public-facing infrastructure in combination with the vulnerabilities of critical importance that were locally-limited lead us to assign an overall score of 12 out of 15.

Summary of Findings

Individual findings were recorded and ranked based on the following CVSS scale, with a point assignment ranging from 0 to 10:

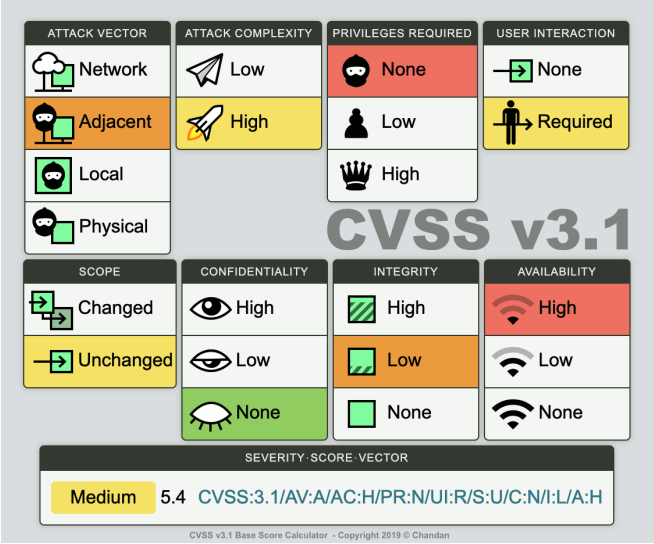
Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

The following table shows the distribution of vulnerabilities that were found:

Moderate	High	Critical
6	8	5

A total of **19 vulnerabilities** were found.

Note that each finding has also been assessed in regards to its attack vector (AV), attack complexity (AC), privileges required (PR), user interaction (UI), scope (S), confidentiality (C), integrity (I), and availability (A), as per the figure below.



Recommendation Summary

The recommendation summary can be roughly divided as follows:

OS hardening: Ensure that on a host-by-host basis, the operating system is updated to the latest available version. In addition, it is recommended to consult the relevant STIG (Security Technical Implementation Guide) for the respective OS for complete hardening. Available STIGs can be found at https://public.cyber.mil/stigs/

Network service hardening: On a host-by-host basis, ensure that frequent and routine software updates are being done as it relates to programs that run as a service on a network port for that host. In addition, review the specific recommendations for services as provided in the **Technical Report** that involve properly securing custom program/application functionality (e.g. handling file uploads, assigning proper permissions to certain programs/services).

Credential policy hardening: Ensure that across all forms of authentication on company services, users are required to implement sufficiently secure credentials, specifically as it relates to password strength. In general, passwords of 10 characters or more should be mandated, with a check to make sure that it is not a password found on commonly available exploited password lists, e.g. rockyou.txt. Consult NIST Special Publication 800-63B for more:

https://pages.nist.gov/800-63-3/sp800-63b.html.
In addition, ensure that no default credentials for any form of company interface exist on the network (e.g. default router login credentials).

Secure information storage: In many cases, sensitive information was exfiltrated from company systems, including customer credit cards. Ensure that any and all information that could provide an attacker with greater power on the network be stored in a secure, encrypted format – if it is necessary to store at all in the first place.

Technical Report

Introduction

The following are the findings that were identified across the penetration test. Once again, note the **Mitigation or Resolution Strategy** section associated with each one in order to get an idea of what to do about each of them.

Finding: *Vulnerable version of SMB protocol on backup domain controller allows for arbitrary remote code execution via specially-crafted network packets, allowing for domain administrator privilege escalation*

Severity Rating: 10.0

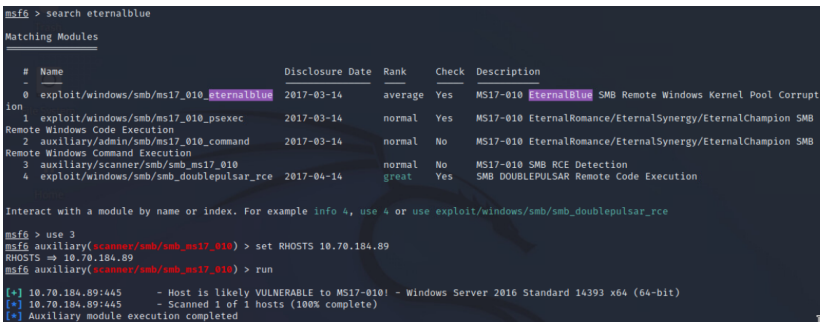
CVSS Base Severity Rating: 10.0 AV:A AC:L PR:N UI:N S:C C:H I:H A:L

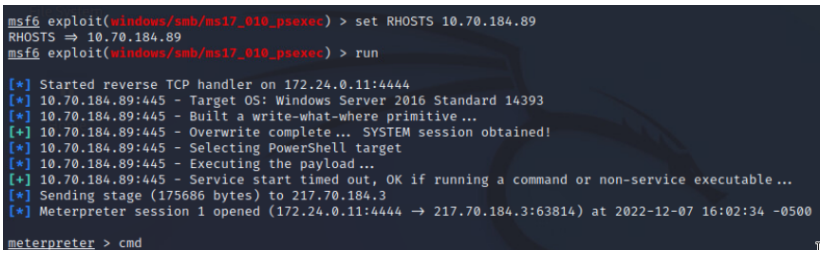
Vulnerability Description

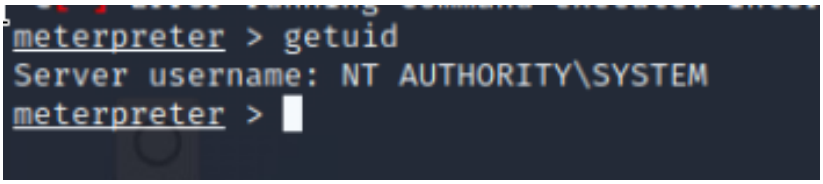
An backup domain controller was found at the IP address of 10.70.184.89 running an outdated version of the SMB protocol, typically used for network resource sharing across an organization. Due to a vulnerability in the way in which this version handles certain packets, an attacker may send a request that grants them arbitrary remote code execution on the machine, which can lead them to gain local administrative permissions.

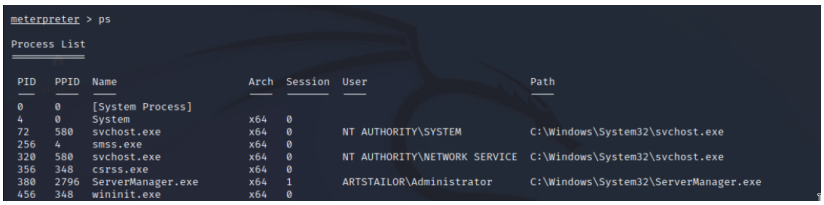
Because many of the processes on this domain controller run as the domain administrator user, a user is then easily able to migrate to those processes and subsequently gain complete domain administrator rights.

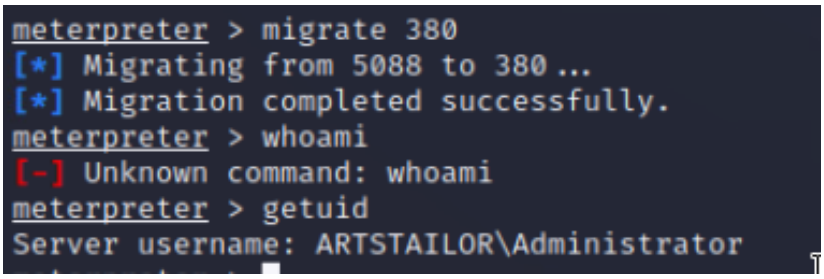
Confirmation method











Mitigation or Resolution Strategy

It is imperative to update the operating system version on the backup domain controller to the most recently issued version by Microsoft (usually done

through the Start Menu → Settings → Windows Updates → Check for Update).

Finding: *Local backdoor, originally installed for password changing, grants root terminal access on Windows logon screen*

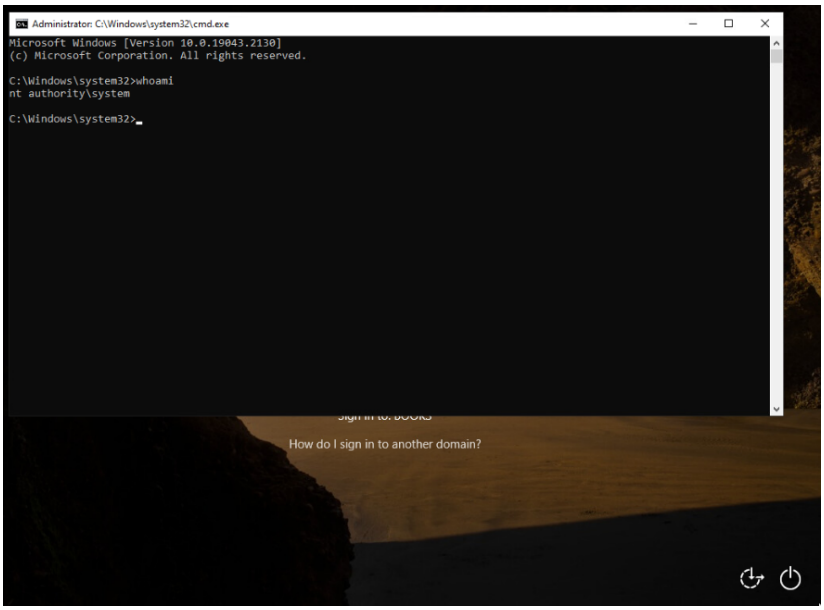
Severity Rating: 9.6

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

Vulnerability Description

Due to a seemingly benign need for the user 'debbie' to change their password, a backdoor was implemented that grants an Administrator-level command prompt shell to anyone who simply goes on the logon screen for the machine (e.g. through RDP) and presses the "Ease of access" button on the bottom-right.

Confirmation method



Mitigation or Resolution Strategy

First, delete the files that allow for this to happen (`reset.bat`, `cmd.bat`, `cmd-8.bat`, `number` in the `C:\` directory). Next, delete the Windows Registry key that sets cmd.exe as the debugging program for utilam.exe. This is located at HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\utilman.exe although it may be automatically deleted

by Windows 10 if the `reset.bat` script wasn't run before exiting the root terminal from the logon screen.

After deleting the files associated with the backdoor, it is important to only change user debbie's password through official means, such as with the Active Directory Users and Computers program with an appropriate domain administrator account.

In addition, it is recommended that an investigation is made into the user Oliver, who seems to be changing Debbie's domain account password.

Finding: *Vulnerable service VSFTPD 2.3.4 contains a malicious, remote backdoor*

Severity Rating

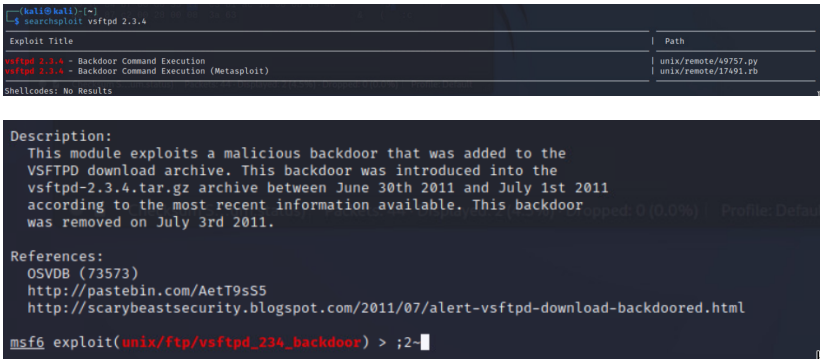
CVSS Base Severity Rating: 8.0 AV:N AC:L PR:N UI:N S:U C:L I:H A:H

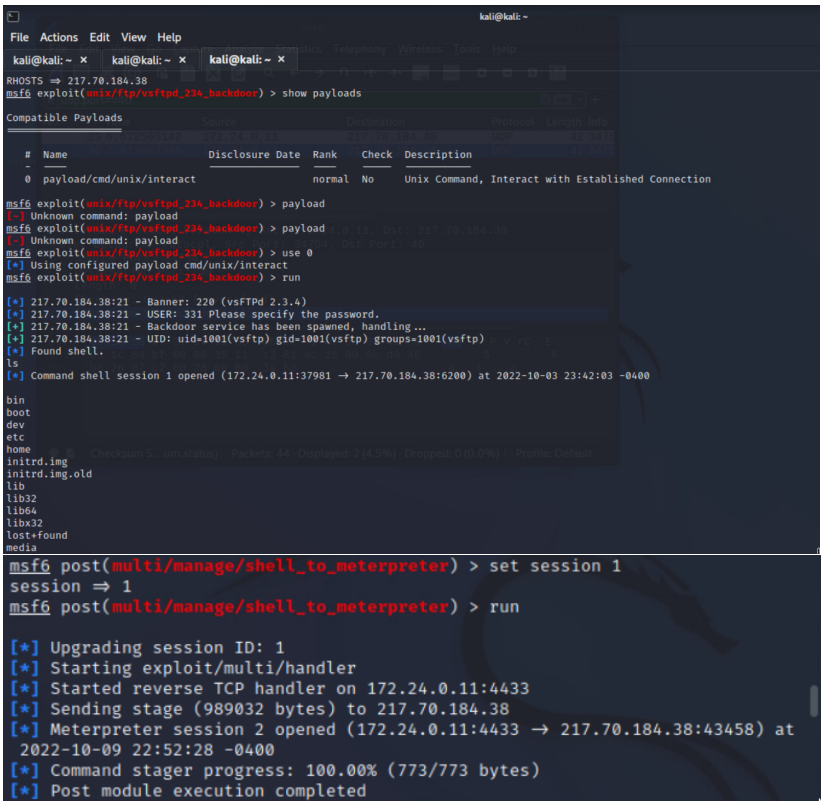
Vulnerability Description

This vulnerability involves a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011, and was removed on July 3rd 2011. It allows malicious users to connect to a shell listening on port 6200 on the remote machine upon logging into the FTP service with a username of `':)'`.

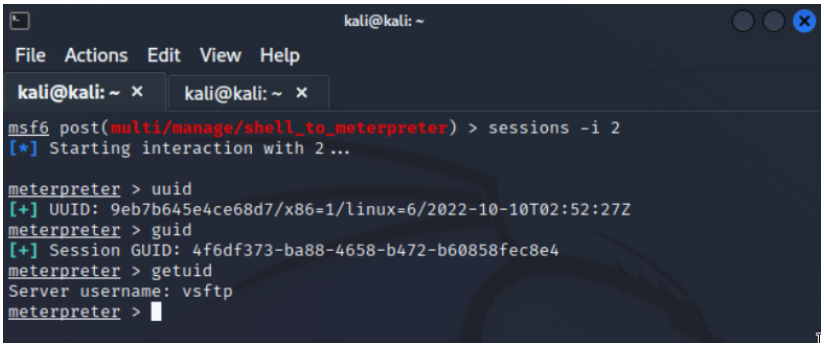
Note: This backdoor typically grants root privileges upon access, however, it only provided access to the `vsftp` upon exploiting it. Thus, the exploit is rated a 8.0 as opposed to the maximum of 10.

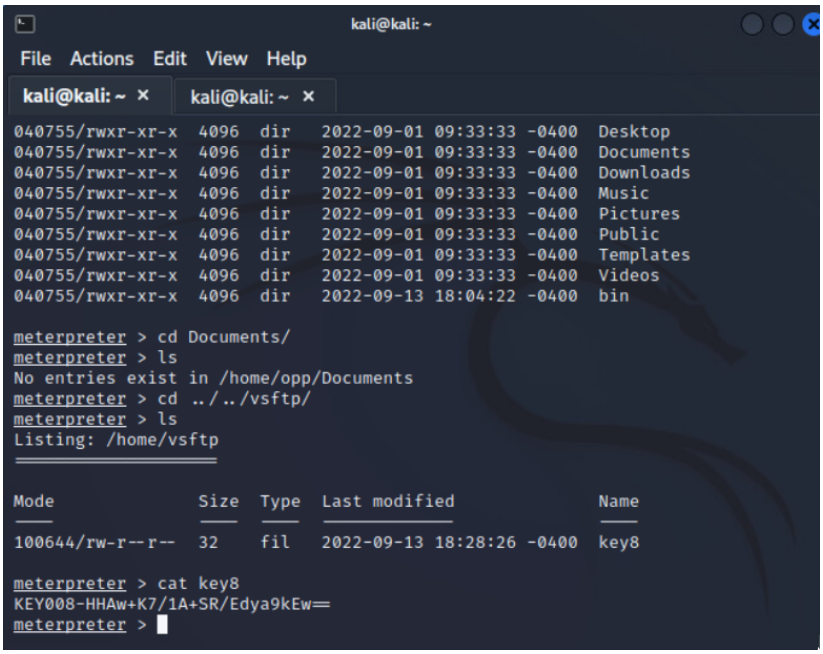
Confirmation methods



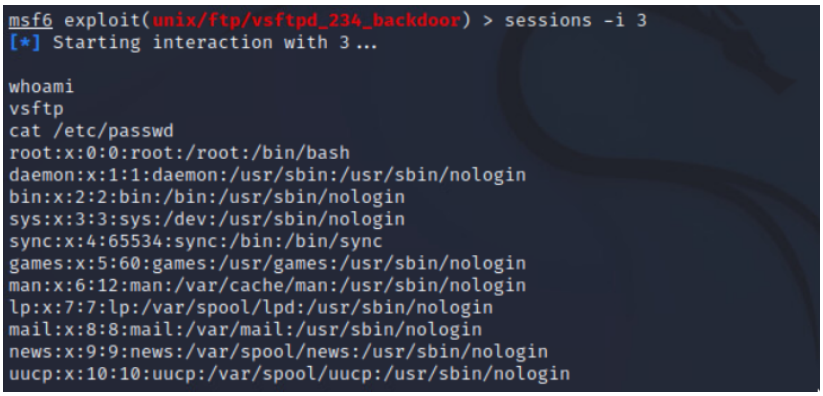


Attempting a shell upgrade using a post-exploitation module from Metasploit





Note the **KEY008** found above upon shell access.



For proof of compromise, the `/etc/passwd` file is shown above.

Mitigation or Resolution Strategy

Uninstall the backdoored version immediately, and replace with one that has been verified against the PGP signature of the developers.

If immediate software upgrading/reinstallation is not possible, then at a minimum, block all traffic attempting to come in through port 6200 on the host.

Finding: *Default credentials on pfSense router admin panel*

Severity Rating: 9.1

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

Vulnerability Description

On the `innerouter.artstailor.com` host (public IP 217.70.184.3), the pfSense credentials for the admin account are unchanged from the default credentials of `admin:pfsense`. This is a crucial vulnerability because it allows attackers complete control over network traffic within the subnet, with attacks such as Man in the Middle and even logging in to unintended services such as RDP now being a possibility if an attacker finds the IP and port combination.

Confirmation method

Simply log on to 217.70.184.3 with the aforementioned credentials on a web browser, and you will now be in the administrator account for the router.

Mitigation or Resolution Strategy

Change the credentials for the administrator ASAP.

Finding: *Insecure, client-side file validation allows for unrestricted file upload to web server, allowing for PHP remote code execution*

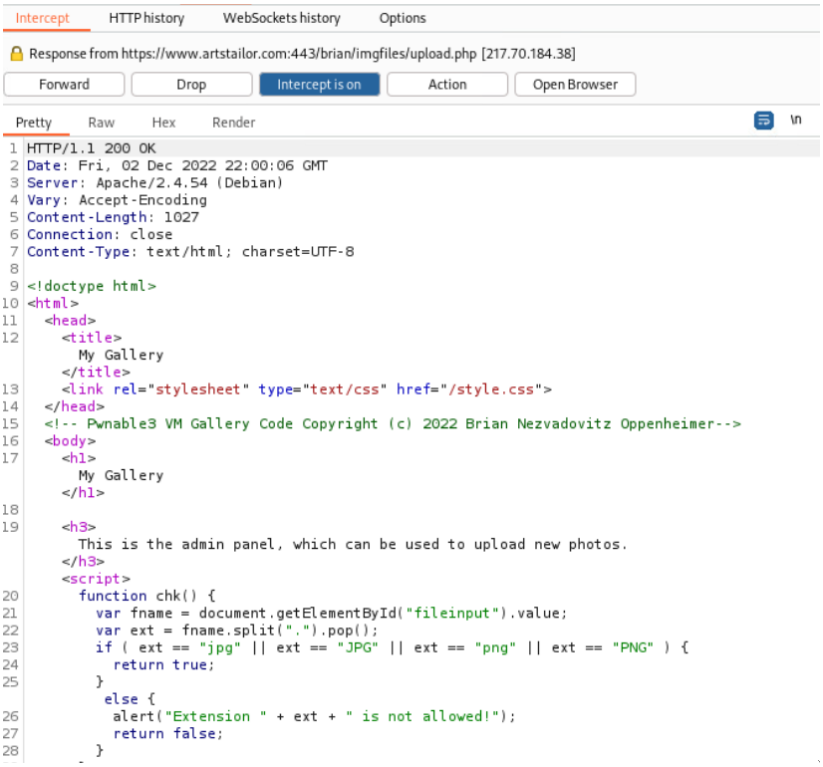
Severity Rating: 9.0

CVSS Base Severity Rating: 9.0 AV:N AC:L PR:H UI:N S:C C:H I:H A:L

Vulnerability Description

On the admin image uploading panel at `www.artstailor.com/upload.php`, employs insecure, client-side file validation to check that only image files are uploaded.

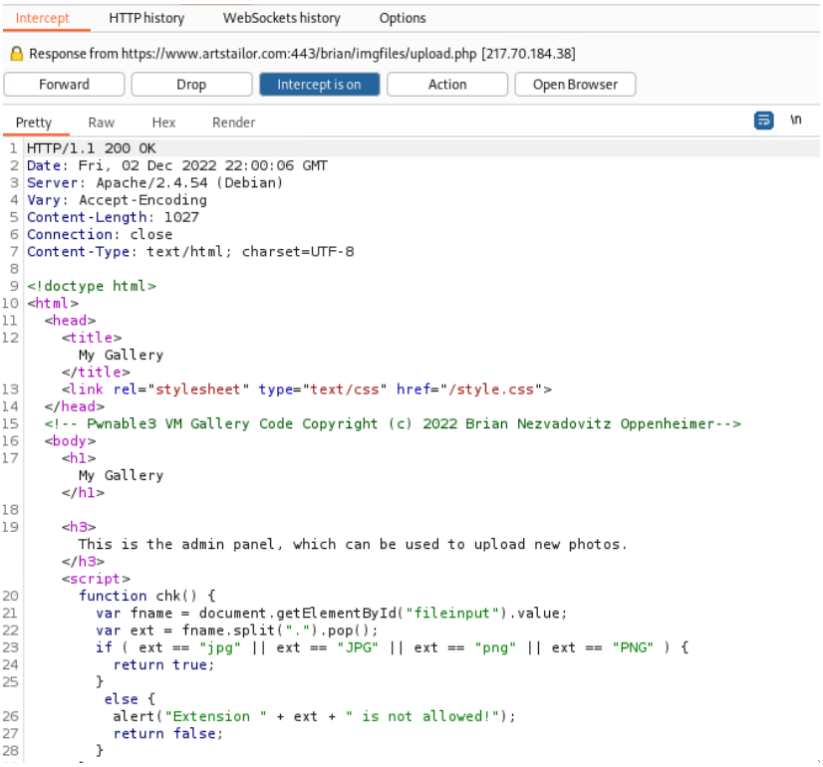
Firstly, the validation mechanism itself is flawed because it merely checks that the last characters of a filename match one of the approved extensions (e.g. png, jpg, PNG, etc.), as seen below:

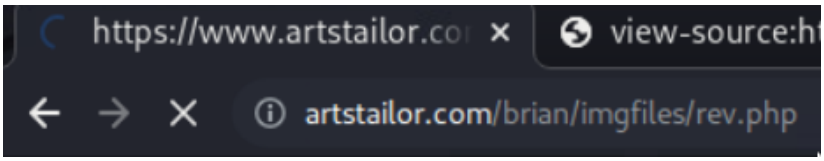


The mechanism can easily be bypassed by changing the filename of a non-image file. Furthermore, the validation is done client-side, meaning that a user can simply modify the `chk()` function's Javascript to allow any other file with any filename to be uploaded, e.g. a malicious PHP reverse shell named `rev.php`.

Taking the aforementioned action results in being able to visit the file's location at `brian/imgfiles/` to potentially run malicious code, as was done in this case to obtain a reverse shell, leading to subsequent exfiltration of sensitive information as shown below.

Confirmation method






```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8888
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::8888
Ncat: Listening on 0.0.0.0:8888
Ncat: Connection from 217.70.184.38.
Ncat: Connection from 217.70.184.38:35128.
Linux www 5.10.0-17-amd64 #1 SMP Debian 5.10.136-1 (2022-08-13) x86_64 GNU/Linux
 17:02:11 up  3:17,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

```
www-data@www:/var/www/html/brian/imgfiles/.information$ ls -lah
ls -lah
total 12K
drwx------ 2 www-data www-data 4.0K Nov 21 22:46 .
drwxr-xr-x 3 www-data www-data 4.0K Dec  2 17:02 ..
---------- 1 www-data www-data   32 Nov 21 22:45 ThisIsTheFileYouAreLookingFor
www-data@www:/var/www/html/brian/imgfiles/.information$ cat T
<es/.information$ cat ThisIsTheFileYouAreLookingFor
cat: ThisIsTheFileYouAreLookingFor: Permission denied
www-data@www:/var/www/html/brian/imgfiles/.information$ chmod u+r Thi
<formation$ chmod u+r ThisIsTheFileYouAreLookingFor
www-data@www:/var/www/html/brian/imgfiles/.information$ cat T
<es/.information$ cat ThisIsTheFileYouAreLookingFor
KEY020-+zot5HSExLMBZG+B9uAg7w==
www-data@www:/var/www/html/brian/imgfiles/.information$
```

Mitigation or Resolution Strategy

Firstly, move the file validation over to the server side as soon as possible.

Next, modify the validation method itself to follow industry best practices, such as those from https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload.

Finding: *Buffer overflow vulnerability on administrative program allows compromise of user account*

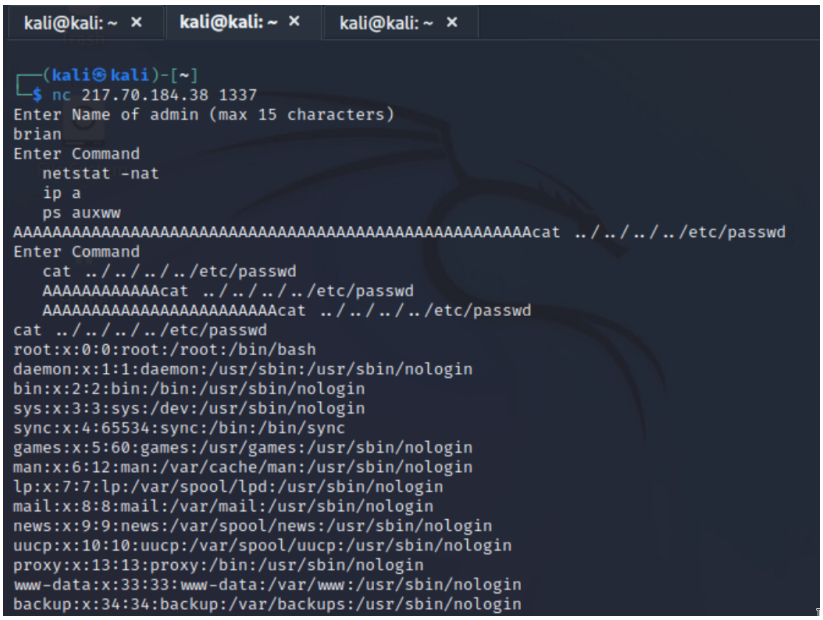
Severity Rating: 7.0

This vulnerability effectively lets a malicious actor run any command that the user `brian` has on the remote system.

> This attack does not, however, grant root permissions to upon exploitation.
> **CVSS Base Severity Rating: 7.0** AV:A AC:L PR:N UI:N S:C C:H I:L A:N

Vulnerability Description

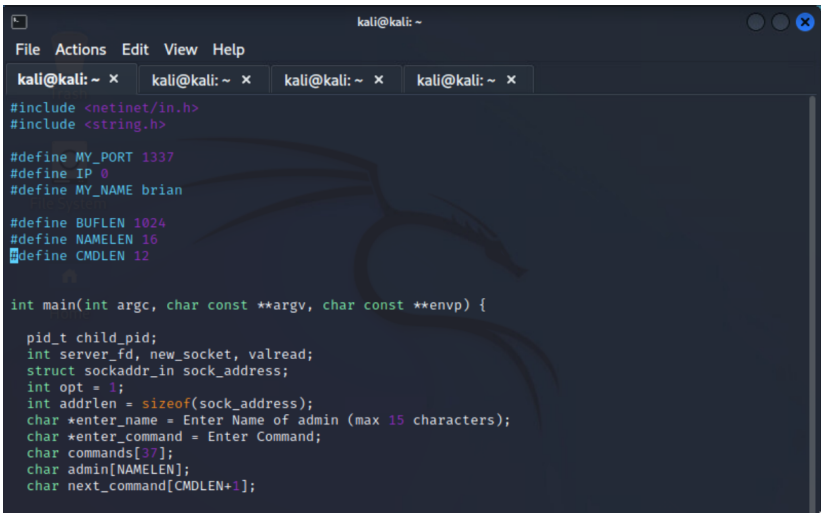
By logging in as `brian` on the service running on port 1337, and then inputting 53 'A's followed by the desired command, this vulnerability lets a malicious actor run any command on the system that the user `brian` is allowed to run, as shown in the following screenshot:



This is possible because the buffer that stores the command that the user input overflows, overwriting the list of acceptable commands from the program with whatever comes after the 53 'A' characters.

Confirmation method

Note the above screenshot showcasing access to the `/etc/passwd` file (`/etc/shadow` was not accessible as the compromised user), along with the following source code (obtained via overflowing the buffer and running `cat /home/brian/toool.c`):



Take note of the fact that BUFLEN is defined as 1024, while NAMELEN and CMDLEN are defined as 16 and 12, respectively. This means that (by looking at the code below the C definitions), that the admin array (buffer) is of length 16, and the next_command array is of length $12 + 1 = 13$. Now, take a look at the two calls to `fgets()` in the following screenshot

Brian made one right choice in using a function like `fgets`, which lets the source code writer specify the size of the input to be read into the buffer, as opposed to a function like `gets` which does not.

The reason this code is still vulnerable, however, is because the code specifies an input length that exceeds the size of the buffer it is reading it into. To be

specific, a maximum of 1024 tcharacters can be read into the `next_command` array (this is the buffer we overflowed in the screenshots) and the `admin` array. However, as we just saw, both of those are significantly smaller than 1024 characters in length. Thus, the possibility is opened that we can overflow these buffers, which we do, and it ends up spilling over into the `commands` array, altering the commands we can execute on the system

Mitigation or Resolution Strategy

As mentioned in the previous vulnerability, the best case scenario is to remove the program entirely as its purpose is better and more securely accomplished through established methods such as ssh'ing into the machine.

Alternatively, the source code can be edited to perform a check that the inputs for both the administrator username and the command name are equal to or smaller than the length of their respective buffers.

Finding: *Lack of authentication on an administrative system program*

Severity Rating: 4.3

This vulnerability may allow unwanted personnel to perform detailed information-gathering on the local machine related to its running processes and network services.

CVSS Base Severity Rating: 4.3 AV:A AC:L PR:N UI:N S:U C:L I:N A:N

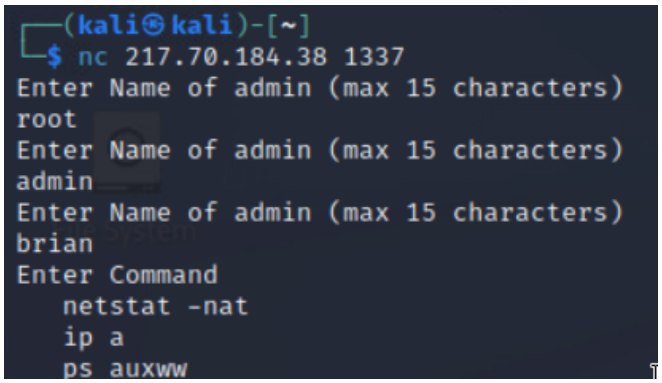
Vulnerability Description

On the 217.70.184.38 machine, the running service on port 1337 is intended to give system administrators information about the host without having to log in. It provides the ability to execute `netstat -nat`, `ip a`, and `ps auxww` to view network information, view local network interfaces, and monitor running processes on the system, respectively.

Due to the lack of authentication however, all a non-authorized user needs to view the same information is to know the name of an administrator, allowing them to view the above information regardless of their lack of authorization to do so.

Confirmation method

With the publicly available information that Brian posted, all a malicious actor needs to do is `nc 217.70.184.38 1337` and input the name `brian` to gain the aforementioned abilities:



Mitigation or Resolution Strategy

Ideally, remove the service altogether. Although it may save some time for system administrators, their job should be done through established, formal

methods that handle authentication already.

An authentication system could be implemented, such as a password system, but doing so properly would involve more time than is typically worth for a simplistic service like this.

Finding: *Weak and easily guessable user account password*

Severity Rating: 6.0

This vulnerability puts a user's business email at risk.
CVSS Base Severity Rating: 4.3 AV:A AC:L PR:N UI:N S:U C:L I:L A:N

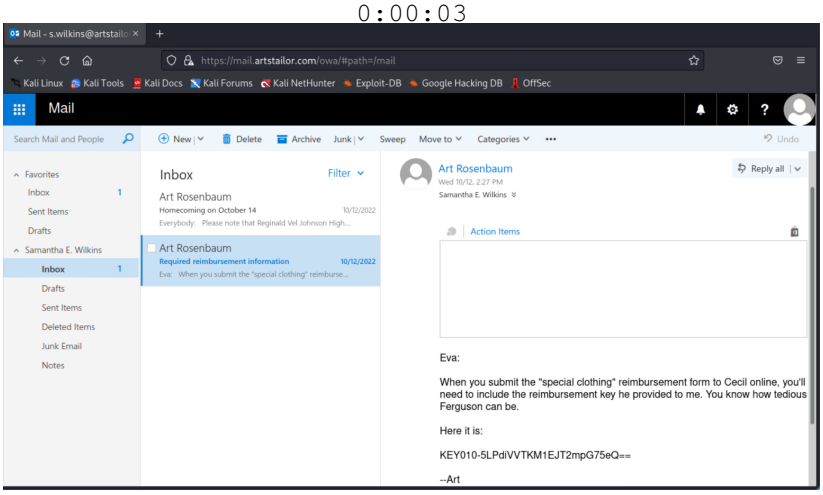
Vulnerability Description

On the `mail.artstailor.com` host, the user 's.wilkins' on the ARTSTAILOR domain has a password that is easily guessable and/or vulnerable to dictionary attacks using common wordlists. As a result, malicious actors may be able to gain access to their email account on the Microsoft OWA instance on the machine.

Confirmation method

Using `atomizer.py` from the *Password Spraying Toolkit*, one can create a password list combining seasons of the year along with year numbers to eventually spray arrive at valid credentials:

```
./atomizer.py <password_list> <user_list> --interval
                                0:00:03
```



Mitigation or Resolution Strategy

Prompt the user s.wilkins to change their password immediately to a more complex string, potentially with special characters and without common verb + year structure.

Finding: *Default credentials on pfSense router admin panel*

Severity Rating: 9.1

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

Vulnerability Description

On the `innerouter.artstailor.com` host (public IP 217.70.184.3), the pfSense credentials for the admin account are unchanged from the default credentials of `admin:pfsense`. This is a crucial vulnerability because it allows attackers complete control over network traffic within the subnet, with attacks such as Man in the Middle and even logging in to unintended services such as RDP now being a possibility if an attacker finds the IP and port combination.

Confirmation method

Simply log on to 217.70.184.3 with the aforementioned credentials on a web browser, and you will now be in the administrator account for the router.

Mitigation or Resolution Strategy

Change the credentials for the administrator ASAP.

Finding: *Vulnerable permissions on a service for a non-administrator user can allow for the creation of an adminsitrator account*

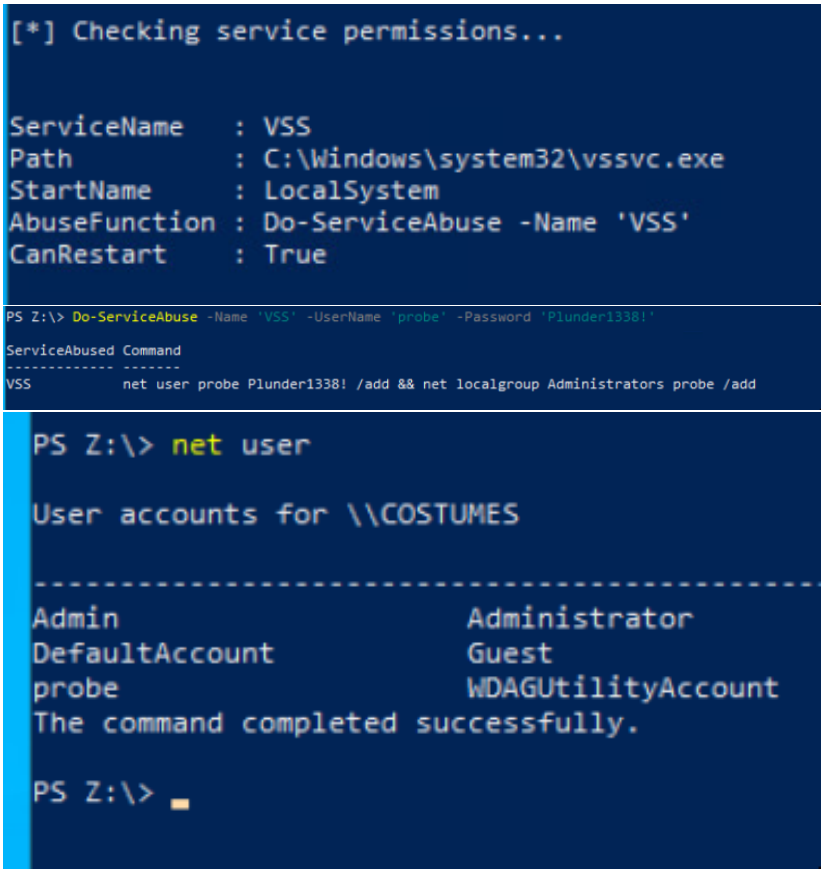
Severity Rating: 8.4

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

Vulnerability Description

On costumes.artstailor.com, non-administrative user 's.wilkins' has permisson to modify the Windows service 'vssvs.exe' such that commands can be executed with SYSTEM-level permissions, meaning an administrator account can be created by a non-administrator user.

Confirmation method



Mitigation or Resolution Strategy

Remove the modification rights that user 's.wilkins' has on this critical Windows program.

Finding: *Weak and commonly exploited password on domain account*

Severity Rating: 4.0

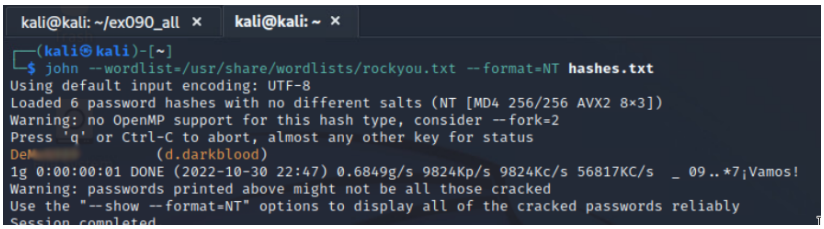
CVSS Base Severity Rating: 4.0 AV:L AC:L PR:N UI:N S:U C:H I:N A:N

Vulnerability Description

User `d.darkblood` on the ARTSTAILOR domain contains a weak, commonly exploited password that is available on a variety of online password lists, or "word lists," that attackers often employ to gain access to accounts.

Should an attacker gain access to the Windows password hashes, such as with a Man-in-the-Middle attack or through exploitation of an individual host on the network, this would allow them to quickly achieve lateral movement to the d.darkblood account.

Confirmation method



Mitigation or Resolution Strategy

As per the NIST SP 800-63B publication, it is good practice to compare user passwords against a common list ("blacklist") of passwords that have been breached many times in the past. In this case, it is recommended that users check if the password they are using for their domain accounts is present in the following list, available online:

https://github.com/praetorian-inc/Hob0Rules/blob/master/wordlists/rockyou.txt.gz

Finding: *Improper storage of private credentials*

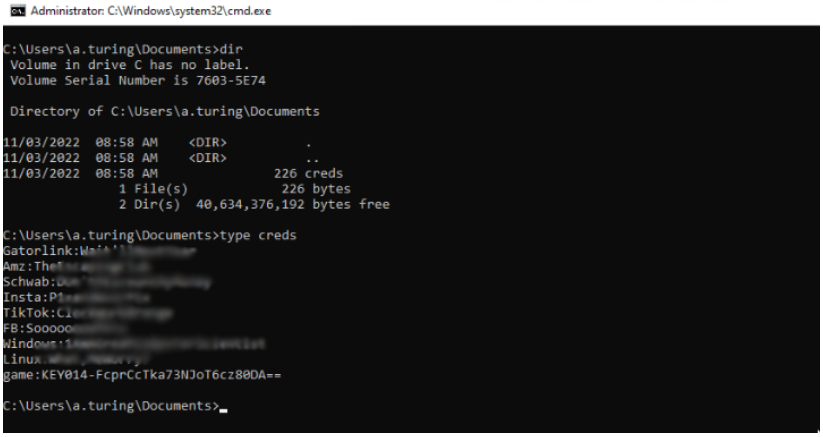
Severity Rating: 6.0

CVSS Base Severity Rating: 6.0 AV:A AC:L PR:H UI:N S:U C:H I:N A:N

Vulnerability Description

On the user a.turing belonging to the BOOKS machine, a list of credentials are stored in a plaintext file for various different accounts, some of potential importance.

Confirmation method



Mitigation or Resolution Strategy

Alert the a.turing user to either opt for an encrypted, secure method of password storage (e.g. a program like KeePassXC), or simply delete the file altogether and leave password-keeping up to official administrators.

Finding: *Corporate login form vulnerable to HTTP downgrade attack with sslstrip Machine in the Middle*

Severity Rating: 6.0

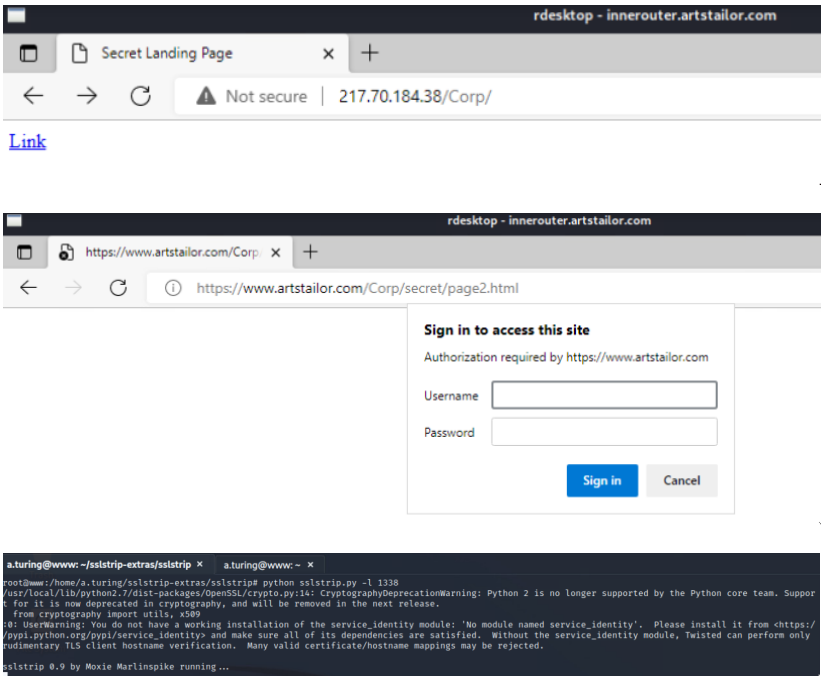
CVSS Base Severity Rating: 6.0 AV:A AC:L PR:N UI:R S:C C:H I:H A:L

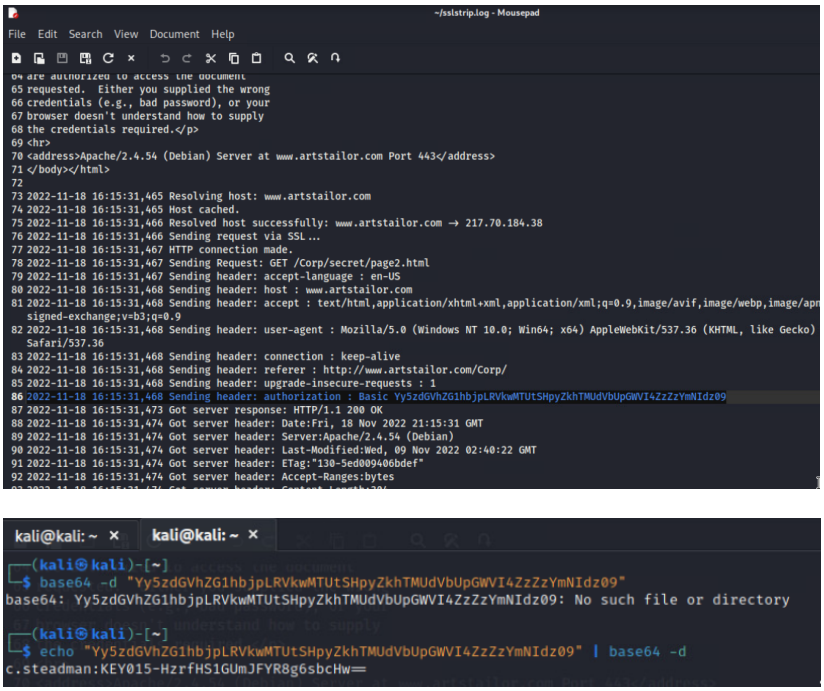
Vulnerability Description

This vulnerability involves the corporate HTTP login page at http://artstailor.com/Corp/, where there is a link leading to an HTTPS (secure) login form. The page containing this link, though, is not secured with HTTPS, and as such, leaves open the possibility for an attacker to set up a man-in-the-middle (MitM) attack that can redirect user to a malicious login page that itself does not use HTTPS, unlike the real one, and sniffs their credentials. This can lead the attacker to gaining the permissions on the company's internal server on whoever falls victim to this attack.

This attack vector is largely mitigated by the use of HTTP Strict Transport Security in modern browers, however, and thus the attack is not as feasible as it once was.

Confirmation method





Mitigation or Resolution Strategy

To remediate this vulnerability, 1) ensure that all corporate network users are using up-to-date browsers that enable HTTP Strict Transport Security, and 2) for best practice, enable HTTPS at the landing with the link leading to the coroprate login form.

Additionally, it may be useful to to enforce having only one Media Access Control (MAC) address per physical network port. This way, an attacker cannot accumulate various different MAC addresses to perform ARP spoofing en-masse.

It is also advisable to have a configuration in a network Intrusion Detection / Prevention System (IDPS) such as Snort (https://www.snort.org/) that checks for suspicious ARP activity, such as a high amount of gratuitous ARP replies

Finding: *Vulnerable version of sudo in combination with illegitimate system binary allows for regular user privilege escalation*

Severity Rating: 8.0

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

Vulnerability Description

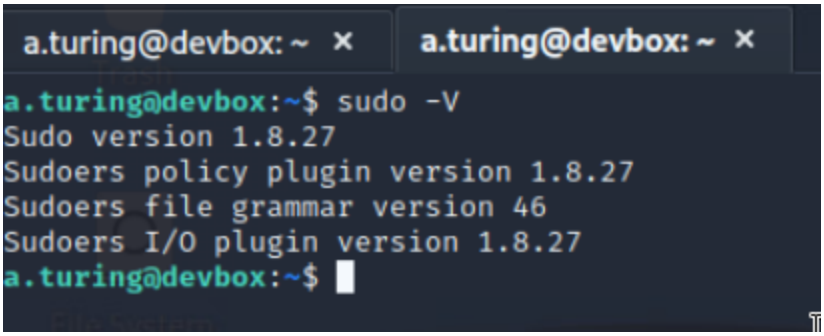
This vulnerability consists of two elements coming together to provide a user a pathway to privilege escalation:

1) The /bin/ps binary on devbox.artstailor.com is edited to execute log the usages of the command, outputting them to a.turing's home directory, upon which the real /bin/ps is then executed (renamed to "realps")

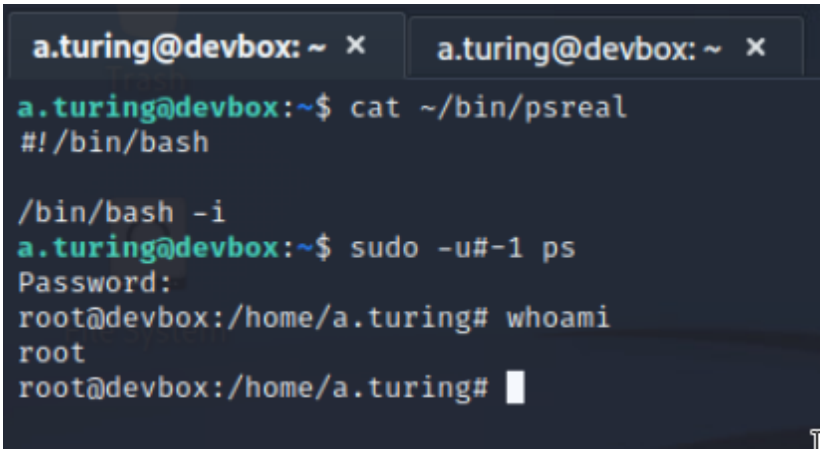
2) With the outdated version of sudo on this machine (see screenshots below) a user may input the `-u-1` argument to execute the specified command they are allowed to execute as per `/etc/sudoers` as root, even if the command in the sudoers file does not specify root as a user they may run the command as.

As a result, a malicious low-permissions user can make a script in a directory they have control over, name it psreal, and change their environment variable to that directory so that when they run "ps", their malicious program gets executed.

Confirmation method



6_ps_sus.png



Mitigation or Resolution Strategy

It is urgent that the correct version of the ps binary gets restored, and that sudo gets updated immediately on devbox.artstailor.com

User a.turing should be inspected for questioning.

Finding: *Network traffic vulnerable to Machine-in-the-Middle attack due to insecure WPAD configuration*

Severity Rating: 5.0

CVSS Base Severity Rating: 5.0 AV:A AC:H PR:N UI:R S:C C:L I:L A:L

Vulnerability Description

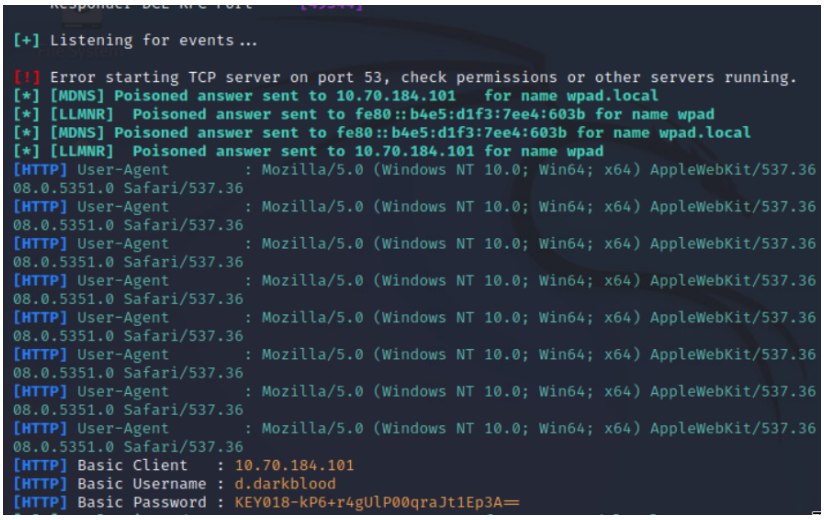
Due to the way that the Web Proxy Auto Discovery protocol (WPAD) is set up on the 10.70.184.0/24 subnet, WPAD network traffic containing a proxy request is able to be answered by a malicious actor, thus leaving any user on the network vulnerable to a Machine-in-the-Middle attack when their host sends out a WPAD request.

Confirmation method

Using the Responder program as follows:

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

the following basic HTTP credentials were captured:



along with the following NTLMv2 hash:

Mitigation or Resolution Strategy

If possible, disable the WPAD service on the appropriate server entirely by following the instructions at https://learn.microsoft.com/en-us/troubleshoot/windows-server/networking/disable-http-proxy-auth-features.

Alternatively, ensure that the correct WPAD host address is specified in the DNS server, so that no other WPAD response may be accepted.

Finding: *EAP configuration on wireless network does not validate AP certificates, leading to potential evil twin attack*

Severity Rating: 6.0

CVSS Base Severity Rating: 6.0 AV:A AC:H PR:N UI:R S:C C:H I:L A:N

Vulnerability Description

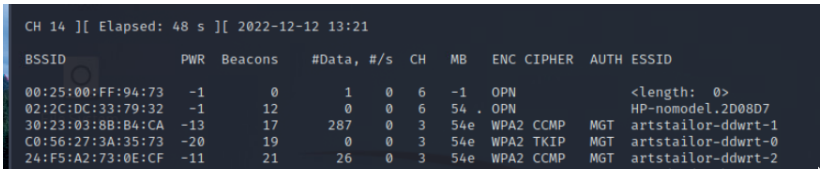
The configuration of Art's Tailor's EAP deployment on their wireless network does not require validation of Access Point (AP) certificates. This creates the possibility for a malicious AP to be configured with the same name as a legitimate AP, that wirless users may subsquently connect to and inadvertently provide their network authentication credentials to.

Confirmation method

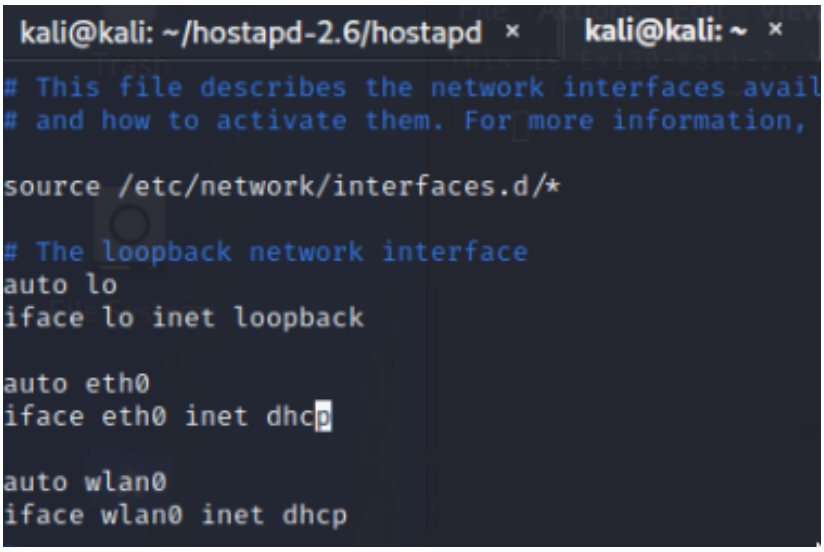
With a monitor-mode capable device, run the following commands to check for the locally available APs from Art's Tailor:

```
sudo airmon-ng check kill
sudo airmon-ng start wlan0
sudo airodump-ng wlan0mon
```

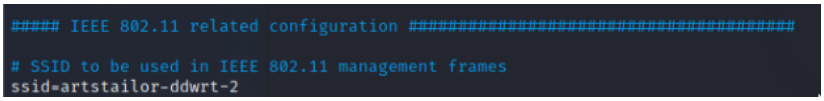
Confirm the channel of the desired AP given by the last command:



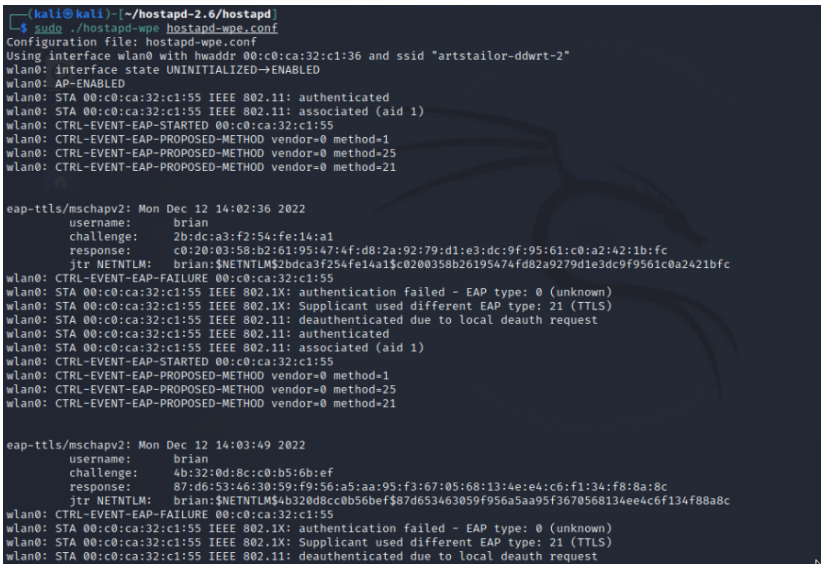
Next, restart the attacker host. run `sudo airmon-ng check kill` once more, and then specify the network interfaces manually in `/etc/network/interfaces`:



Now, c reate a `hostpad-wpe.conf` file with the following:



...and run it as follows:



As seen in the screenshot above, hashed credentials for the legitimate AP were captured. These can be subsequently cracked for an attacker to authenticate with to the real network (see Attack Narrative for details).

Mitigation or Resolution Strategy

Enable AP certificate validation for the current deployment of EAP, and ensure that clients are required to validate said certificate upon connecting.

Consult section 14.4 of EAP-TTLS RFC (https://www.rfc-editor.org/rfc/rfc5281#section-14.4) for more information.

Finding: *Plaintext storage of customer credit card information on local database*

Severity Rating: 7.5

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

Vulnerability Description

The MySQL database at `db.artstailor.com` stores customer credit card numbers in plain text in the `ccard` table within the `customerdb` database.

This is in violation of the Payment Card Industry Data Security (PCI-DSS) standards, which states that Primary Account Numbers (the 16 digits of a card) must be encrypted if they are to be stored on an enterprise database – this is assuming there is a legitimate need for this data to be captured and stored in the first place, which there may not be.

Confirmation method

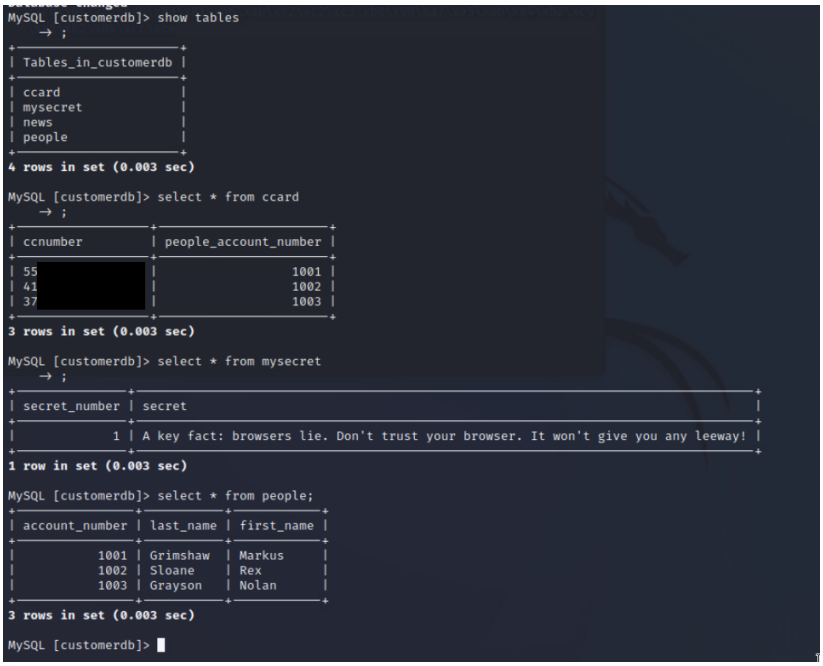
With the credentials for the database administrator account on hand, one may connect with the following command:

```
mysql -h db.artstailor.com -u db_admin_token -p"<admin
                    password>"
```

Upon connecting, run

```
use customerdb;
select * from ccard;
```

which yields the following result (card numbers censored):



Mitigation or Resolution Strategy

Firstly, determine if there is a legitimate legal (consult the legal team) or business reason to store customer credit card information locally. For example, if customers do not usually make recurring purchases, then storing their card information may not be necessary at all, and the risk of PCI-DSS fines may be avoided entirely.

If there is a legitimate legal or business need to store credit card numbers, ensure this data is only stored in an encrpyted form, with vetted industry encryption libraries as opposed to in-house solutions.

In addition, follow all requirements for credit card information specified in the latest PCI-DSS standard, which can be found at

https://listings.pcisecuritystandards.org/documents/PCI-DSS-v4_0.pdf

https://listings.pcisecuritystandards.org/documents/PCI-DSS-v4_0.pdf

Finding: *Hardcoded MySQL database credentials stored in Android APK binary*

Severity Rating: 4.0

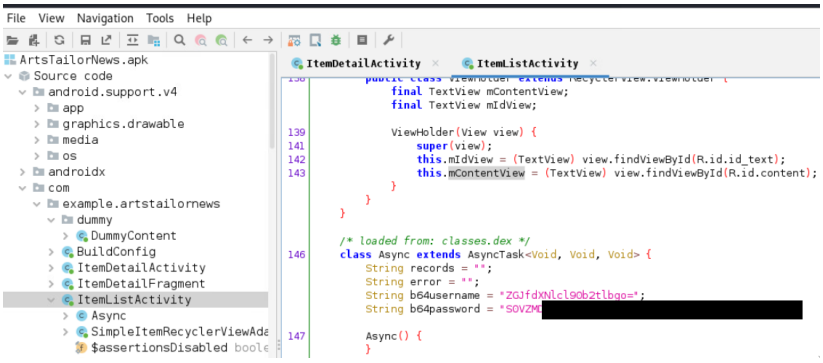
CVSS Base Severity Rating: 4.0 AV:N AC:L PR:N UI:N S:U C:L I:N A:N

Vulnerability Description

Within the `ItemListActivity` class file in the `com/example/artstailornews` directory of the Android application that Art's Tailor Shoppe is developing, there are hardcoded credentials that can be used to access the MySQL database at `db.artstailor.com`.

Confirmation method

First, decompile the APK binary with an Android decompiler such as jadx. Then, navigate to the `com/example/artstailornews/` directory, in which the `ItemListActivity` class file is contained with the aforementioned credentials:



Mitigation or Resolution Strategy

Do not have the Android app directly communicate with the database. Instead, interface these requests through an internal host, that then authenticates to the SQL server, and passes along the returned information to the Android application on the user end.