Christian Stevens

Client Interview - Summary Report Template

Designed to capture detailed information about an organization's network and systems infrastructure.

Organization Information

1. **Organization Name:** [redacted]
2. **Contact Person:**

Scope and Drivers

1. **Target Systems:** https://demo-compliance.[redacted]/
2. **Testing Drivers:**
3. **Business Drivers:**

Scheduling and Logistics

1. **Start Date:6/25/24**
2. **End Date:7/9/24**
3. **Expected Duration: 3 Weeks**
4. **Outside Working Hours:** No

System and Infrastructure

1. **Critical Systems on Target IP Addresses:** No
2. **Public IP Addresses:** 172.67.204.29; 104.21.52.215
3. **Cloud Environment:** AWS
4. **Company Domain Names:** https://demo-compliance.[redacted]

Additional Information

1. **Known Vulnerabilities**
    1. SSL Medium Strength Cipher Supported (SWEET32).
    2. TLS Version 1.0 Protocol Detection & 1.1 Deprecated Protocol.
    3. Email Address Disclosed
    4. Credit Card Numbers Disclosed
2. **Security Documentation:** N/A

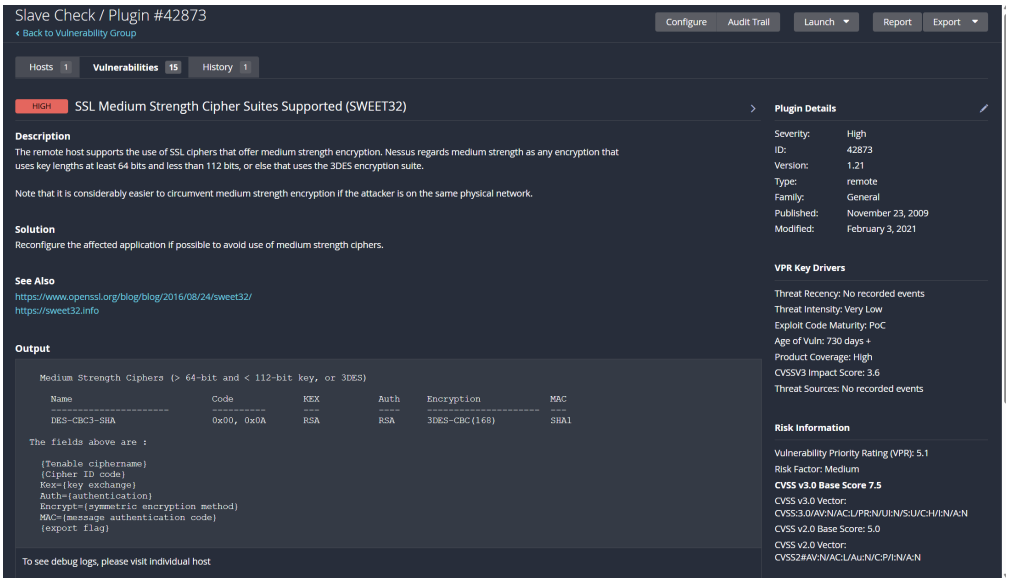
Testing Exclusions

1. **Specific systems or functionalities:** (Explain why these are excluded)
2. **Data or environments:** (Explain why these are excluded)

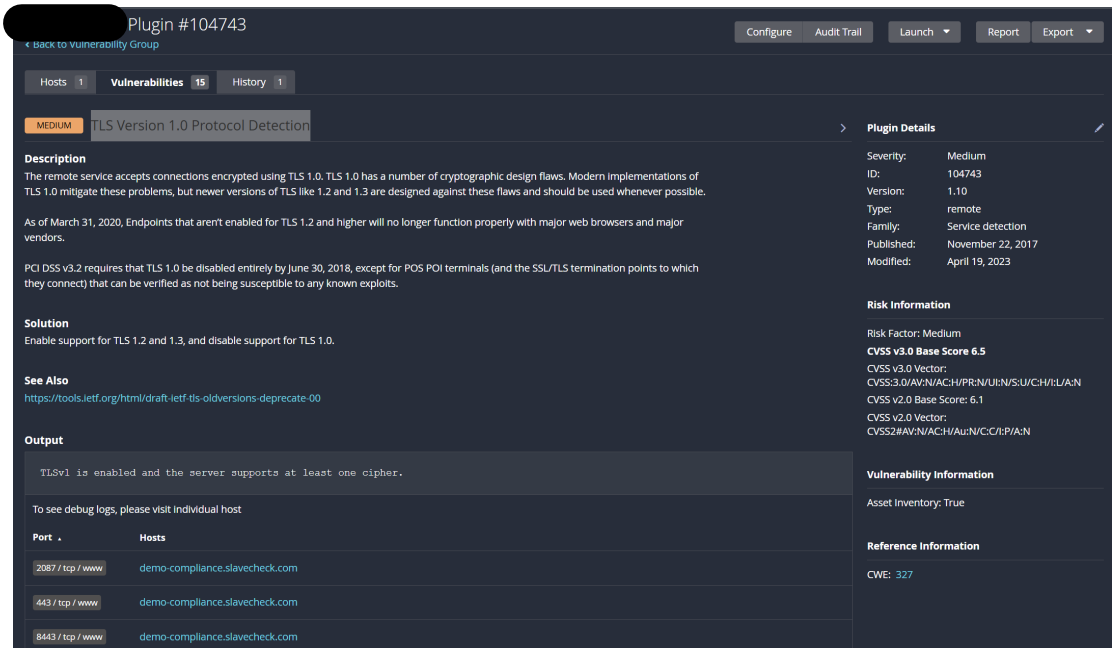
Nessus Findings

All vulnerabilities (including those classified as Low):

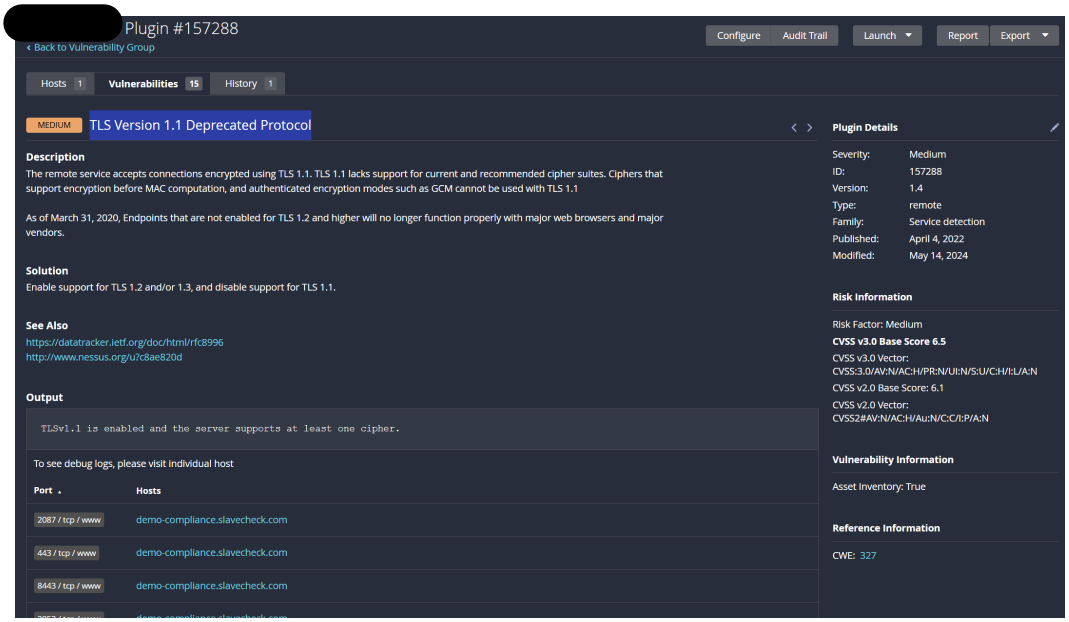
- URL: demo-compliance.[redacted]
- Nessus Plugin ID: 42873
- Vulnerability name: SSL Medium Strength Cipher Supported (SWEET32)
- Description: The remote host supports the use of SSL ciphers that offer medium-strength encryption. Nessus regards medium-strength encryption as any encryption that uses key lengths of at least 64 bits and less than 112 bits, or that uses the 3DES encryption suite.
- CVE: 2016-2183
- CVSS Score: 7.5
- Likelihood: Medium Risk
- Consequence & Impact: Data exposure, man-in-the-middle attacks, loss of confidentiality and integrity, compliance issues, decreased trust, and operational impact.
- Recommendation: Disable 64-bit Block Ciphers, Use Stronger Ciphers, Keep Systems Updated, and Audit and Test Configurations.
- Screenshots of scan results:



- URL 2.: demo-compliance.[redacted]
- Nessus Plugin ID: 104743
- Vulnerability name: TLS Version 1.0 Protocol Detection
- Description: The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has several cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.
- CVE:
- CVSS Score: 6.5
- Likelihood: Medium Risk
- Consequence & Impact: Weak Encryption can allow attackers to decrypt data transmitted between a client and server. This can expose the client to Beast Attacks, Poodle Attacks, and Downgrade Attacks.
- Recommendations: Disable TLS 1.0 Reconfiguration Servers, update all software, conduct regular audits, and educate and train staff.
- Screenshots of scan results:



- URL 3.: demo-compliance.[redacted]
- Nessus Plugin ID: 157288
- Vulnerability name: TLS Version 1.1 Deprecated Protocol
- Description: The remote service accepts connections encrypted using TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation and authenticated encryption modes such as GCM cannot be used with TLS 1.1
- CVE:
- CVSS Score: 6.5
- Likelihood: Medium Risk
- Consequence & Impact: Weak Encryption can allow attackers to decrypt data transmitted between a client and server. This causes compliance issues and will have operational implications.
- Recommendations: Disable TLS 1.1, Update software, educate and train staff, and implement stronger protocols.
- Screenshots of scan results:



“Informational” findings (list)

- HTTP (Multiple Issues)
- IETF Md5 (Multiple Issues)
- Web Server (Multiple Issues)
- Service Detection
- Nessus SYN Scanner
- SSL Certificate Chain Contains Certificates Expiring Soon
- Common Platform Enumeration (CPE)

- Device Type
- Nessus Scan Information
- OS Identification
- TCP/IP Timestamp Supported
- Traceroute Information

NMAP scan results

Nmap Findings

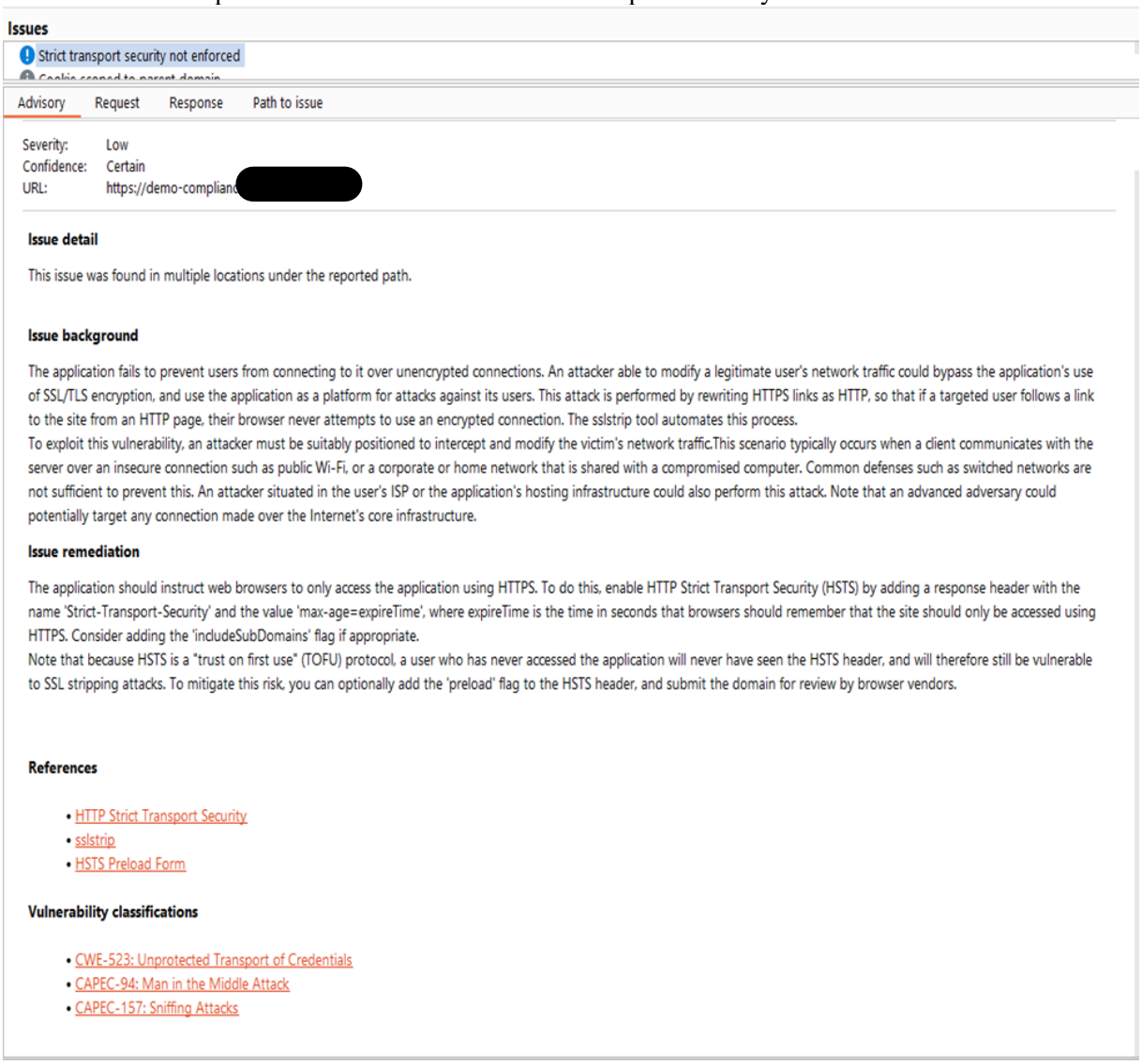
- Hostname: demo-compliance.[redacted]
- IP1: 104.21.52.215
- Ports Open:443/80/8080/8443
  - TCP:104.21.52.215
  - UDP:
- Nmap command used: sudo nmap –script vuln -v demo-compliance.slavecheck.com
- Observations: phpMyAdmin allows remote attackers to include local users' denial of service
  CVE-2005-3299
- Screenshots of scan results:

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap --script vuln -v demo-compliance
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-19 01:31 UTC
NSE: Loaded 105 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 01:31
Completed NSE at 01:31, 10.01s elapsed
Initiating NSE at 01:31
Completed NSE at 01:31, 0.00s elapsed
Initiating Ping Scan at 01:31
Scanning demo-compliance.slavecheck.com (104.21.52.215) [4 ports]
Completed Ping Scan at 01:31, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:31
Completed Parallel DNS resolution of 1 host. at 01:31, 0.18s elapsed
Initiating SYN Stealth Scan at 01:31
Scanning demo-compliance.slavecheck.com (104.21.52.215) [1000 ports]
Discovered open port 443/tcp on 104.21.52.215
Discovered open port 80/tcp on 104.21.52.215
Discovered open port 8080/tcp on 104.21.52.215
Discovered open port 8443/tcp on 104.21.52.215
Completed SYN Stealth Scan at 01:31, 4.75s elapsed (1000 total ports)
NSE: Script scanning 104.21.52.215.
Initiating NSE at 01:31
Stats: 0:07:59 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE: Active NSE Script Threads: 1 (1 waiting)
NSE Timing: About 99.75% done; ETC: 01:39 (0:00:01 remaining)
Stats: 0:08:07 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE: Active NSE Script Threads: 1 (1 waiting)
NSE Timing: About 99.75% done; ETC: 01:39 (0:00:01 remaining)
Stats: 0:08:18 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE: Active NSE Script Threads: 1 (1 waiting)
NSE Timing: About 99.75% done; ETC: 01:39 (0:00:01 remaining)
Stats: 0:08:19 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE: Active NSE Script Threads: 1 (1 waiting)
NSE Timing: About 99.75% done; ETC: 01:39 (0:00:01 remaining)
Stats: 0:08:20 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE: Active NSE Script Threads: 1 (1 waiting)
NSE Timing: About 99.75% done; ETC: 01:39 (0:00:01 remaining)
Stats: 0:12:39 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE: Active NSE Script Threads: 1 (1 waiting)
NSE Timing: About 99.75% done; ETC: 01:44 (0:00:02 remaining)
Completed NSE at 02:01, 1761.63s elapsed
Initiating NSE at 02:01
Completed NSE at 02:01, 0.02s elapsed
Nmap scan report for demo-compliance (104.21.52.215)
Host is up (0.0092s latency).
Other addresses for demo-compliance.slavecheck.com (not scanned): 172.67.204.29 2606:4700:3037::6815:34d7 2606:4700:3034::ac43:cc1d
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
443/tcp  open  https
| http-phpmyadmin-dir-traversal:
|   VULNERABLE:
|   phpMyAdmin grab_globals.lib.php subform Parameter Traversal Local File Inclusion
|     State: UNKNOWN (unable to test)
|     IDs:  CVE:CVE-2005-3299
|       PHP file inclusion vulnerability in grab_globals.lib.php in phpMyAdmin 2.6.4 and 2.6.4-pl1 allows remote attackers to include local files via the $__redirect parameter, po
ssibly involving the subform array.
|
|     Disclosure date: 2005-10-nil
|     Extra information:
|       ../../../../../etc/passwd :
|   <!doctype html><html lang="en"><head><meta charset="utf-8" /><link rel="icon" href="/favicon.png" /><meta name="viewport" content="width=device-width,initial-scale=1" /><meta
name="theme-color" content="#000000" /><meta name="description" content="Slave check web app" /><link rel="apple-touch-icon" href="/logo192.png" /><link rel="manifest" href="/mani
fest.json" /><title>SlaveCheck</title><script defer="defer" src="/static/js/main.8c767991.js"></script><link href="/static/css/main.065ffb64.css" rel="stylesheet"></head><body><no
script>You need to enable JavaScript to run this app.</noscript><div id="root"></div><script>(function(){function c(){var b=a.contentDocument||a.contentWindow.document;if(b){var d
=b.createElement('script');d.innerHTML="window.__CF$cv$params={r:'8a56f41e4db66183',t:'MTcyMTM1MjcwMy4wMDAwMDA='};var a=document.createElement('script');a.nonce='';a.src='/cdn-cgi
/challenge-platform/scripts/jsd/main.js';document.getElementsByTagName('head')[0].appendChild(a);";b.getElementsByTagName('head')[0].appendChild(d)}}if(document.body){var a=docume
nt.createElement('iframe');a.height=1;a.width=1;a.style.position='absolute';a.style.top=0;a.style.left=0;a.style.border='none';a.style.visibility='hidden';document.body.appendChil
d(a);if('loading'!==document.readyState)c();else if(window.addEventListener)document.addEventListener('DOMContentLoaded',c);else{var e=document.onreadystatechange||function(){};do
cument.onreadystatechange=function(b){e(b);'loading'!==document.readyState&&(document.onreadystatechange=e,c())}}})();</script></body></html>
|       References:
|         https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3299
|_        http://www.exploit-db.com/exploits/1244/
|_http-vuln-cve2017-1001000: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-majordomo2-dir-traversal: ERROR: Script execution failed (use -d to debug)
| http-vuln-cve2010-0738:
|_  /jmx-console/: Authentication was not required
| http-enum:
|   /blog/: Blog
|   /weblog/: Blog
|   /weblogs/: Blog
|   /wordpress/: Blog
|   /wiki/: Wiki
|   /mediawiki/: Wiki
|   /wiki/Main_Page: Wiki
|   /tikiwiki/: Tikiwiki
|   /cgi-bin/mj_wwwusr: Majordomo2 Mailing List
|   /majordomo/mj_wwwusr: Majordomo2 Mailing List
|   /j2ee/examples/servlets/: Oracle j2ee examples
|   /j2ee/examples/jsp/: Oracle j2ee examples
|   /dsc/: Trend Micro Data Loss Prevention Virtual Appliance
|   /reg_1.htm: Polycom IP phone
|   /adr.htm: Snom IP Phone
|   /line_login.htm?l=1: Snom IP Phone
```

BurpSuite Findings

All vulnerabilities (including those classified as Low):

- URL:: https://demo-compliance[redacted]
- Vulnerability name: Strict Transport Security Not Enforced
- Description: The application fails to prevent users from connecting to it over unencrypted connections.
- Vulnerability Classification: Low
- Risk: CWE-523
- Confidence: Certain
- Solution (Recommendation):: Enable HTTP Strict Transport Security (HSTS) by adding a response header with name ‘Strict-Transport-Security’.



Issues

Strict transport security not enforced

Advisory | Request | Response | Path to issue

Severity: Low
Confidence: Certain
URL: https://demo-complianc[redacted]

Issue detail

This issue was found in multiple locations under the reported path.

Issue background

The application fails to prevent users from connecting to it over unencrypted connections. An attacker able to modify a legitimate user's network traffic could bypass the application's use of SSL/TLS encryption, and use the application as a platform for attacks against its users. This attack is performed by rewriting HTTPS links as HTTP, so that if a targeted user follows a link to the site from an HTTP page, their browser never attempts to use an encrypted connection. The sslstrip tool automates this process.
To exploit this vulnerability, an attacker must be suitably positioned to intercept and modify the victim's network traffic.This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

Issue remediation

The application should instruct web browsers to only access the application using HTTPS. To do this, enable HTTP Strict Transport Security (HSTS) by adding a response header with the name 'Strict-Transport-Security' and the value 'max-age=expireTime', where expireTime is the time in seconds that browsers should remember that the site should only be accessed using HTTPS. Consider adding the 'includeSubDomains' flag if appropriate.
Note that because HSTS is a "trust on first use" (TOFU) protocol, a user who has never accessed the application will never have seen the HSTS header, and will therefore still be vulnerable to SSL stripping attacks. To mitigate this risk, you can optionally add the 'preload' flag to the HSTS header, and submit the domain for review by browser vendors.

References

- HTTP Strict Transport Security
- sslstrip
- HSTS Preload Form

Vulnerability classifications

- CWE-523: Unprotected Transport of Credentials
- CAPEC-94: Man in the Middle Attack
- CAPEC-157: Sniffing Attacks

“Informational” findings (list):

- Cookie scoped to parent domain

- Email address disclosed
- Credit card numbers disclosed

Additional Observations

- [Any additional notes that the assessor finds relevant]
- .

Preliminary Conclusions

- The website's overall health is not at tremendous risk; most seem outdated.
- An upgrade may be made if it meets the company’s budget.

False positives

- The credit card information pulled up seems like a false positive as it looks more like a phone number. Also, SlaveCheck does not provide those services.

Next Steps

- Check each code that pulled up in the scans and access the risk associated and make sure that they are actual risk
- Discuss findings with teammates and put together the presentation for the client.

Appendix: Documenting vulnerabilities - explanation

- **Vulnerability**: Enter the descriptive name assigned to the vulnerability by the testing tool (Nessus, OpenVas, etc.). This name should clearly explain the issue.
- **Description**: Provide a brief description of the vulnerability. This information can often be found in the testing tool's output details section. The description should explain the vulnerability and how it can be exploited.
- **Risk Rating:** Record the overall risk score assigned by the testing tool. This score typically combines the likelihood and consequence of the vulnerability
- **Likelihood:** In this section, students should **estimate** the likelihood of the vulnerability being exploited. They can consider factors such as:
    a. **Prevalence of exploit code:** Is there readily available exploit code for this vulnerability?
    b. **Exploit difficulty:** How technically challenging is it to exploit this vulnerability?
    c. **Value of the target:** Is your system or data a high-value target for attackers?

    d. **Patch availability:** Is a patch available to fix the vulnerability? Students should use their understanding of the vulnerability and the environment to make an informed judgment about the likelihood of exploitation.
- **Consequence:** In this section, students should analyze the **potential consequences** of exploiting the vulnerability. They can consider the same factors mentioned for Impact but with a focus on the severity of the potential damage:
    a. **Confidentiality:** What sensitive data could be exposed if exploited?
    b. **Integrity:** How critical could the systems or data be compromised?
    c. **Availability:** What would impact a system or service outage?
    d. **Financial Loss:** Could the vulnerability lead to significant financial losses? By considering these factors, students can assess the potential severity of the consequences if the vulnerability is exploited.
- **CVE:** Enter the unique identifier for the specific vulnerability identified by the testing tool (Nessus, OpenVas, etc.). This will typically be a code like CVE-2023-XXXX
- **CVSS:** Enter the CVSS score assigned by the testing tool. The CVSS score (0.0-10.0) reflects the severity of the vulnerability
- **Recommendation:** Provide recommendations for fixing the vulnerability. This may involve patching the system, disabling the affected service, or taking other steps to mitigate the risk