Senior Design II

Professor Michael Chu

Group 1: Gurpinder Chattha, Dalton Grovatt, Randy Truong, Isaac Stafford, Danial Malik, Shamme Alam, Salwa Ghaffar

Table of Contents

[Introduction: 2](#_Toc1271524092)

[Network Team 3](#_Toc1050402448)

[Tools: 3](#_Toc1729348160)

[Scope: 3](#_Toc1694223623)

[Findings: 5](#_Toc1315347280)

[Database Team 7](#_Toc826611524)

[Data Tools 7](#_Toc1740049318)

[Nikto 7](#_Toc1208944698)

[Skipfish 8](#_Toc1140985071)

[ZAP (Zed Attack Proxy) 8](#_Toc546421521)

[Data Results 9](#_Toc1070084310)

[High Vulnerabilities: 9](#_Toc45654784)

[Medium Vulnerabilities: 10](#_Toc882215148)

[Low Vulnerabilities: 11](#_Toc1603054667)

[CWE 12](#_Toc2073202798)

### Introduction:

In this report, we will detail the entirety of our comprehensive vulnerability scan launched in January 2022, including the tools used during the scanning, the scanning process, and the results of the scan. The report will be divided between the two teams of this group, the Network team, and the Database team.

## Network Team

### Tools:

In order to conduct the vulnerability scan, the team spent the majority of the first term finding an adequate tool that was customizable in scanning measures, up to date in common vulnerabilities and signatures, free, and reliable. Shodan is an internet-facing device search engine, which is very useful for finding a large number of internet-facing devices that may contain known exploitable vulnerabilities and was the first step we took to get a good scope of Drexel.

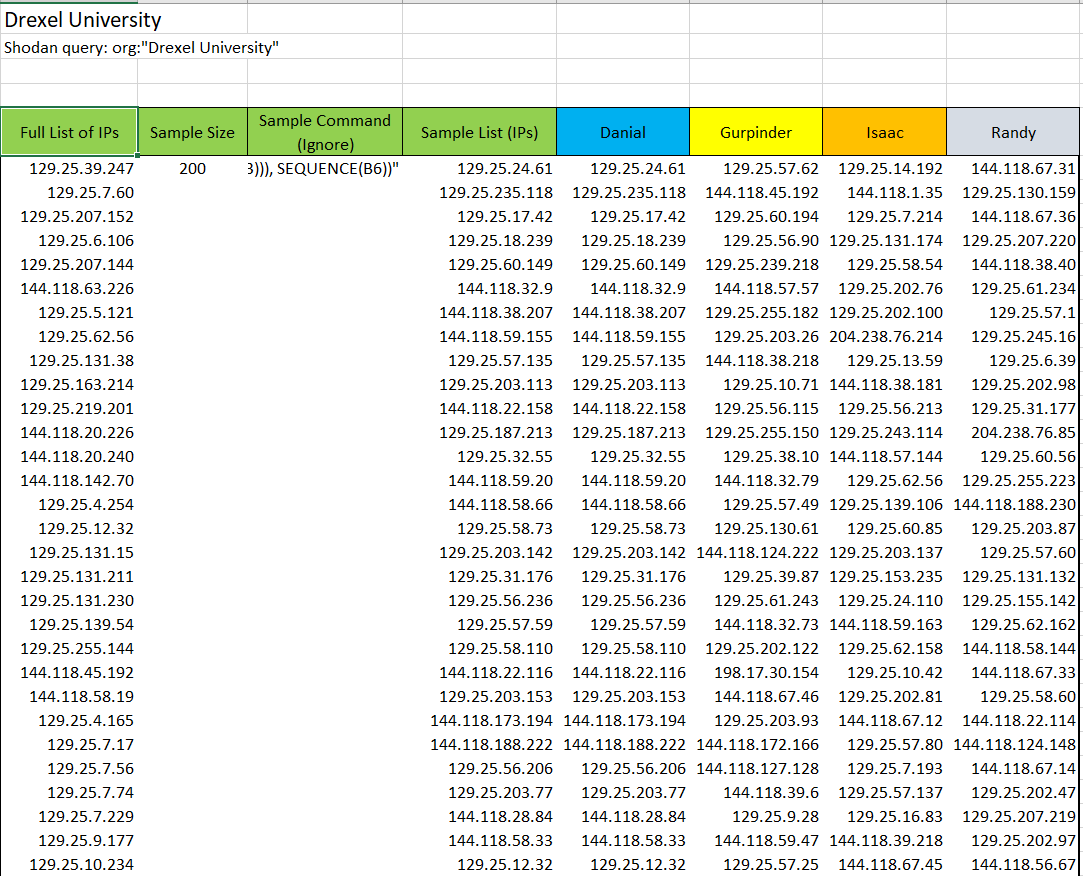
The first tool which the team tried was Tenable’s Nessus, which is a highly reliable enterprise-level vulnerability scanner with customization, user-friendly interfaces, and accurate vulnerability signatures. Nessus was unfortunately not scalable for the operation without paying for the full version; for instance, each individual of the team had the ability to scan up to 16 IP addresses prior to being restricted from further scans. Thus, the team could not use the tool as the full vulnerability scan would be over 200 IP addresses.

The next tool of interest was Legion. Legion was a contender as it was free, generally customizable, and simple to use; however, it was discovered that Legion’s internal vulnerability database was out-of-date and would not pick up on issues unrelated to SSH (secure shell).

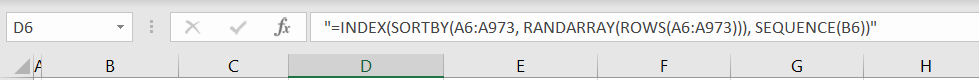
The final tool found and ultimately used was OpenVAS by Greenbone. OpenVAS, although the most difficult to install out of the three and with a much less user-friendly interface, was a free, up-to-date, customizable, and scalable vulnerability scanner that would accomplish the task at hand. The team struggled with the tool’s “scanning phase”, often freezing or stopping at random; however, with some tweaking and by limiting the number of IP addresses per scan, the team acquired the results it needed for analysis.

### Scope:

The scan included a randomly sampled list from a comprehensive Shodan query of the university’s externally facing hosts. The sampled list was derived from compiling all externally facing IP addresses in Excel (left-most column):



The following command was then issued to sample from the list of 968 IP addresses, which simply sampled from full list of IPs to derive a sampled list:



The fourth column (Sampled List (IPs)) contained the 200 IP addresses that were distributed amongst certain individuals to scan, depending on their role in the team.

#### Findings:

There were multiple vulnerabilities detected across Drexel’s network, with 101 being categorized as critical or high, 429 as medium and 31 as low. This large volume of vulnerabilities is all saved as part of our findings, but only certain findings will be called out in this report.  
  
High Severity:

* PhpMyAdmin End of Life Detection - An EOL version of phpMyAdmin is not receiving any security updates from the vendor. Unfixed security vulnerabilities might be leveraged by an attacker to compromise the security of this host.
* PhpMyAdmin SQL Injection Vulnerability - A vulnerability was reported where a specially crafted database name can be used to trigger an SQL injection attack through the designer feature. This affects phpMyAdmin versions 4.8.6 or lower.
* jQuery End of Life Detection - An EOL version of jQuery is not receiving any security updates from the vendor. Unfixed security vulnerabilities might be leveraged by an attacker to compromise the security of this host.
* Apache HTTP Server Multiple Vulnerabilities – An Apache HTTP server was found with an outdated version, which exposes it to the following vulnerabilities: CVE-2022-22719, CVE-2022-22720, CVE-2022-22721, and CVE-2022-23943. These vulnerabilities are all high severity and could lead to buffer overflow attacks, HTTP request smuggling, and even read/write boundary issues.

Medium Severity:

* SSL/TLS Missing ‘Secure’ Cookie Attribute - The flaw is due to cookie is not using 'secure' attribute, which allows cookie to be passed to the server by the client over non-secure channels (http) and allows attacker to conduct session hijacking attacks.
* WordPress iframe Plugin XSS Vulnerability - The WordPress plugin iframe is prone to a cross-site scripting (XSS) vulnerability. Successful exploitation would allow an attacker to inject arbitrary HTML and JavaScript into the site.
* Deprecated SSLv2 and SSLv3 Protocol - It was possible to detect the usage of the deprecated SSLv2 and/or SSLv3 protocol on this system. An attacker might be able to use the known cryptographic flaws to eavesdrop the connection between clients and the service to get access to sensitive data transferred within the secured connection. Furthermore, newly uncovered vulnerabilities in these protocols won't receive security updates anymore.

Low Severity:

* Weak MAC Algorithms Supported (SSH) - The remote SSH server is configured to allow weak MAC algorithms. The remote SSH server supports the following weak client-to-server MAC algorithms:  
    
  hmac-md5  
  hmac-md5-96  
  hmac-md5-96-etm@openssh.com  
  hmac-md5-etm@openssh.com  
  hmac-sha1-96  
  hmac-sha1-96-etm@openssh.com  
    
  The remote SSH server supports the following weak server-to-client MAC algorithms:  
    
  hmac-md5  
  hmac-md5-96  
  hmac-md5-96-etm@openssh.com  
  hmac-md5-etm@openssh.com  
  hmac-sha1-96  
  hmac-sha1-96-etm@openssh.com
* Oracle MySQL Server Security Update - An unspecified error exists in the 'MySQL Server' component via unknown vectors related to 'Connection' sub-component. Successful exploitation will allow a remote attacker to affect confidentiality via unknown vectors.

## Database Team

To scan Drexel for data vulnerability, our initial attempt was to utilize Windows OS to run Nessus and Shodan, similar to the Networking team, however Drexel University uses Microsoft ASP hosting, which offers a very secure web.

The team used Shodan to look for SQL vulnerabilities and then used Nessus to scan the IP addresses they found. There were no high or medium vulnerabilities found after the Nessus scan. After performing a few scans and not receiving any useful data, the team started to look into alternative options. After doing research, the data team determined that utilizing Windows would not yield satisfactory results, therefore the team decided to shift to Kali Linux and conduct the scans there.

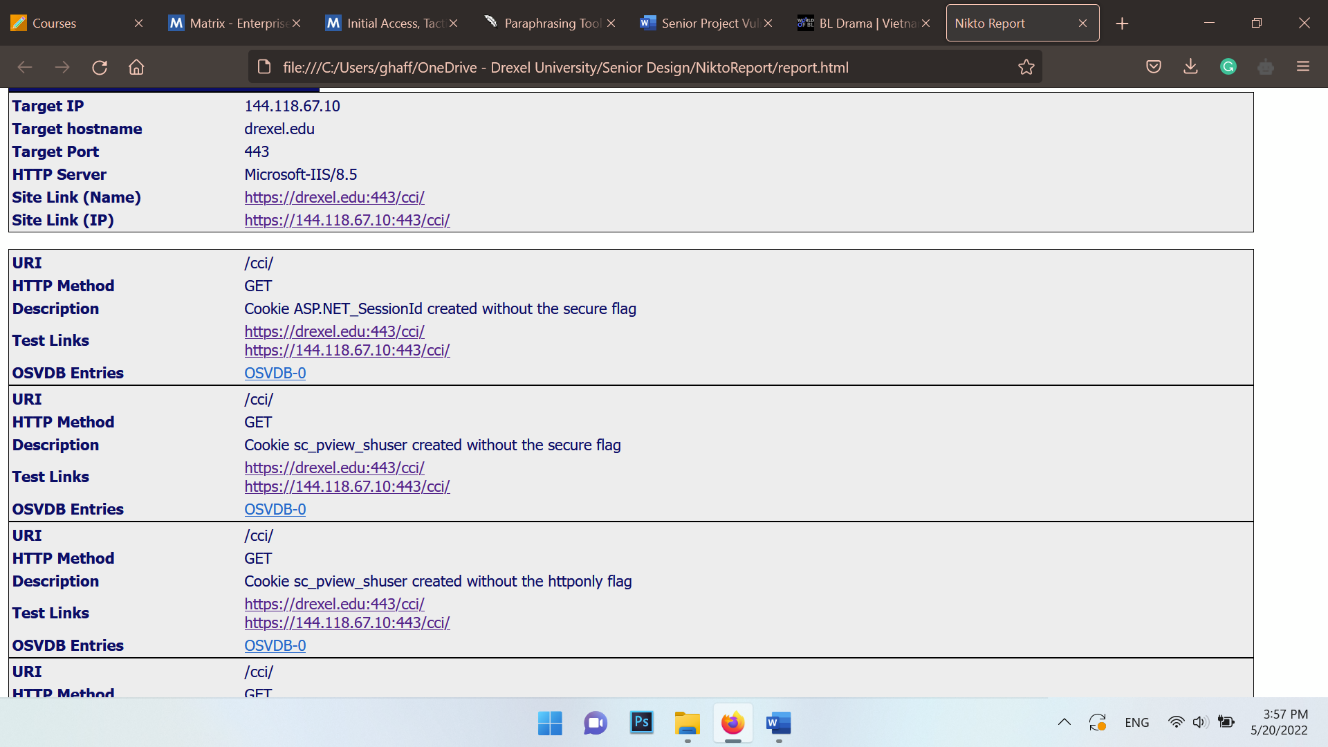
In addition to Nessus and Shodan, the team noticed that there are a few other tools we may use throughout our research. We went with Nikto and Skipfish, which are both Linux-based apps. We also downloaded the ZAP scanning application for Linux.

* Nikto – Nikto is a web server vulnerability scanner. This tool was written by Chris Sullo and David Lodge. Nikto automates the process of scanning web servers for out-of-date and unpatched software as well as searching for dangerous files that may reside on web servers.
* Skipfish – Skipfish is an active web application security reconnaissance tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The resulting map is then annotated with the output from a number of active (but hopefully non-disruptive) security checks. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments.
* ZAP – Zed Attack Proxy (ZAP) is a free, open-source penetration testing tool. ZAP is designed specifically for testing web applications and is both flexible and extensible. At its core, ZAP is what is known as a “man-in-the-middle proxy.” It stands between the tester’s browser and the web application so that it can intercept and inspect messages sent between browser and web application, modify the contents if needed, and then forward those packets on to the destination. It can be used as a stand-alone application, and as a daemon process.

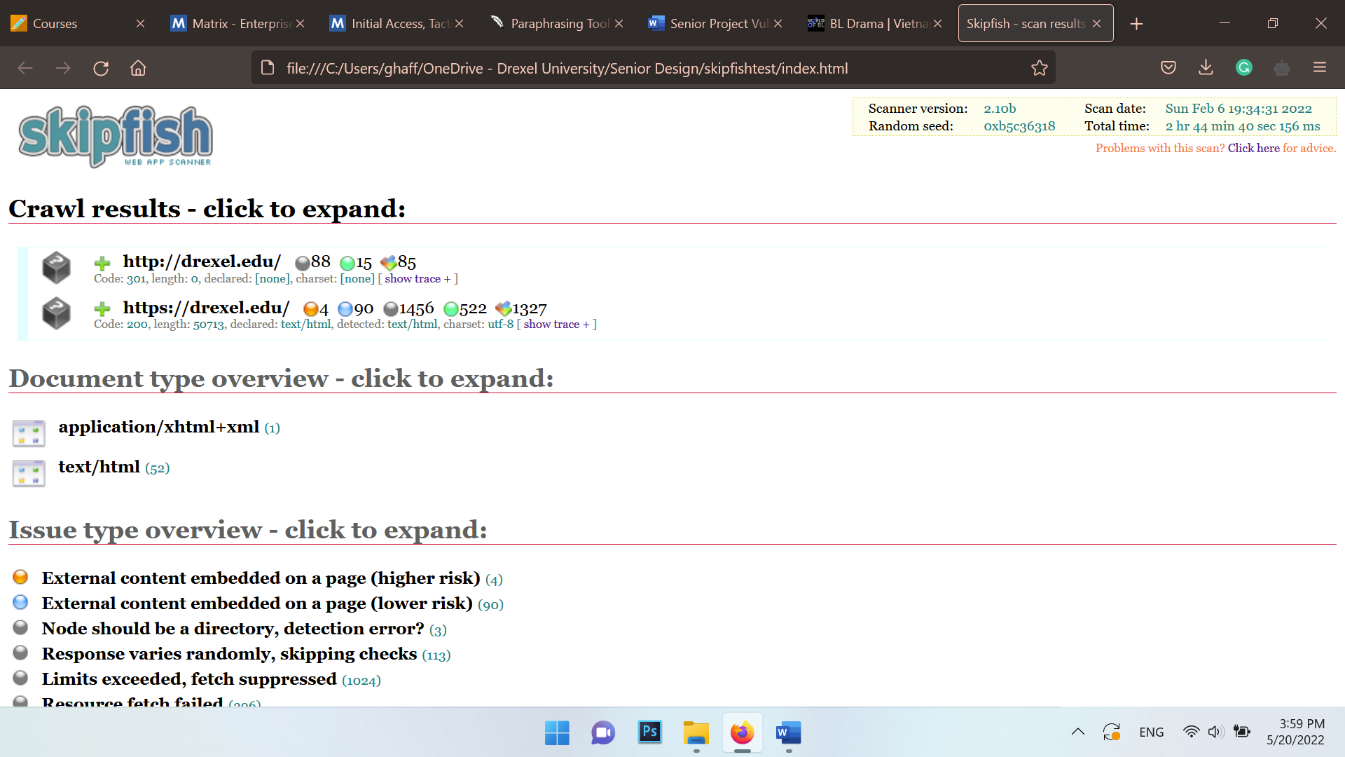
### Data Tools

After running a scan over three tools Nikto, Skipfish, and ZAP the team was able to get a good number of results, especially from ZAP.

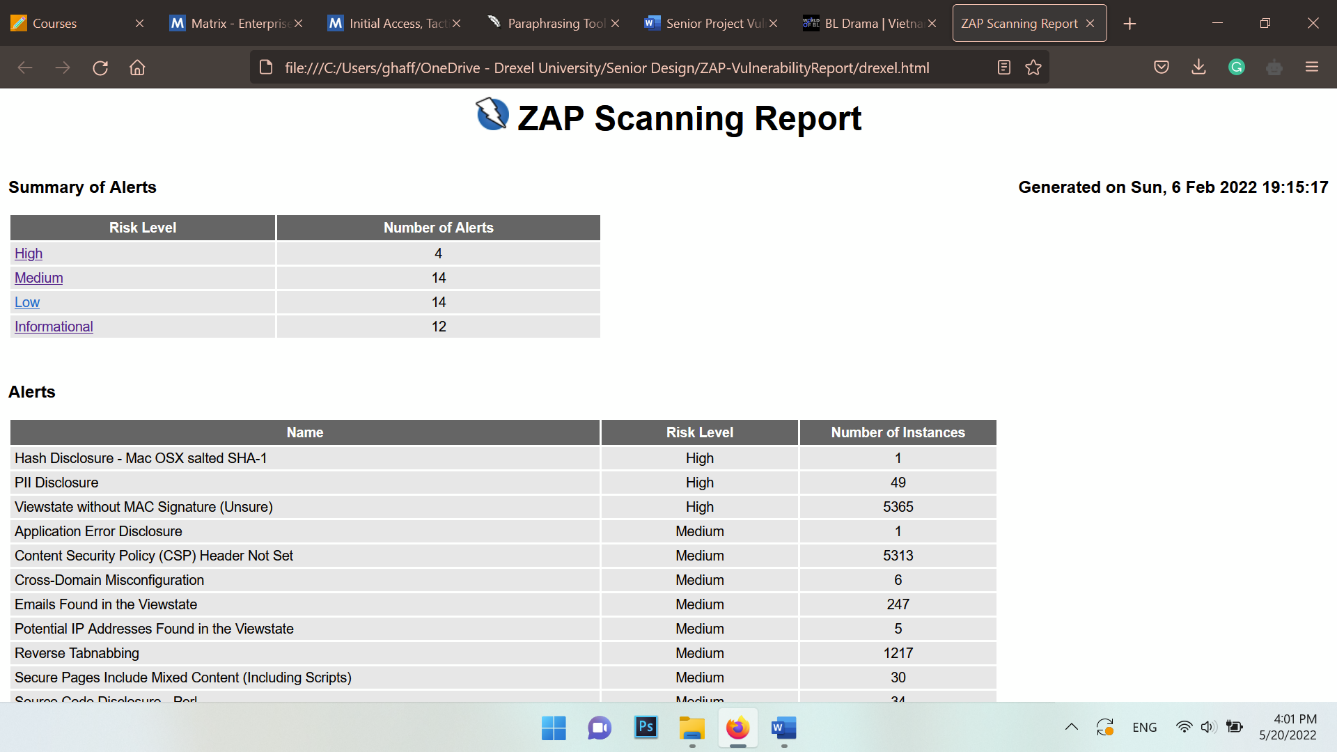
#### Nikto



#### Skipfish



#### ZAP (Zed Attack Proxy)



### Data Results

There were a lot of High, medium, and low Vulnerabilities after the scan. Three vulnerabilities from each have been discussed below:

#### High Vulnerabilities:

As previously stated, Drexel uses ASP hosting, which provides a very secure web. There were only six high severity vulnerabilities detected. Vulnerabilities that are high must be addressed right away. They're quite simple for attackers to exploit, and they might give them complete control over the systems they're attacking. The majority of these high vulnerabilities were discovered on the HR and employee benefits websites. Hash Disclosure, PII Disclosure, and View state without MAC Signature are some of the high vulnerabilities

* Hash Disclosure - Mac OSX salted SHA-1 was the vulnerability discovered on the Drexel website. This Vulnerability has the potential to be dangerous. Hash disclosure refers to the possibility that an encrypted version of a password is publicly available on your website. According to the scan report, this vulnerability was left during the architectural and design process. The CWE (Common Weakness Enumeration) id was also supplied in the report to refer to further information about the vulnerability.

Solution - Ensure that hashes that are used to protect credentials or other resources are not leaked by the web server or database. There is typically no requirement for password hashes to be accessible to the web browser

* PII Disclosure is another vulnerability discovered in the report as a result of failing to establish a security plan during the architecture and design phase. The result contains Personally Identifiable Information (PII), such as credit card numbers, a Social Security number, and other sensitive information. The majority of these findings came from the HR benefits and HR retiree sites. Another major PII vulnerability uncovered on the HR site was a credit card vulnerability, with the credit type being Master Card. The site, according to CWE, does not adequately protect a person's private, personal information from actors who are not explicitly permitted to access the information.

Solution – Ensure that the security plan has been established successfully.

* Viewstate without MAC Signature was another noticeable high vulnerability discovered on the website. This vulnerability was discovered on a number of different websites, including CCI, HR, Financial Aid, and Graduate Programs, to mention a few. If the ASP.NET settings do not ensure that the form authentication tickets are tamper-proof and encrypted, as well as that ViewState is tamper-proof, when the server processes the data, it will not notice any changes to the ViewState or authentication tickets made on the client's machine or over the network. There were 5365 instances of this vulnerability, according to the report, and the report's recommendation is to make sure the MAC is set for all pages on this website.

Solution - Ensure the MAC is set for all pages on this website

#### Medium Vulnerabilities:

There were 14 vulnerabilities of medium severity. These vulnerabilities typically provide attackers with information that might help them plan future attacks on the network. These should also be fixed as soon as possible, but they are not as critical as the other vulnerabilities. According to the report some of the medium vulnerabilities included information regarding Content Security Policy (CSP) Header Not Set, Reverse Tabnabbing, and Source Code Disclosure.

* Content Security Policy (CSP) Header Not Set was identified as the most critical medium vulnerability in the report. Content Protection Policy (CSP) is an additional layer of security that aids in the detection and mitigation of certain types of threats, such as Cross-Site Scripting (XSS) and data injection. These attacks are used for a variety of purposes, including data theft, site defacement, and malware delivery. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images, and embeddable objects such as Java applets, ActiveX, audio and video files. There were 5313 instances of this vulnerability in the scan report.

Solution - Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header, to achieve optimal browser support

* Reverse Tabnabbing was another medium vulnerability discovered during the scan. Reverse tabnabbing is a type of phishing assault in which an attacker convinces a victim to input their credentials on a false website which the attacker controls. This vulnerability was discovered at a variety of sites, including CCI, Lebow, Med, and Drexel Central, to mention a few. There were 1217 occurrences of Reverse Tabnabbing in the report, and at least one link on the page is vulnerable to Reverse Tabnabbing because it utilizes a target attribute in the "rel" attribute without both the "noopener" and "noreferrer" keywords, allowing the target website to take control of this page.

Solution - Do not use a target attribute, or if you have to then also add the attribute: rel="noopener noreferrer".

* Source Code Disclosure was another Medium Vulnerability discovered in the report. This Source Code Disclosure included Perl, PHP, Python, and SQL scripts. Source code disclosure makes sensitive application information like input validation filters, database connection strings and queries, and hard-coded passwords available to the public. An attacker with information of input validation filters might be able to create a request that will get around the filter. A number of instances of disclosure were recorded in the report. The most instances were found in SQL. It is suggested that this script be removed from the web server and moved to a place that is not accessible from the Internet.

Solution – Ensure that application Source Code is not available with alternative extensions, and ensure that source code is not present within other files or data deployed to the web server, or served by the web server

#### Low Vulnerabilities:

Low vulnerabilities have little influence on an organization's operations. Exploiting such vulnerabilities generally requires local or physical system access. Our Scan report also showed low vulnerabilities such as the absence of Anti-CSRF Tokens, Application Error Disclosure, Big Redirect Detected (Potential Sensitive Information Leak), and Cookie No HttpOnly Flag, to mention a few.

* Absence of Anti-CSRF Tokens was one of the low vulnerabilities discovered in the report. There were at least 5374 instances when no Anti-CSRF tokens were identified in an HTML submission form. A CSRF (cross-site request forgery) attack includes forcing a victim to submit an HTTP request to a target location without their knowledge or purpose in order to conduct an action as the victim. When the victim is logged in to the target site, CSRF attacks are effective. CSRF has generally been used to conduct an operation against a target site using the victim's privileges, but new approaches have been developed to leak information via getting access to the response.

Solution – Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

* Application Error Disclosure was another Low vulnerability found in the report. According to the report, the site displays an error/warning message that may reveal sensitive information such as the location of the file that caused the unhandled exception. This information can be used to conduct further attacks against the web application. If the problem message is discovered on a manual page, the warning might be a false positive.

Solution - Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.

* Another low vulnerability to be cautious of is "Big Redirect Detected" (Potential Sensitive Information Leak). According to the report, the server responded with a redirect that appeared to have delivered a huge response. This might imply that, despite sending a redirect, the server also responded with body content (which may include sensitive details, PII, etc.). This vulnerability was found in around 58 instances.

Solution - Ensure that no sensitive information is leaked via redirect responses. Redirect responses should have almost no content.

#### CWE

The team was able to make specific recommendations for CWE, which were also included in the report. The acronym CWE stands for Common Weakness Enumeration. The Common Weakness Enumeration is a classification method for weaknesses and vulnerabilities in hardware and software. The following is information from one of the High Vulnerability PII Disclosures discovered on CWE. CWE-359 was the vulnerability ID revealed by the scan.

