Vulnerability Scanning

[1]Prajakta Subhash Jagtap
[1]M.Tech Student
[1]K J Somaiya College of Engineering

Abstract - **Scientific advances of higher education institutions make them attractive targets for malicious cyberattacks. Modern scanners such as Nessus and Burp can pinpoint an organization's vulnerabilities for subsequent mitigation. However, the correction reports generated from the tools typically cause important info overload whereas failing to produce unjust solutions. Consequently, higher education institutions lack the appropriate knowledge to improve their cybersecurity posture. However, while not understanding vulnerabilities in a very system, it would be difficult to conduct successful network defence in order to prevent intruders in the real world. Therefore, vulnerability scanning is a key element to the success of cybersecurity curriculum. In this paper, we tend to review the state of the art of current open source vulnerability scanning tools. Literature survey is done on vulnerability, vulnerability scanning, vulnerability scanning tools, security vulnerabilities, system security and application security, malicious cyber-attacks shows that a lot of work is being carried out in vulnerability assessment and reporting. In this report gives exhaustive study on vulnerability scanning tools. We presented two main aspects in this paper vulnerability scanning and reporting. Then we identify the gaps in relevant practices and presenting selected results, we highlight future directions and conclude this research. We provide thorough descriptions on the top open source network vulnerability scanning tools. We then propose our hands-on labs research design in detail on network vulnerability scanning that we design specifically to enhance the cybersecurity curriculum.**

keywords - **Vulnerability, vulnerability assessment, Shodan, Nessus, Burp Suite, National Vulnerability Database**

Introduction

Security of information and communication systems has become one of the most crucial concerns for both system developers and users [1]. The threats to our computer network infrastructure are increasing and constantly changing in every day [2]. Hackers are launching more sophisticated attacks on every possible weakness in our computer network system and trying to damage or crush our security system. It is crucial that we train adequate cybersecurity professionals to defend our system and prevent cyberattacks. One of the main reasons of successful attacks, malicious intrusions and virus infections are software vulnerabilities in computer systems, communication equipment, smartphones and other intellectual devices. Most courses in cybersecurity education are concentrating on defensive techniques such as cryptography, intrusion detection, firewalls, and access control; or offensive techniques such as buffer overflow attacks, exploitation, and post-exploitation. Before conducting network defence, understanding what kind of vulnerabilities that exist in computer systems is the first and the most important step in protecting our security system. Therefore, understanding and teaching vulnerability scanning is a key element in cybersecurity curriculum. Vulnerability scanning as one of the initial steps in ethical hacking and network defense education. Higher education institutions have made remarkable advances in physics, medicine, and applied sciences [3]. However, these advances create a target for malicious cyberattacks. Higher education institutions depend on the confidentiality and integrity of sensitive data stored at their facilities (e.g., intellectual property, financial data). Institutions find security implementation difficult due to culture, staffing, and resources conflicting with the demand for robust security [4]. Technical vulnerabilities such as weakness in software or misconfigurations are key contributors toward risk in enterprise networks. Systems protecting data regularly contain exploitable flaws, many of which can be detected with popular vulnerability scanners (e.g., Nessus) [5].

Organization of the report

In this paper, we analyze and discuss vulnerability scanning tools hands-on our college network. The contributions of this paper are as follows:

- We explore the definitions and processes of vulnerability scanning.
- We provide descriptions of the top vulnerability scanning tools.
- Then I performed experiment of vulnerability scanning tools on our college network and observe the results of tools depending on some features, parameters.

The rest of the paper is organized as follows.

Literature Survey provides the background knowledge of vulnerability, vulnerability scanning including security vulnerabilities, system security and application security. I explore the top network vulnerability scanning tools, observe the results and also compared the tools based on some parameters. Finally concludes the paper with outcomes.

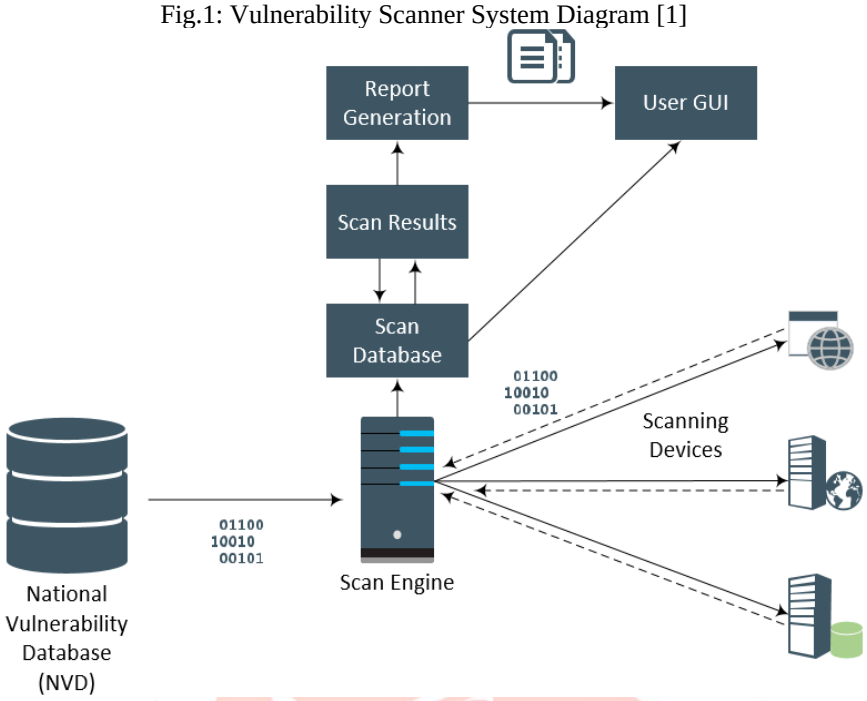
Literature Survey

Vulnerability Scanning

A vulnerability is "a flaw among a system, application, or service that permits AN aggressor to bypass security controls and manipulate systems in ways that the developer never intended" [1]. Vulnerability scanning is that the method of

mistreatment one laptop to seem for weakness in another laptop. It can also be used to determine vulnerabilities in a network [2]. Security specialists will use vulnerability scanning to seek out weakness in systems so as to mend and shield the systems. On the opposite hand, intruders may also use it to attack a system and hurt the system. Vulnerability scanners area unit package to scan the design of a network, report detected vulnerabilities and provide instructions on how to remediate them. The core design of a vulnerability scanner is represented in Figure 1. [1]

Fig.1: Vulnerability Scanner System Diagram [1]



Scanners often use definitions provided by the National Vulnerability Database (NVD) [1]. NVD contains information on Common Vulnerabilities and Exposures (CVE), which is a list of standardized names for known vulnerabilities [1].NVD have a risk level categorization known as the Common Vulnerability Scoring System (CVSS). Factors like attack vector, complexity, privileges needed, user interaction, and also the impact of confidentiality, integrity, and handiness area unit incorporated into CVSS scores [1]. Sample vulnerabilities in each are provided in Table 1.

Table I: Risk Levels And Examples For Vulnerabilities [1]

Risk Level	CVSS Ranges	Example Vulnerabilities
Critical	10.0	Remote Code Execution, Buffer Overflows, Default Credentials, Unsupported Operating System Versions
High	7.0 – 9.9	Malformed Packet Injection, Redirect Denial of Service, Privilege Escalation, Password Hash Disclosure
Medium	4.0 – 6.9	Remote Information Disclosure, Cryptographic Protocol, Command Injection, Web Directory Traversal & File Access
Low	0.1 – 3.9	Unencrypted Communications, Internal Information Disclosure, Browsable Web Directory
Informational	0.0	Software Version Disclosure, Protocol Detection, Operating System Identification, Device Type

Security Vulnerabilities

Vulnerability in computer security is a weakness or an unintended flaw in software code or a system that allows an intruder to exploit and reduces the system's information assurance.

Vulnerability typically consists of 3 elements: a system status or flaw, intruder's access to the flaw, and intruder's capability to exploit the flaw [2]. In order to use associate vulnerability, associate entrant must have a minimum of one applicable tool or technique to attach to a system having weaknesses.

Remediation & Reporting

Vulnerability remediation aims to remove vulnerabilities through a compensating security control. Security controls involving remediation can provide technical (e.g., applying a patch, security architecture changes, system or application hardening) strategies to address risk within an organization. The length of time a device remains unprotected on a network increases the potential for exploitation. Removal of the susceptible application, protocol, or device is ideal granted removal does not impede functionality [1]. Knowledge of needed devices, protocols, processes and applications in an

organization enhances a security professionals' ability to identify dangerous assets and apply appropriate defence. Removing a system deficiency usually affects many others, making a consequence.

Top Network Vulnerability Scanning Tools

Network scanning tools detect the networked devices and identify the services on those devices using fingerprint techniques. This chapter introduces two network scanning tools Nmap and Nessus, that are used in the experiment [9].

1) Nessus: Nessus provides vulnerability scanning for network devices, virtual hosts, operational systems, databases, internet applications and IPv4/IPv6 hybrid networks.

Nessus used to be an open source tool and can be found in Backtrack5 (BT5), but it is no longer free anymore.

2) Nmap: Nmap is popular due to its features of flexibility, capacity, portability, and simplicity. It is a flexible tool because a network filled with packet filters, firewalls, routers, and other obstacles can be mapped by Nmap. Nmap can be used to scan a network as large as having thousands computer hosts, and even as small as having a single host. It is portable because Nmap is supported by many popular operating systems including Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, and Sun OS. Nmap can be found in many systems, such as BT5, and Kali Linux [2].

3) OpenVAS: OpenVAS could be a framework of many tools and offers a comprehensive and powerful vulnerability scanning and vulnerability management answer. Its main part, the safety scanner, is accompanied with a daily updated feed of Network Vulnerability Tests and it's free for UNIX system, Windows, and alternative operative systems. OpenVAS is not the easiest scanner to install and use, but it is one of the most powerful security scanners that you can use for free. It can scan thousands of vulnerabilities and offers false positive management of scanning results [2].

4) Shodan: Shodan is use to discover which of your devices are connected to the Internet, where they are located and who is using them. Keep track of all the computers on your network that square measure directly accessible from the net. Shodan lets you understand your digital footprint. Shodan to perform empirical market intelligence. Shodan gives us greater visibility into the insecure, interconnected cyberphysical world in which we all now live.

Web application vulnerability scanners:

1) **Burp Suite:** Burp Suite may be a leading vary of cybersecurity tools, delivered to you by PortSwigger. We believe giving our users a competitive advantage through superior analysis. Burp Suite is AN integrated platform for playacting security testing of net applications [5]. Its various tools work seamlessly on to support the whole testing methodology, from initial mapping And analysis of an application's attack surface, through to finding and exploiting security vulnerabilities. Burp offers you full management, belongings you mix advanced manual techniques with progressive automation, to create your work quicker, more practical, and additional fun. *Burp Scanner* is a tool for automatically finding security vulnerabilities in web applications.

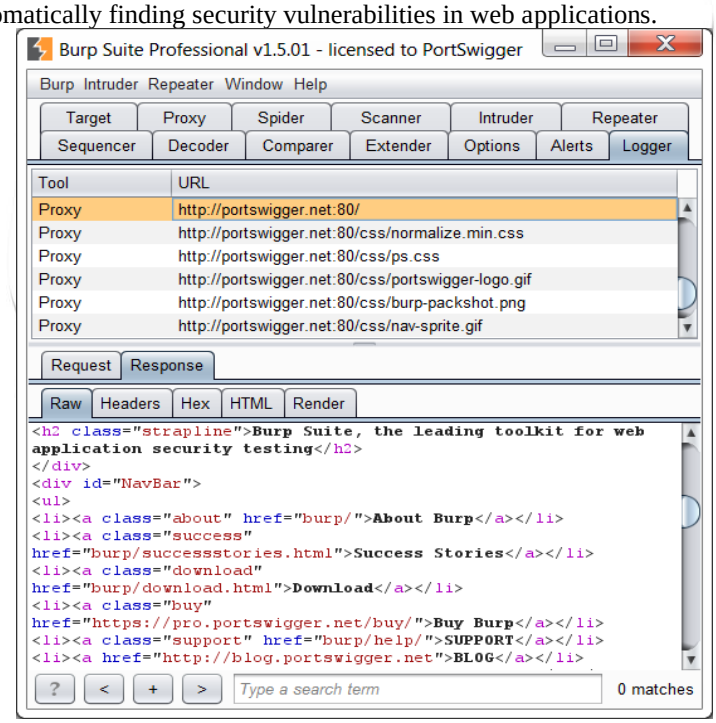


Fig.2: Burp Suite : Web Application Vulnerability Scanning Tool

2) W3af (Web Application Attack and Audit Framework): It is an open source web scanner that finds security vulnerabilities and aids in penetration testing efforts. It provides a vulnerability scanner and exploitation tool for Web applications [5]. W3af is written in Python language and is available for many popular operating systems such as Microsoft Windows, Linux, Mac OS X, FreeBSD, and Open BSD. W3af has mainly two parts, the core, and the plug-ins. It identifies most web application vulnerabilities using number of plug-ins.

3) Vega: It is a free and open source GUI based web scanner that is available for Windows, Linux and OS. Vega can find and validate SQL Injection, Cross-Site Scripting (XSS) and Remote File Inclusion. Vega includes an automated and fast scanner for quick tests and an integrating proxy for tactical inspection [5].

4) Acunetix : Acunetix square measures the pioneers in automated web application security testing with innovative technologies [5]. It is a multithreaded, lightning fast crawler and scanner that can crawl hundreds of thousands of pages without interruptions. It has highest detection of Word Press vulnerabilities, an easy to use Login Sequence Recorder that allows the automatic scanning of complex password protected areas. It has a built-in vulnerability management which can generate a wide variety of technical and compliance reports.

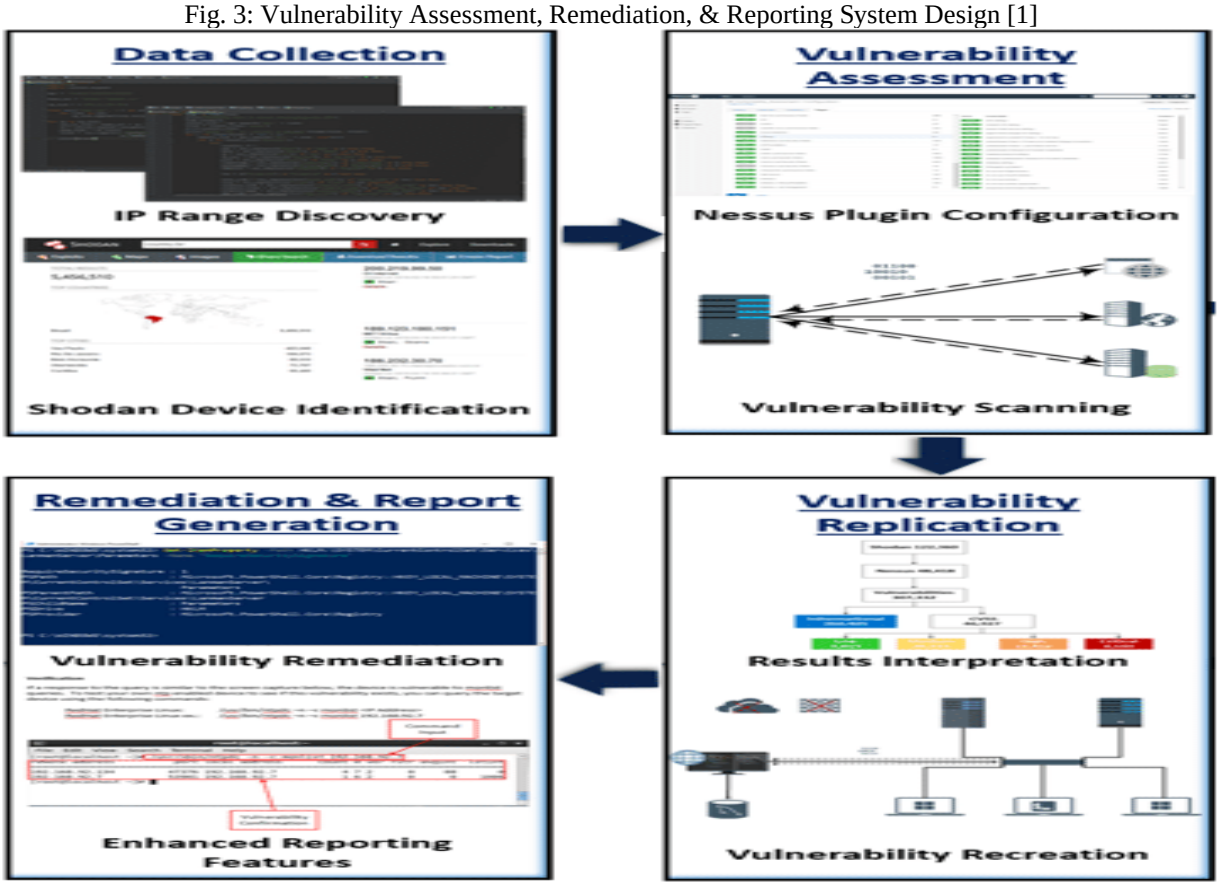
Vulnerability Scanning

1. System Design

We developed a system design (Figure 2) with four major components:

1. Data Collection
2. Vulnerability Assessment
3. Vulnerability Replication
4. Report Generation.

Fig. 3: Vulnerability Assessment, Remediation, & Reporting System Design [1]



2. Experiment with Nessus And Burp Suite Tools

I selected two tools for experiment Nessus and Burp Suite. I studied Nessus Professional and Burp Suite Professional which are available in our college.

I) Nessus Professional:

Nessus is a vulnerability scanner that lists the various vulnerabilities present in the remote host. There are different scans available in Nessus like Advanced Scan, Network Scan, Dynamic Scan, Mobile Scan, Malware Scan, Credentials Patch audit, Cloud Scan. Web application test can also perform in the scanner. Either the scanning can be done at the first instance provided or a template can be created first for a particular host and then it can be launched to run the scan against that host. Multiple scanning of the hosts can be done at once.

I selected Advanced Scan Policy, did configuration and gave the target 201.191.205.60/ to check the vulnerabilities. The vulnerability found by Nessus exists in five different types of severity- Critical, High, Medium, Low and Informal. Results are also saved as soon as the scan of a particular host is completed.

The results are provided in two different ways- vulnerabilities by plug-ins and vulnerabilities by host. Using the generated report, problems can be identified and fixed easily. The results can be exported in any desired format (e.g. PDF, HTML, CSS etc).

Fig. 4: Nessus Professional: Various Policies

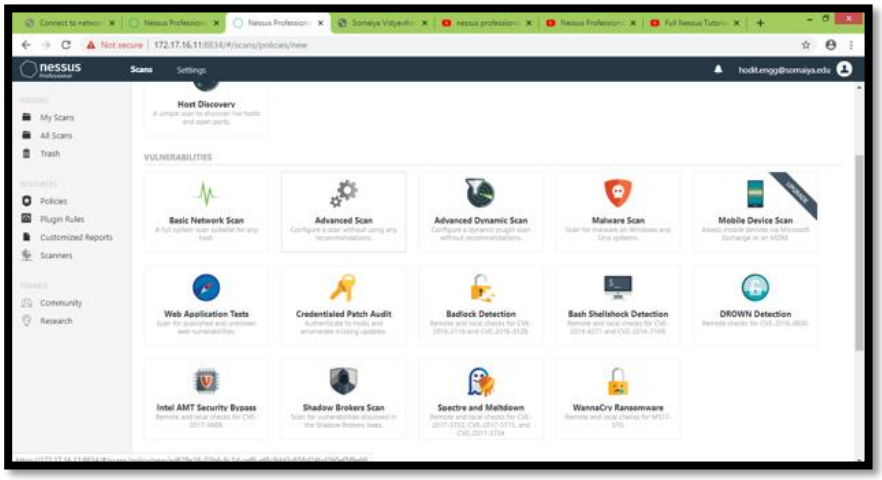


Fig. 5: Nessus Professional: Scanning Targeted Website

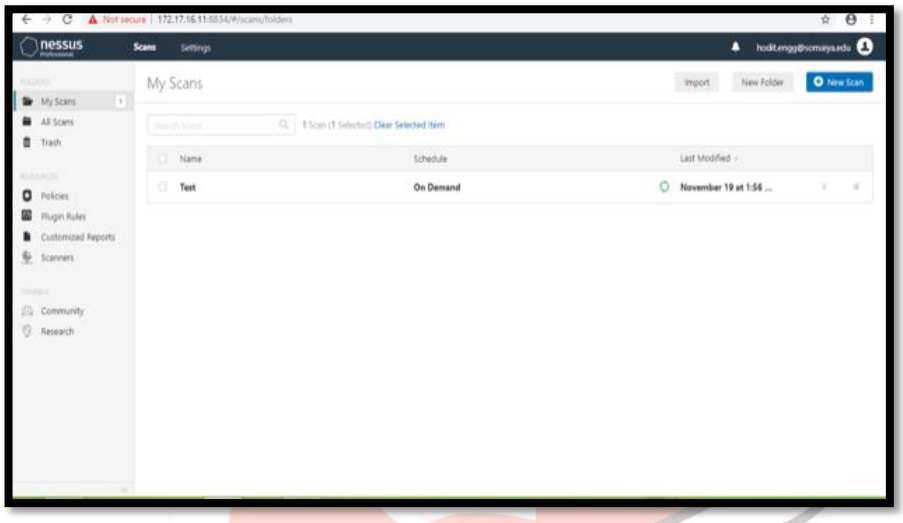


Fig.6: Nessus Professional: Scanning Results with various severity levels

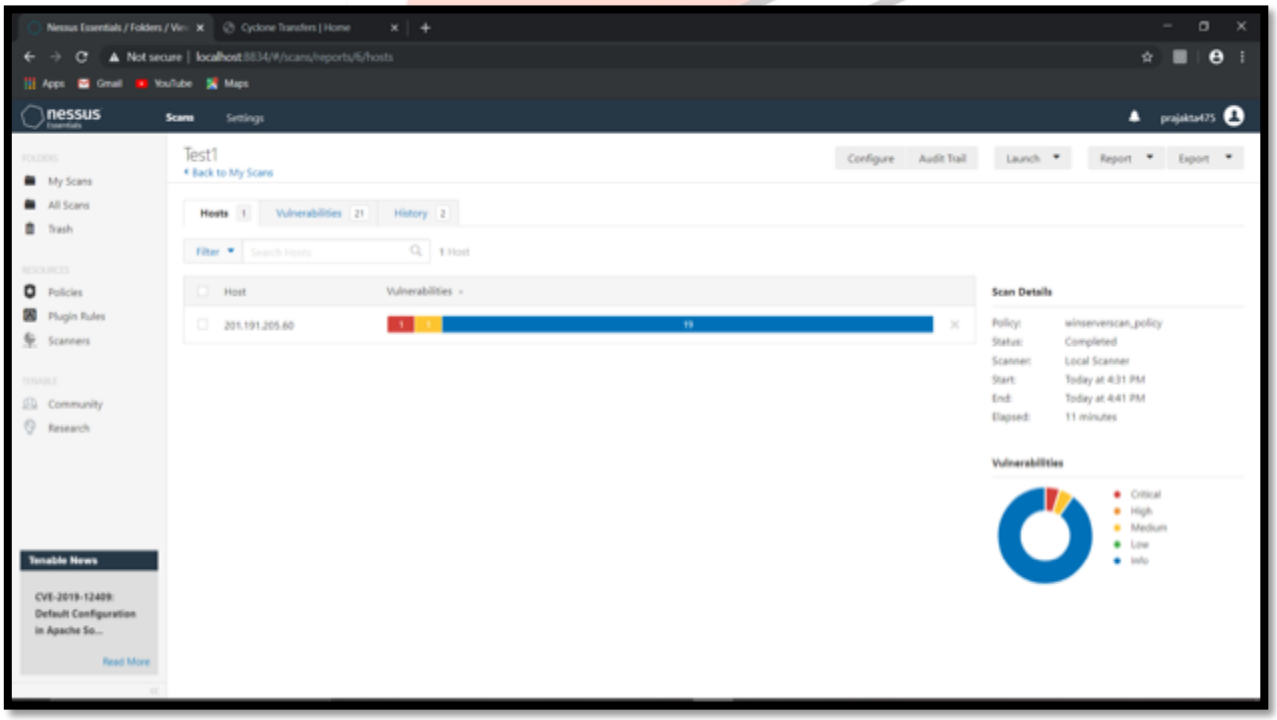


Fig. 7: Nessus Professional: Results showing Different vulnerabilities

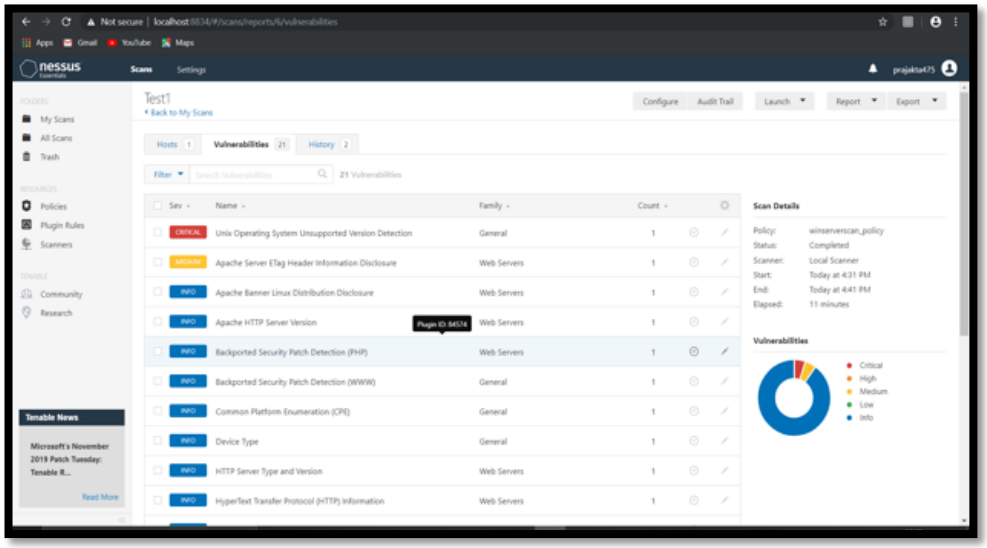
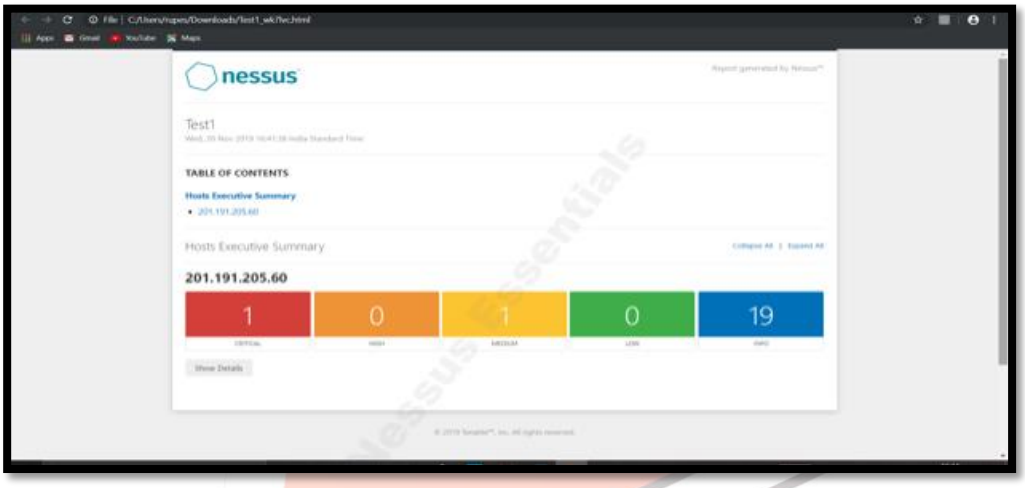


Fig. 8: Nessus Professional: Generated Report



II) Burp Suite Professional:

Burp is a proxy-based tool package. It consists of various functional specifications. To start working with Burp, it first requires setting the proxy in the browser whichever is being used as 127.0.0.1. After the proxy is set in the browser, Burp is ready to begin with. Burp window involves many tab specifications such as Target, Proxy, Intruder, Spider, Repeater, Sequencer, Scanner, Decoder, Actions, User options etc. where each tab has its own sub tabs. For instance, Target tab has two sub tabs-Site Map, Scope. I set the target to http://201.191.205.60/cyclone/ and perform testing using different tabs.

Fig. 9: Burp Suite Professional: Initial Proxy Setting

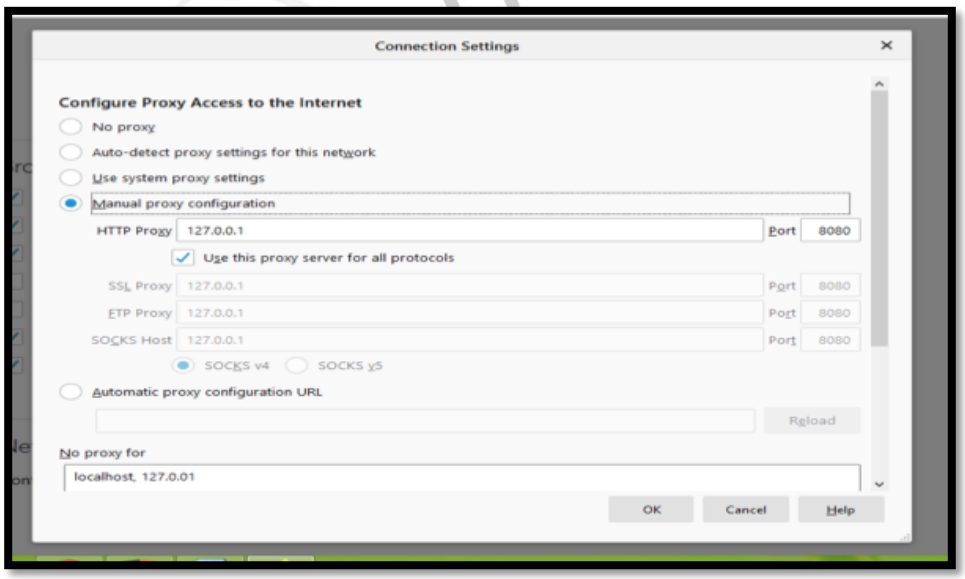
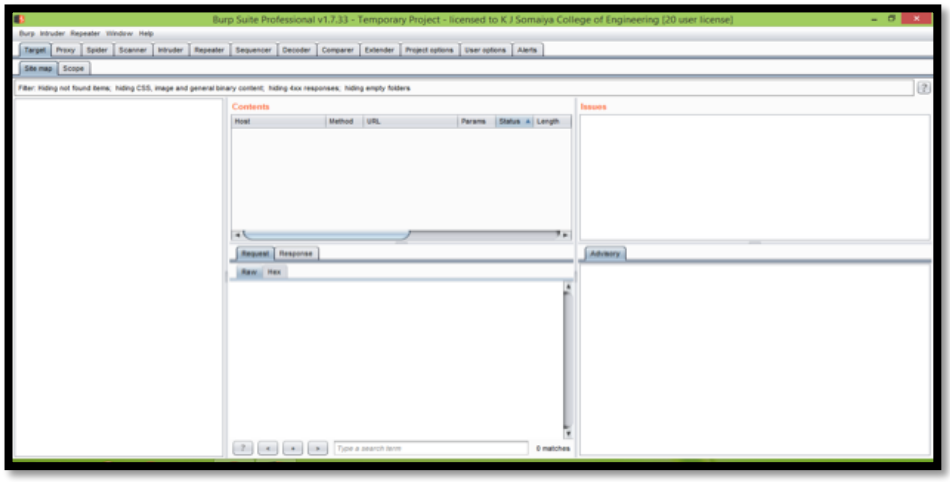
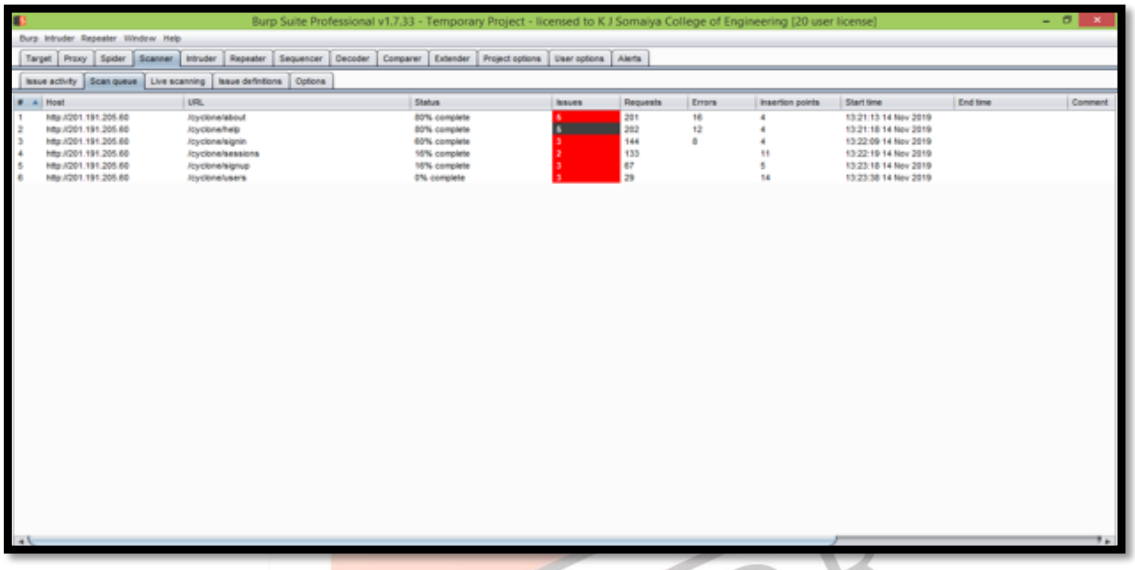


Fig. 10: Burp Suite Professional: Window with various tabs



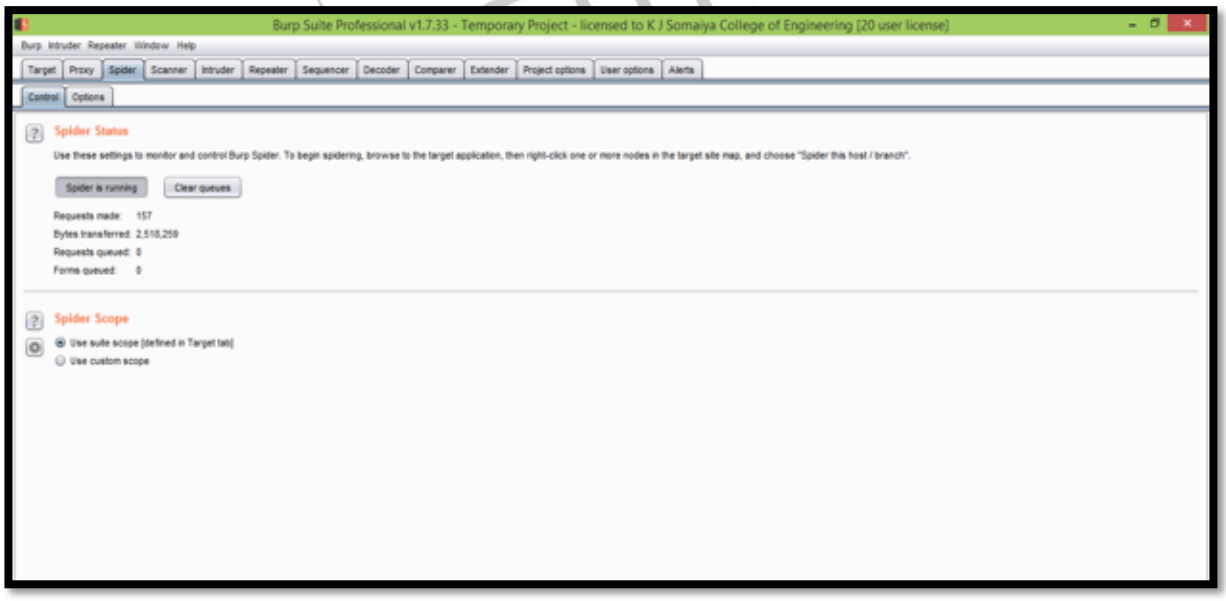
a) Scanner tab : It scans the target website and provide the scan queue which contains host, issues, requests, errors, start and end time.

Fig. 11: Burp Suite Professional: Scanner Tab Results



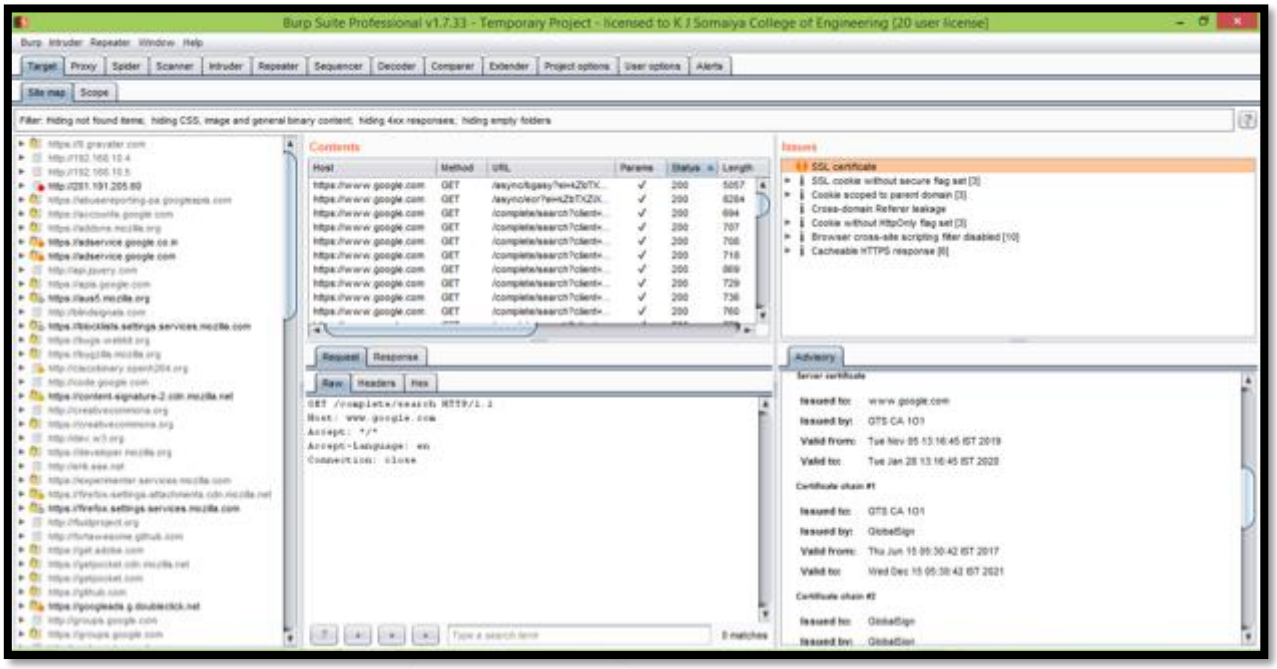
b) Spider Tab: Spider tab provides the crawling feature in the web application test.

Fig. 12: Burp Suite Professional: Spider Tab Result



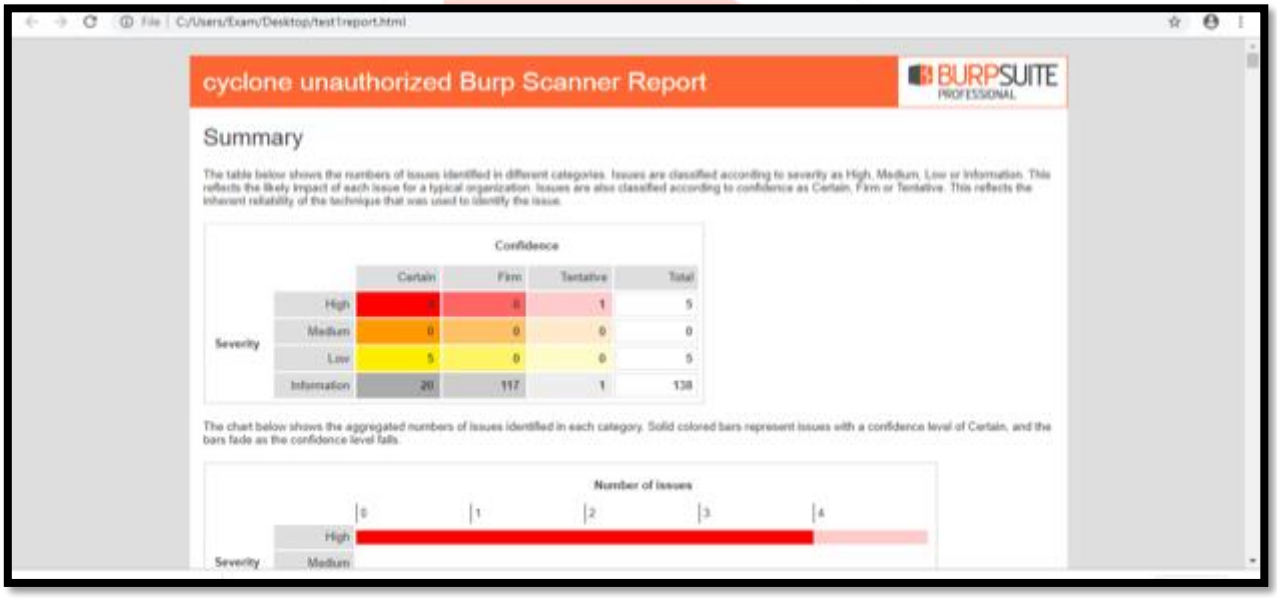
c) Target tab: It shows Contents, Site map, Issues. After clicking on specific issue it will display issue details in right bottom panel.

Fig. 13: Burp Suite Professional: Target Tab Result



Burp Suite performs the scanning of the hosts. Scanning involves testing the hosts for the vulnerabilities present in it. It identifies the type of vulnerability its severity and confidence.
The reports can be exported in any desired format like HTML, CSV. We can save the project also.

Fig. 14: Burp Suite Professional: Result of scanned website



Comparison of Nessus And Burp Suite

1) Comparison by Features:

Table II: Comparison of Nessus And Burp Suite

Features	Nessus	Burp Suite
1) Company	Tenable	Portswigger
2) Description	Large-scale vulnerability assessment tool with 80,000+ plug-ins designed to access various vulnerabilities	Integrated platform designed to identify vulnerability and verify attack vectors
3) Max no. of Hosts Available	Default – 30 Licensed – Unlimited	Scans multiple hosts via text file
4) Vulnerability Detection Lis	Systems, Networks, Applications, Malware, Control Systems, Mobile, Cloud, Devices etc.	114 vulnerabilities built in to scan such as SQL Injection, XSS, OS command injection, ASP.NET tracing enabled, File path traversal, etc. Also supports multiple plug-ins.
5) Testing	Only Automatic	Automatic and Manual
6) Severity	Critical, High, Medium, Low and Informal.	High, Medium, Low and Information.

2) Comparison of Results:

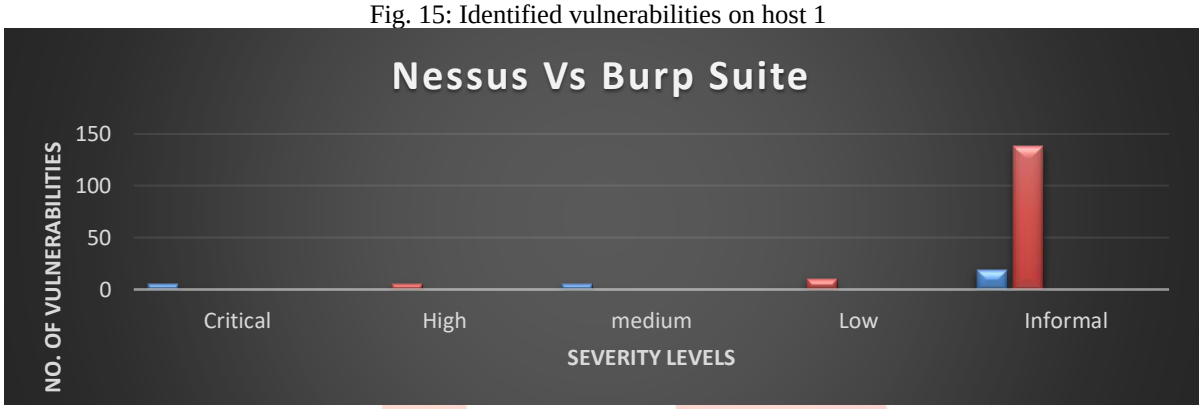
Table III : Number of vulnerabilities found in each range by Nessus.

Severity	Vulnerabilities Found	Example Vulnerabilities Found
Critical	1	Unix Operating System Unsupported Version Detection
High	0	None Found
Medium	1	Apache Server ETag Header Information Disclosure
Low	0	
Informal	19	Device type, CVE, OS identification, Service detection

Table IV: Burp Suite Vulnerabilities by Severity

Severity	Vulnerabilities Found	Example Vulnerabilities Found
High	5	SQL injection, flash cross domain policy,
Medium	0	None Found
Low	5	password field with autocomplete enabled
Information	138	Cross site request forgery

Fig. 15: Identified vulnerabilities on host 1



OWASP top 10 vulnerabilities and preventive measures

The Open Web Application Security Project (OWASP) is a non-profit organization dedicated to providing unbiased, practical information about application security. The OWASP Top 10 Web Application Security Risks was updated in 2017 to provide guidance to developers and security professionals on the most critical vulnerabilities that are commonly found in web applications, which are also easy to exploit [12]. These 10 web application vulnerabilities are dangerous because they may allow attackers to plant malware, steal data, sensitive information or completely take over your computers or web servers. The following table shows each of the OWASP Top 10 Web Application Security Risks and offers best practices to prevent or remediate them.

Table V: OWASP Top 10 Vulnerabilities and preventive measures

Sr. No.	Web Application Security Risks	Description	Preventive Measures
1	**Injection**	Injection flaws, such as SQL injection, CRLF injection and LDAP injection occur when an attacker sends untrusted data to an interpreter.	Application security testing can easily detect injection flaws. Developers should use parameterized queries when coding
2.	**Broken Authentication and Session Management**	Incorrectly configured user and session authentication could allow attackers to compromise passwords, keys, or session tokens, or take control of users' accounts to assume their identities.	**Multi-factor authentication**, such as FIDO or dedicated apps, reduces the risk of compromised accounts.
3.	**Sensitive Data Exposure**	Applications don't properly protect sensitive data such as financial data, usernames and passwords, could enable attackers to access such information to commit fraud.	**Encryption of data at rest and in transit** can help you comply with data protection regulations.
4.	**XML External Entity**	Poorly configured XML processors evaluate external entity references within XML documents. Attackers can use external entities for attacks	**Static application security testing (SAST)** can discover this issue by inspecting dependencies and configuration.
5.	**Broken Access Control**	Improperly configured or missing restrictions on authenticated users allow them to access unauthorized functionality or data, such as	**Penetration testing** is essential for detecting non-functional access controls; other testing methods only detect where access controls are missing.

		accessing other users' accounts, viewing sensitive documents	
6.	**Security Misconfiguration**	This risk refers to improper implementation of controls intended to keep application data safe, such as misconfiguration of security headers, error messages and not patching or upgrading systems, frameworks.	**Dynamic application security testing (DAST)** can detect misconfigurations, such as leaky APIs.
7.	**Cross-Site Scripting**	Cross-site scripting (XSS) flaws give attackers the capability to inject client-side scripts into the application, for example, to redirect users to malicious websites.	**Developer training complements security testing** to help programmers prevent cross-site scripting with best coding best practices, such as encoding data and input validation**.**
8.	**Insecure deserialization**	Insecure deserialization flaws can enable an attacker to execute code in the application remotely, tamper or delete serialized (written to disk) objects, conduct injection attacks.	**Application security tools can detect deserialization flaws** but penetration testing is frequently needed to validate the problem
9.	**Using Components With Known Vulnerabilities**	Developers frequently don't know which open source and third-party components are in their applications, making it difficult to update components when new vulnerabilities are discovered.	**Software composition analysis** conducted at the same time as static analysis can identify insecure versions of components.
10.	**Insufficient Logging and Monitoring**	Insufficient logging and ineffective integration with security incident response systems allow attackers to pivot to other systems.	**Think like an attacker** and use pen testing to find out if you have sufficient monitoring; examine your logs after pen testing.

CONCLUSION

Researchers and educator's area unit creating outstanding progress in pushing the boundaries of information at intervals educational activity. However, these advances often make higher education institutions the target of malicious cyberattacks. While many vulnerability assessment tools can aid institutions identify the significant information overload and lack of actionable fixes provided by reports generated by these tools prevents the efficient and effective remediation of detected vulnerabilities. In this study, we leveraged Nessus, a state-of-the-art vulnerability assessment tool and burp suite to identify numerous vulnerabilities in education institutions. In this paper, I have presented a performance comparative study between two most used vulnerability scanning tools: Nessus and Burp Suite. The comparison is based on three main features: The ability to search, Scanning Time, the ability to detect vulnerabilities. Both scanners performed very well in vulnerability identification. In terms of speed with Web Application feature, Nessus performed much faster than Burp Suite. In this paper I have implemented the licensed version. The future work will be focused on performance evaluation based on other features, aside from ability to search, Scanning Time, and the ability to detect vulnerabilities. Also, we intend to evaluate other vulnerability scanning tools, not only software based but also hardware based. The vulnerability assessment was helpful as it provided information about the security of the selected websites. There are several promising directions for future research. First, work is drained collaboration with elect and interested educational activity establishments to help in correction efforts. Second, the enhanced reports can be created for additional vulnerabilities and posted on a vulnerability-wiki to assist in unifying remediation efforts. Finally, a usability study could be conducted to further identify faults in current reporting mechanisms to quantitatively measure the time to remediate a vulnerability compared to the current reporting mechanisms. In future research, we seek to outline preventive maintenance scheduled and well-known techniques to secure websites owner and also educate to higher education institutions. Each direction can enhance an organization's ability to remediate detected vulnerabilities and help ultimately ensure a safer, more secure society.

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my guide "Dr. Manish Potey" sir for their able guidance and support in completing my project. I would also like to extend my gratitude to the Principal mam "Dr. Shubha Pandit" and HOD of Computer Department Sir "Dr. Deepak Sharma" for providing me with all the facility that was required.

REFERENCES

[1] Harrell, Christopher R., et al. "Vulnerability Assessment, Remediation, and Automated Reporting: Case Studies of Higher Education Institutions." *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2018.

[2] Wang, Yien, and Jianhua Yang. "Ethical hacking and network defense: Choose your best network vulnerability scanning tool." *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, 2017.

[3] Holm, Hannes, and Teodor Sommestad. "Sved: Scanning, vulnerabilities, exploits and detection." *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016.

[4] Appiah, Vincent, et al. "Survey of Websites and Web Application Security Threats Using Vulnerability Assessment." (2018).

[5] Aarya, P. S., et al. "Web Scanning: Existing Techniques and Future." *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2018.
[6] Gorbenko, Anatoliy, et al. "Experience report: Study of vulnerabilities of enterprise operating systems." *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2017.
[7] Yadav, Ravinder, and Aakash Goyal. "Web Application Security." *International Journal of Computer Science and Mobile romnlltlno-Vol1 Tccru* 10 (2014): 349-355.
[8] Kushe, R. "Comparative Study of Vulnerability Scanning Tools: Nessus Vs Retina." *Security & Future* 1.2 (2017): 69-71.
[9] Im, Sun-young, et al. "Performance evaluation of network scanning tools with operation of firewall." *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2016.
[10] Stephenson, P. "Tenable Network Security Nessus." (2015).
[11] Bairwa, Sheetal, Bhawna Mewara, and Jyoti Gajrani. "Vulnerability Scanners-A Proactive Approach To Assess Web Application Security." *arXiv preprint arXiv:1403.6955* (2014).