Vulnerability Assessment and Penetration Testing through Artificial Intelligence

Prof. Shraddha More¹, Arpit Rohela²
¹,²Department of Information Technology, SJCEM,

Abstract— Every organization has some serious assets to be taken care of in aspects of security. For this purpose they employ professionals, these professionals cannot take care of a problem if they don’t have total insights of the assets of the organization in the proper format, internally and externally i.e., relation between all the assets, their states and configurations. Later comes the penetration testing part where all of these assets test for their hardening level. Today’s solutions lacks the proper relation between representation, gathered data and make sense on its own moreover take action on that and total framework where all of the obtained information and test plans with test results can be brought to us from the single platform and compared for changes to know whether they are effectively enough, or are just posing an overhead for effective business and technology environment. The security auditors give away a lot of time and effort for collection of data/evidence and making information out of it whether by scans or open source information collection. This framework is designed to ease the penetration testing and security hardening jobs. It uses all the updated technologies like AI, cloud and big data. The AI is for making relation between the collected data, relate the vulnerability in its root configuration or versions of the assets and suggest or execute the exploit in order to test the hardening of the assets. The cloud will pose as the platform which will stabilize the major extent of processing, hold information/evidence and data apart from big data for managing AI training dataset.

Keywords—Vulnerability Assessment; Penetration Testing; Artificial Intelligence; CVE; Two-factor authentication; man-in-the-middle attack; Cross-Site Request Forgery (CSRF); Un-validated Redirects.

I. INTRODUCTION
The vulnerability assessment and penetration testing are as old as security in the computer world. Vulnerability testing promises vulnerability discovery based on the known symptoms while penetration testing is practically putting weakness to test to reveal real hardness of the security in place. While vulnerability analysis focus more on whether the weakness exists or not, penetration testing is to determine if unauthorized access to key assets are a possibility [1]. Answer to most of the vulnerability scanners scan results can be measured in percentage of correctness and are burdened with probability of false positive in numerous cases whereas penetration testing is deviation less which means the solution of the process is definite i.e., Yes( exploitable) or No(Not Exploitable).

The importance of this framework with respect to vulnerability assessment and penetration testing is to implement automatic defense systems that can monitor, discover and prove what it discovered is right, with exploit generation, and correct software flaws in real-time, effectively using AI [2]. The framework bring different sorts of AI paradigm and dynamic programming in use. One of the AI paradigms is Reinforcement Learning under which it solves control problems and sequential decision making tasks, second one is Supervised Learning which is used to solve pattern recognition, regression (function approximation), further comes Unsupervised Learning which solves estimation problem, grouping (clustering), estimation of statistical distribution, compressing and filtering [3].
Some of the past work done is from Darktrace, which promises anomaly detection driven by unsupervised machine learning [4] and some participated machines from 2016 Cyber_Grand_Challenge form DARPA. Those systems brought in use numerous tools, techniques and expert knowledge to create fully autonomous self-driven systems that perform automated software patching, vulnerability detection and exploit generation in binary software without human intervention [5].

![Fig.1: Intrusion detection of network with the help of AI](image)

The rest of the paper is organized as follows: Section II describes related work. In Section III discuss about working of proposed system. Section IV presents implementation details about the proposed system. Section V projects on conclusions and future scope.

### II. LITERATURE SURVEY

The Patternex is a system that essentially focus on Humans and computers to work together to identify evolving cyber-attack patterns buried in our data [6], which basically is allowing analyst to help machine learn their techniques to detect and act on cyber-attack. Vectra can differentiate among anomalous user and a potential hacker threat behaviors. It also automatizes attacker detection and allows to respond faster to the most serious threats [7] while Patternex works externally, Vectra is used after the attacker is in the protected parameter after successful breach of the network. While Patternex involves humans accuracy which can be considered best when taking decisions on whether or not to act on the detection of event or sequence of events, there exists another detection approach called Darktrace which uses algorithms and unsupervised learning, which is pretty successful [2], how it basically works is comparison of the already existing system infrastructure with anything unusual or unseen detected.

The above systems are related to network intrusions but nowhere related to penetration testing of the software that contribute to network traffic. The system that can do this penetration testing and can contribute to improvements are systems are the one that participated in DARPA’s Cyber Grand Challenge like Mahem.
III. PROPOSED WORK

The Patternex is a system that essentially focuses on Humans and computers to work together to identify evolving cyber-attack patterns buried in our data[6], which basically is allowing analysts to help machine learn their techniques to detect and act on cyber-attack. Vectra can differentiate ability scanner will first scan the webserver, retrieve the results by eliminating the false positives. The second stage will intake the results from vulnerability scanner and perform the operations which are generally performed by penetration testers to make a system secure with the help of Artificial Intelligence (AI) technology. The system here eliminates the need of penetration tester’s repetitive work or complete need of penetration tester’s involvement in the security analysis process. The network vulnerability scanner on the cloud is scalable and hence performs fast port scans on the server IP address, finds the open ports of the server, reveals the service and service versions of the open ports. These services are searched in the CVE database and known vulnerability are tested by the AI algorithms, the results from the results are shown in the reports. The web app scanner will perform test modules from the stack of attack provided by us on the vulnerability scanner, then will move on to other attacks that will be performed by the AI tech applied on the cloud by gathering all the information from the web app vulnerability scanner and there on. At the last the vulnerabilities that are high on priorities are sent as email immediately with overall report that list other medium and low information leaking vulnerabilities.

Fig 2: System Architecture

The first step is to check if the host to be scanned is up and running in order to avoid wasting time on scanning a dead or unreachable host. This detection is done by probing some well-known TCP and UDP ports. If the scanner receives at least one reply from the remote host, it continues the scan. The second test is to check if the host is behind any firewalling/filtering device. This test enables the scanner to gather more information about the network infrastructure and will help during the scan of TCP and UDP ports. The third step is to detect all open TCP and UDP ports to determine which services are running on this host.
Once the TCP port scanning has been performed, the scanner tries to identify the operating system running on the host. This detection is based on sending specific TCP packets to open and closed ports. Once TCP/UDP ports have been found open, the scanner tries to identify which service runs on each open port by using active discovery tests. Once the scanner has identified the specific services running on each open TCP and UDP port, it performs the actual vulnerability assessment. The scanner first tries to check the version of the service in order to detect only vulnerabilities applicable to this specific service version. Every vulnerability detection is non-intrusive, meaning that the scanner never exploits a vulnerability if it could negatively affect the host in any way.

IV. EXPERIMENTAL RESULTS

The system implementation is divided into following phases:

4.1 Cross-Site Request Forgery (CSRF)

A CSRF attack is shown in fig.2 which forces a logged-on victim’s browser to send a forged HTTP request, including the victim’s session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim’s browser to generate requests the vulnerable.

![Fig 3: CSRF Admin page](image)

4.2 Un-validated Redirects

Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

![Fig 5: Redirect Python Run](image)
4.3 Using Components With known Redirects
Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

4.4 Mail Login
Passwords for the mail id are guessed, applied and validated from a dictionary that consist of list of most commonly passwords. As the password is found the attacker gets notified with the correct password and can use it further exploit the user.
4.5 Sensitive Data Exposure
Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

4.6 Security Misconfiguration
Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.
4. 7 SQL Injection
Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an
interpreter as part of a command or query. The attacker’s hostile data can trick the interpreter into
executing unintended commands or accessing data without proper authorization.

4.8 Broken Authentication and Session Management
Application functions related to authentication and session management are often not implemented
correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other
implementation flaws to assume other users’ identities.

V. CONCLUSION AND FUTURE WORK
In this system, we presented a Vulnerability scanner that aimed at detecting web application
vulnerabilities. The simplicity and friendliness is the advantage of the project. The environment is
made user friendly to the maximum so that anyone can run the scans provided and could access the
system via the secure authentication system via the time based one-time passwords. The
vulnerabilities for which the system will be scanned for are the most common major threats to web
application which comes under OWASP top 10 vulnerabilities. The scanner need not to be installed
on any system for use which is a bigger advantage and will be accessible through an internet based
service hence no performance load will be an impact on the users system.
The scanner was built by keeping in mind the ease with which the user can access the service without having any prior knowledge continuing with it in the future the results obtained by scanning will directly be mailed to the users registered email address. Patches for the detected vulnerabilities will be suggested to the user in the report itself so that the time consumed in repairing the system will be considerably decreased.

REFERENCES

IV. “DARK TRACE”: https://www.darktrace.com, Accessed on 28.08.2016, 05.00 PM.