RESEARCH ARTICLE OPEN ACCESS

Penetration Testing An Effective And Versatile Tool For Software Security

*Sachin Chaudhury[*1], Mansee Aggarwal[1] Sunayana Chaudhury[2]
[1]*University School of Information & Communication Technology, Guru Gobind Singh Indraprastha University, Sector 16C, Dwarka, Delhi-110078, INDIA*
[2]*QA InfoTech Private Limited, A-8 Sector , Noida, Uttar Pradesh, India*
[*]*Corresponding Author: Sachin Chaudhury*

ABSTRACT: Software is being developed and released for public utility in applications of mobile phone technologies to aircraft and spaceship explorations at a very fast pace which requires proper testing prior to their release so as to make them full proof and free of malware attacks. Artificial Neural Network (ANN) has a vital role in Penetration Testing. Artificial Intelligence (AI) algorithms learn from test cases to provide sagacious insights like application stability, failure patterns, defect hotspots, failure prediction. These insights further help to anticipate, automate, and enhance decision-making capabilities. ANN makes Penetration Testing far better and user friendly for a tester by elimination of hundreds and thousands of redundant, vague, undesirable teat cases, thereby, reducing project costs. It is envisaged that in future, development of intense algorithms, AI will make Penetration Testing even better by generating new and relevant test cases for a given test system.

Keywords: Artificial Intelligence, Artificial Neural Network, Penetration Testing, Software Testing

---

Date of Submission: 24-12-2017 Date of acceptance: 14-01-2018

I. INTRODUCTION

Making a system secure is everyone's number one priority. A safe and secure system leads to a better welfare of the society. But prior to that we need to understand how we can secure our system from the anti-social elements who are striving for gaining access to your personal data files and information for misappropriations. Recently, in May 2017, for internet users a group of individuals hacked the systems globally by introducing "Ransomware" a malicious software through a technique of "Crypto Extortion" by blocking access to the individual systems and demanded large sum of money as a ransom in order to release the system from their captivity and for smooth operations. The "WannaCrypt" attack infected over more than 75,000 individuals in more than ninety nine countries around the world, especially, in Europe https://en.m.wikipedia.org. Cyber security is most prominent and should be monitored regularly. One such testing that helps in securing the system is *Penetration Testing*. There are various tools that make Penetration Testing plausible. Testing makes a system secure and assures the user about their privacy and cyber safety. Moreover with the growing technology, effects of Penetration Testing could be improved by *Artificial Neural Networks* (ANN) allowing the pen testers to focus on more perilous threats. In this high tech world of machinery and robotics there is millions of sophisticated software's running continuously day and night. The ways software are developed and released now are way different if compared twenty years ago. Recent software's are much more fulfilling and secured, and it's all due to a new phase in software development cycle and that is *software testing*. Software testing has made devices, products and general programs somewhat immune. Back in the days, companies generally released its product without actually testing it firmly which led to several software crisis and loss of millions of dollars. Owing to fast development of IT industry and search for improvised software testing tools, products are now much more reliable. But with the burgeoning of technology, several threats to harm the systems have also augmented. *Black Hat Hackers* have indulged in several methods to access crucial personal information that users give while using internet applications and software. Therefore it's the company's job to make a robust system so that it is totally safe and

sound. Penetration Testing is one such testing technique that ensures total system's security. Penetration testing is a kind of security testing which is used to test the insecurity of an application. It is performed to find and detect the security risk/malware which might be present in the system as reported by Northcutt *et al.* 2017 [1]. To make things even better, the most prominent gift of the Twenty First Century, *Artificial Intelligence* can take Penetration Testing to a whole new level, thereby, leaving a minimal scope to the hackers to attack the system. In the present paper an attempt has been made to highlight the importance and relevance of Penetration Testing.

II. PENETRATION TESTING

Penetration Testing can be defined as "**A series of activities undertaken to identify and exploit security vulnerabilities"** [1]. The most general idea in a Penetration Testing is to simulate you as a hacker and take down all the possibilities of how a hacker could possibly attack a system. As there could be millions of possibilities to attack a system, tester couldn't prove the system to be 100% secure although testers can make the system vulnerable. Let's take an example to get a better understanding of why Penetration Testing is important. A series of unpatched vulnerabilities in the civilian government agency's Web server were employed for tracking usernames and passwords which were then reused on a host of enterprise systems. Hence they got Windows domain administrator privileges thereby; gaining complete access to a majority of Windows based system in the enterprise, including workstations used by Department of police. In this way through remotely controlling the entire system that automatically gave complete access to the workstations of the FBI's crime database. Therefore, this susceptibility of hacking the system by third party could have been easily avoided by way of definitely separating the police network and the enterprise network domains has been reported [2]. http://www.cica.ca/index.cfm/ci_id/15758/la_id/1.htmlTherefore, it is understandable that Penetration Testing is the most versatile software testing tool in IT industry.

2.1Phases of Penetration Testing

Penetration Testing should not be construed as merely a dull serial execution of automated tools and bundles of technical operations. It should provide a concise and definite solution for securing an organization's information and database systems from potential attacks of hackers. One intriguing factor for the Penetration Testing success lies in the methodology adopted. Therefore, a systematic and guided move toward should be maintained for successfully documenting a test and generating quality reports for onward transmission to different levels of management within an organization has been reported [3] http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9 087441", 2017-04-25. There are three major phases in a Penetration Testing that can be broadly grouped as: **1) Test preparation phase, 2) Testing phase** and **3) Test analysis phase.**For **Test Preparation Phase** all the important and required documents for tests are assembled and finalized. The testers and the company discuss and chalk out the objectives, duration, scope and timing for conducting the test. Various problems pertaining to information leaks and downtime are resolved and recorded as a legal document. Various other legal agreements which are essential for both the tester and the company are also included and signed during this phase [3]. The majority of the **Penetration Testing Process** is performed during the test phase. A collection of different automated tools are employed in this phase. This phase essentially has the following three major steps: Information gathering, Vulnerability analysis, and Vulnerability exploits. In information gathering step the tester is required to gather bodily and also the rational areas of a test target and categorize all the relevant information needed in the vulnerability analysis phase. Based upon the in sequence collected or in some case provided by the organization, then the penetration tester tries to analyze the vulnerabilities which may exist within the target's network, host and application. The tester may choose to employ manual testing but various other automated tools are available for performing the test. In the final step, it allows the tester to find exploits for the vulnerabilities eventually. In case such exploits do not lead to

what is intended, then there is a creation of loop between vulnerability analysis and vulnerability exploit phases [3]. The results of the test are properly examined during the **Test Analysis Phase**. Finally, the results of the test are provided to the company ensuring that it must be comprehensive and systematically documented. Clear cut proposal of a mitigation plan is imperative in Penetration Testing has been reported by Bacudio *et al.* 2011 [4]. Hence, it is absolutely, compulsory to include a mitigation plan section in the test analysis report.

2.3. Penetration Testing Tools

To secure a system, there are several automated and manual testing tools. Each of the tools has its own significance. For any system, it is not necessary that only one tool is used for testing. But dozens of tools can be used for different purpose according to threats generated in order to carefully secure the system from the hackers. A few major testing tools are: **1) Nmap, 2) WireShark and 3) Metasploit** which are explained below.

2.3.1. Nmap

In this, the scanner is competent of craft packets and performs scans to a modicum Transmission control protocol (TCP) level, e.g. ACK scan, SYN scan. It possesses a default signature-checking algorithm that guesses Operating system as well as the version, based on network responses such as a TCP handshake. Nmap is competent enough in finding remote devices, appropriately detect firewalls, routers, as well as make and model. Network administrators is empowered to apply Nmap in detecting which are open ports, and ascertain whether these ports could be exploited later on in simulated attack by the hackers. The output is ordinarily in a plain text and pleonasm; hence, this tool can be easily scripted to mechanize custom odd jobs. This is perhaps one of the only tools to remain eminent for almost a decade has been reported by Saindane, 2017 [5].

2.3.2. Wireshark

The most initial step in vulnerability assessment is to have an intelligible depiction of what is occurrence on the network. Wireshark operates in promiscuous mode for capturing all traffic of a TCP broadcast domain. Tailor made filters can be designed to detect and terminate specific traffic; e.g., capturing communiqué between two IP addresses. Thus the traffic data can be placed in a capture file, which could be observed later on. Various other filters can also be set throughout the review process. Characteristically, the tester is looking for the random IP addresses, spoofed packets, freak packet drops, and apprehensive packet generation from a single IP address. Wireshark provides a broad and a definite portrait of what is occurring on the network [5].

2.3.3. Metasploit

Once the above sniff and scan has been accomplished with the help of previous tools, it is the correct point in time to move forward to the OS and application stage. Metasploit is a remarkable, dominant open source framework which performs meticulous scans in opposition to a set of different IP addresses. Unlike other frameworks, it can also be used for anti-forensics. Professional programmers will develop a portion of code exploiting a particular vulnerability, by employing Metasploit for its detection [5]. The progression can be technically reversed, e.g., when a dangerous malware attacks by way of some concealed vulnerability, Metasploit becomes very handy for patching it up.

3.1. Artificial Neural Network

The main objective of a software tester is for reducing number of test cases for a given software in such a way that, the system stays safe and secure as well as the cost should always remain quite economical. To attain this phenomenon, *Artificial Intelligence* or particularly, *Artificial Neural Network* is used. An ANN comprises of a number of processing units called as neurons which are associated to each other through weighted links has been reported by Phatak, 2017 [6]. The neurons are formed in a number of multiple layers. It includes an input layer, followed by a hidden layer, and finally an output/target layer as shown in Fig. 1.

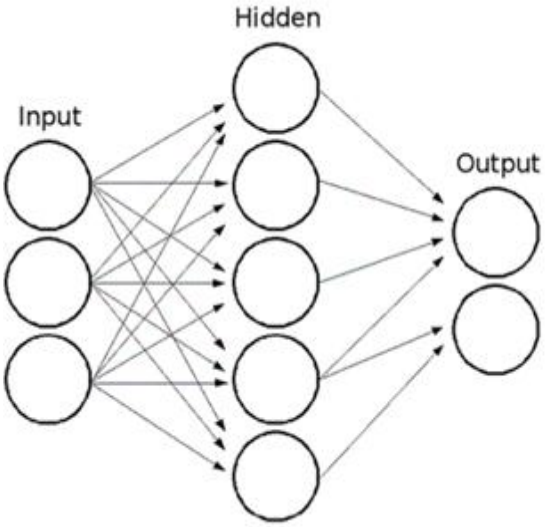

Fig 1: Simple Neural Network (Phatak, 2017) .

The Input layer receives an exterior activation vector which is passed on via weighted connections to neurons in the first hidden layer. The neurons in hidden layers compute a weighted sum of their activation inputs which a sigmoidal function is typically and overtake the result to neurons in following layer. Function signifies entire network, and traces the input vector onto the output vector which is then resolved by associations and their weights has been reported by Hagan *et al.* 2017 [7]. Number of nodes chosen for each layer depends upon the problem Neural Network is trying to resolve, type of data network it is dealing, quality of data, amount of information available and several other parameters as well. Numbers of input/output nodes are certainly depended on training set in hand has been reported by Ali *et al.* 2017 [8]. Larose, 2005 [9] asserted that determining the number of nodes in hidden layer can be quite demanding. Number of possible computation that an algorithm deals with is directly proportional to the number of neurons in hidden layer. If there are excessive neurons in hidden layer, the algorithm might be time consuming. Selecting fewer nodes in the hidden layer may thwart algorithm of its learning capability. Hence optimum equilibrium needs to be chosen. Also it is vital to scrutinize and observe the advancement of Neural Network throughout its preparation, if outcome are not humanizing, precise alteration to the model might be needed has been reported by Cilimkovic, 2015 [10].

3.2. Setting Weights

The Neural network is controlled by manipulating and adjusting weights between nodes. Initially the weights are set at some arbitrary numbers and subsequently adjusted during neural network training. In addition, as per report of Fogel, 2002 [11] focus should not be at changing one weight at time, but changing of all the weights be done concurrently. In some cases, neural networks consists of more than thousands, or even millions of nodes, therefore, tweaking one or two at a given point of time wouldn't help in modulating the network to get optimum results in time complex mode. Logic in the wake of weight updation is somewhat basic. Throughout the neural network training weights are reorganized after iterations, if results of neural network after weights updation are comparatively superior than the original weights get changed with the fresh ones and in this manner iteration progresses. Discovery of set of weights helping in minimizing error should be key endeavor while setting weights [9].

3.3. Activation Function

Activation functions are considered necessary for hidden layer of the neural network in order to establish nonlinearity. Without nonlinearity network would simply be a plain observation. Activation function can be a threshold or sigmoid function. Sigmoid activation function is generally employed for hidden layers simply as it combines nearly linear behavior, curvilinear behavior and nearly constant behavior relying on the input data [8]. For a better understanding of activation refer to Fig 2. SUM is referred to the set of the output nodes from the hidden layer which is multiplied by connection weights. Product is added to get single number, subsequently, put through sigmoid function. Input to sigmoid can be any value between negative infinity and positive infinity digit; on the other hand output value can only be between 0 and 1 has been reported by Knerr *et al.* 1990 [12].

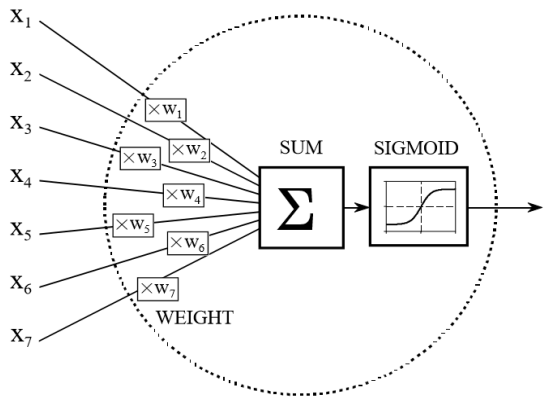

Fig 2 Activation Functions (Knerr *et al.* 1990) [12].

Running and Training Neural Network

Operation the Neural Network consists of two passes i.e *Forward Pass* and a *Backward Pass*. In the forward pass outputs are calculated meaning multiplication of weights and strength of numbers is then comparing with desired outputs. Inaccuracy between actual and desired output are calculated. While in backward pass the calculated error is employed to change the weights in the network in such a way that range of the error should get reduced in the next iteration of forward pass. This phenomenon is repeated until the error is low enough. For a general scenario let us consider a set of 1000 samples, we could employ100 of them for training the network and the remaining 900 for testing. This way we could train our network the best and it will show the desired result we expected [11]. Among the most well-known algorithm to solve such arduous problem is *back propagation algorithm* has been reported by Smith, 2011 [13].

4.2. Role of ANN in Penetration Testing

Now we have a basic idea of what is Penetration Testing and how Artificial Neural Network (ANN) works. We should now consider on how ANN can play important role in Penetration Testing. Software system has several inputs and output attributes, as a result number of combinatorial test cases is very high which makes it impossible to enlist and implement all of them. Meanwhile, it is mandatory and highly desirable for ensuring that nearly all the software faults have been properly detected and rectified. As a result, the choice of test cases is an insignificant predicament. One way out is to set up the criteria for choosing the most effective, important test cases and removing the redundant ones. There may perhaps be abundant perspectives of effectual or imperative test cases. Neural network training is blessed with predictive accuracy of the network for hidden data. Discovering the smallest and effective model which can fit the data and will be compatible for the hidden data is quite tricky. Two universal approaches to this dilemma have been reported [13]. The first one will be training a network that is bigger than required followed by pruning it. The second one will be by adding nodes and links to the network as and when required only. Further, the pruning algorithms can be subdivided into two major groups. Criterion used for removing the links is sensitivity of the error function to the removal of the link being the first group. Here the links which affect the error function to a minimum can be deleted. On contrary, the second group adds a penalty term to the objective function. This term penalizes redundant weights by reducing them to zero and adds on the important weights. It is often seen that weights with less significant magnitudes will not affect the accuracy of the network and can be safely deleted. Weights with larger magnitudes should invariably be retained as reported by Last *et al.* 2004 [14]. The predictive accuracy of the neural network remains very high than any other educating methods but it is quite difficult in understanding the induced concept because of complex architecture involvement. Therefore, it is highly desirable to extract classification rules from the network structure for explaining the decisions arrived at. Investigation based methods for rule-extraction often pretense restriction on values which can be taken by the concealed units. Here they search the subsets of combinations of inputs that exceed the predisposition on the concealed unit, thereby, diminishing the time complexity of the search and associates these values with inputs and outputs [14].

5.1. Role of Artificial Intelligence in Penetration Testing

1. Proper usage of Artificial Intelligence (AI) in Penetration Testing involves four major

steps: **1) Discover, 2) Learn, 3) Sense and 4) Respond** (Limited, Infosys, 2017) [15].

2. **Discover**: Create smart test cases using data repositories including defects, tickets, logs, etc. that can be used for analysis.
3. **Learn:** Identify relationship between test cases such as defects and software requirement document for developing insights.
4. **Sense**: Predict the occurrence of an incident, impact and likelihood led by analytics and insights.
5. **Respond**: Respond to an incident, input the resolution. Trigger automated response e.g. scripts to test targeted features which results in continuous learning.

III. CONCLUSIONS

In this paper we have first understood what is Penetration Testing and its tools, then what is Artificial Neural Network, and finally ANN's practicality in Penetration Testing. Artificial Intelligence (AI) algorithms learn from test cases to provide sagacious insights like application stability, failure patterns, defect hotspots, failure prediction, etc. These insights will further help to anticipate, automate, and enhance decision-making capabilities. ANN makes Penetration Testing way better for a tester by elimination of hundreds and thousands of redundant, vague, undesirable test cases. This, later, helps in decreasing the cost of the project and also the time consumption to finish a given project. Penetration Testing could be performed without the help of AI, but undoubtedly AI takes the testing to a whole new level. Leaving the tester ample of time to find and develops errors in many other safety loopholes of a system. In coming years, with the help of intense algorithms, AI will make Penetration Testing even better by generating new and relevant test cases to test a system. A new test case means better ways to secure a system. Finally, it can be safely concluded that AI is like a gift for the world of testing.

REFERENCES

[1]. Northcutt, Stephen et al. (2017) "Penetration Testing: Assessing Your Overall Security Before Attackers Do". (2017): n. pag. Print.

[2]. Using An Ethical Hacking Technique to Assess Information Security Risk. Rep. The Canadian Institute of Chartered Accountants / Information Technology Advisory Committee, June 2003. Web. 25 April 2017. <http://www.cica.ca/index.cfm/ci_id/15758/la_id/1.html>.

[3]. Six hours to hack the FBI (and other pen-testing adventures)"http://www.computerworld.com/action/ article.do?command=viewArticleBasic&articleId=9 087441", 2017-04-25

[4]. Bacudio, Aileen G et al. 2017 "An Overview Of Penetration Testing". International Journal of Network Security & Its Applications 3.6 (2011): 19-38. Web. 26 Apr. 2017.

[5]. Saindane, M. 2017. "Penetration Testing – A Systematic Approach," http://www.infosecwriters.com/text_resources/pdf/PenTest_MSaindane.pdf, accessed on April 26 2017.

[6]. Phatak, Prashant. 2017 "Top 10 Security Assessment Tools - Open Source For You". Open Source For You. N.p., 2017. Web. 26 Apr. 2017.

[7]. Hagan, M. T., Demuth, H., and Jesus, O. De. 2017 "An introduction to the use of neural networks in control systems," Int. J. Robust Nonlinear Control, vol. 12, no. 11, pp. 959–985, April 2017.

[8]. Ali Mumtaz, Dan E. Tamir, Naphtali D. Rishe, Abraham Kandel, 2017. "Complex intuitionistic fuzzy classes", Fuzzy Systems (FUZZ-IEEE) 2016 IEEE International Conference on, pp. 2027-2034, April 2017

[9]. Larose, D.T.. 2005. Discovering knowledge in data: an introduction to data mining. Wiley-Interscience. ISBN 9780471666578. URL http://books.google.ie/books?id=JbPMdPWQIOwC.

[10]. Cilimkovic, Mirza. 2015 "Neural Networks And Back Propagation Algorithm". (2015): n. pag. Print.

[11]. Fogel, D.B.. 2002 Blondie24: playing at the edge of AI. The Morgan Kaufmann Series in Evolutionary Computation. Morgan Kaufmann Publishers. ISBN 9781558607835

[12]. Knerr S., Personnaz L., Dreyfus G. (1990) Single-layer learning revisited: a stepwise procedure for building and training a neural network. In: Soulié F.F., Hérault J. (eds) Neuro computing. NATO ASI Series (Series F: Computer and Systems Sciences), vol 68. Springer, Berlin, Heidelberg.

[13]. Smith, Steven W. 2011. The Scientist And Engineer's Guide To Digital Signal Processing. 1st ed. [S.l.: s.n.]. Print.

[14]. Last, Mark, Abraham Kandel, and Horst Bunke. 2004. Artificial Intelligence Methods In Software Testing. 1st ed. Singapore: World Scientific. Print.
[15]. Limited, Infosys. 2017. "Artificial Intelligence (AI) In Software Testing - Offerings | Infosys". Infosys.com. N.p. Web. 1 May 2017.

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

*Sachin Chaudhury. "Penetration Testing An Effective And Versatile Tool For Software Security." International Journal of Engineering Research and Applications (IJERA), vol. 08, no. 01, 2018, pp. 52–58.