Brainwave: A Multidisciplinary Journal (ISSN: 2582-659X), Vol. 2, Special Issue No. 1, May 2021, pp. 50--59,

A Systematic Analysis of the Attack Pattern in Penetration Testing

Shantanu Mukherjee[1*], Paramita Chatterjee[1], Sandip Roy[1], Rajesh Bose[1]
[1]*Department of Computational Science, Brainware University, India*
[*]*shantanum@outlook.com*

**Corresponding Author*

Abstract

Data confidentiality has emerged as the prime concern in securing applications across the web. The hackers look for loopholes in the data integrity so that it could be broken and the confidential data could be stolen. Thus, the most preferred way to prevent web applications from revealing the data confidentiality is by scanning and eliminating those loopholes or so-called vulnerabilities. Adopting safe and secure development strategies also help in reducing the security vulnerabilities and they can further be fool proofed by engaging various security measures like intrusion detection systems, firewalls or the like. It started with manual code review, which was considered the best possible method to look for security vulnerabilities in the web applications. The advent of automated SAST and DAST tools brought about a revolution which could help perform security testing without requiring an expert drilling down the code and that too within quite a short span of time. Software security testing, like any other testing methodology, can be broadly split into two categories - Black-Box Testing and White-Box Testing. In this paper I work on the various attack patterns which may be used to elicit the best results out of any penetration test.

Keywords: information security, penetration testing, benefits, attack pattern, procedural, structural, pattern repository.

1. Introduction

There are various approaches to scan web applications for security vulnerabilities, one of the most common ones being static analysis of the source code. As the name suggests, the assumption here is, the source code is available for thorough scrutiny. The logic applied in this process is that if the user input received by the application is able to affect the syntactic structure of the sensitive operation arguments then the application would be vulnerable to injection attacks [1]. However, this static analysis procedure is also not free from drawbacks, the major one being a high rate of false positives arising due to analysis imprecisions.

Another prominent approach available for testing security vulnerabilities is that of analysing the web application from an end user standpoint. The basic premise of this approach is that the source code is not available and hence the scanning is performed by forcing all sorts of malicious inputs like cross site scripting and others to exploit the vulnerabilities, if any. The analysis of the application errors indicates possible vulnerabilities; however, the drawback of this approach is that it does not guarantee accuracy or completeness of the testing results.

Penetration testing is the most common technique adopted in this approach. Penetration testing simulates the attacking techniques used by a hacker [2]. These attacks are aimed at stealing or manipulating data.

Classification of Penetration Testing:

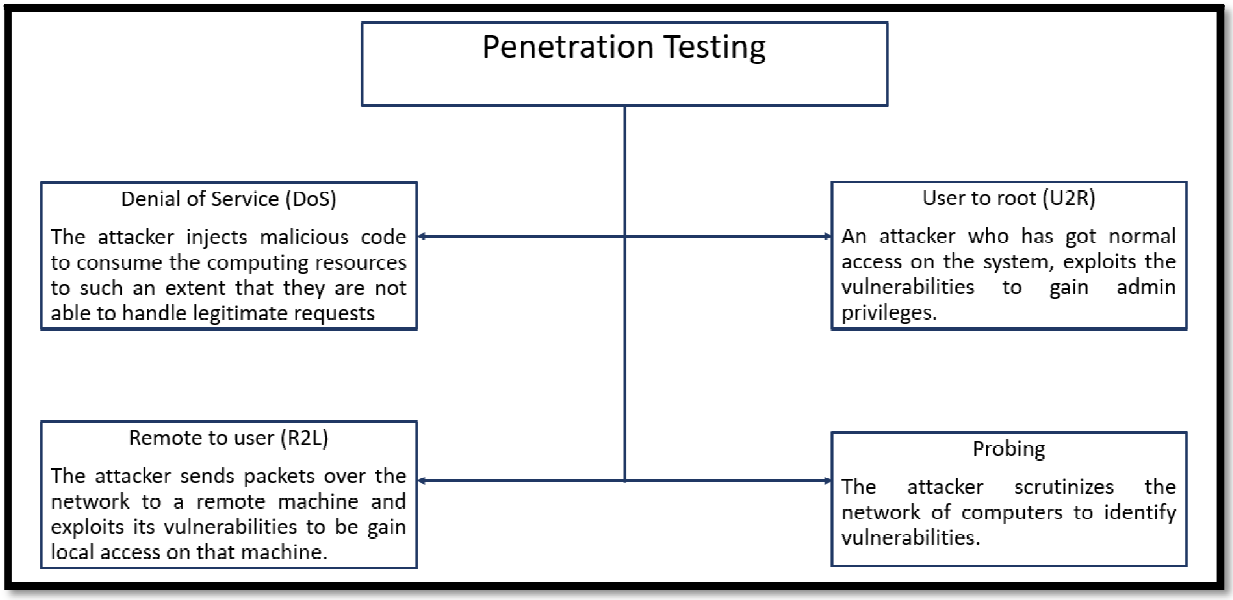

Figure 1.Classification of Penetration Testing

Penetration Testing follows the given sequence of steps:

- Conducting and initial reconnaissance on the target system
- Scan and consolidate information about the available services and protocols
- Detecting and gaining access to the applications running on the target system
- Applying known methods to exploit the system
- Exploring options of penetrating into the system

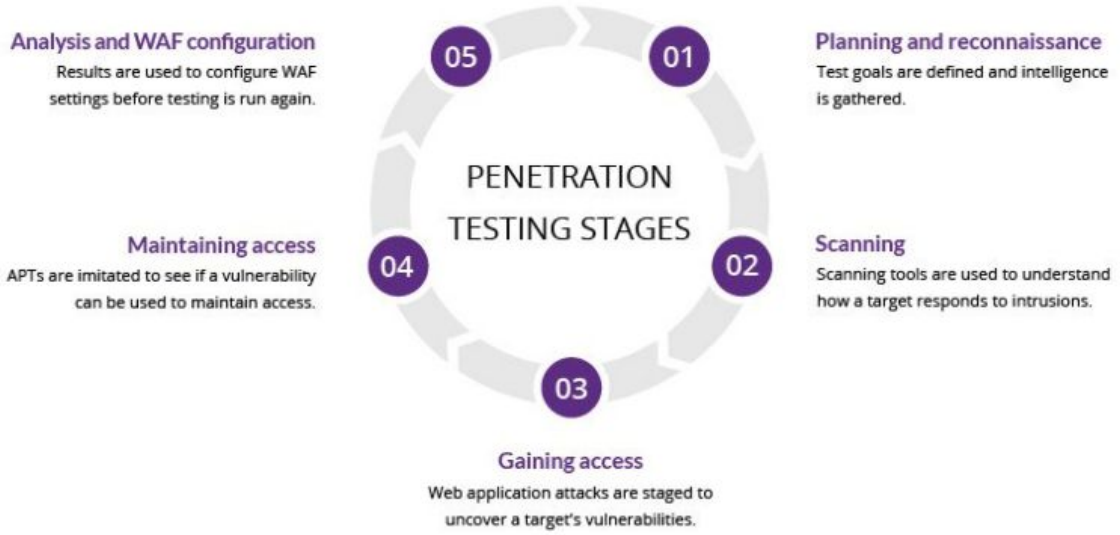

Figure 2: Stages in Penetration Testing

2. Related works

In [4] A methodology for automated penetration testing of cloud applications, the authors present a methodology that enables the automation of penetration testing techniques based on both application-level models, used to represent the application architecture and its security properties in terms of applicable threats, vulnerabilities and weaknesses, and on system-level models, adopted to automatically generate and execute the penetration testing activities.

In [5] A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment, the authors focus on empirical papers, and based on the findings, and identify a number of potential research challenges and opportunities, such as scalability and the need for real-time identification of exploitable vulnerabilities.

In [6] Systematic Literature Review on Penetration Testing for Mobile Cloud Computing Applications, the authors pursue two objectives, namely, provide a comprehensive systematic literature review of Mobile Cloud Computing, security and penetration testing domains and to establish the requirements for penetration testing of Mobile Cloud Computing applications.

In [7] Web Services Attacks and Security- A Systematic Literature Review, the authors presented a systematic review on the studies of web service security.

In [8] Threat modelling – A systematic literature review, the authors conclude that, most threat modelling work remains to be done manually, and there is limited assurance of their validations.

In [9] Safety and Security Co-Analyses: A Systematic Literature Review, the authors focused on the early system development stages for combined safety and security approaches that consider the mutual influence of safety and security.

In [10] Autonomous Security Analysis and Penetration Testing, the authors propose an autonomous security analysis and penetration testing framework (ASAP)that creates a map of security threats and possible attack paths in the network using attack graphs.

In [11] Constructing a Benchmark Dataset for Advanced Persistent Threats, the authors benchmark DAPT 2020 dataset on semi-supervised models and show that they perform poorly trying to detect attack traffic in the various stages of an Advanced Persistent Threats.

In [12] NIG-AP: a new method for automated penetration testing, the authors work on an algorithm that formalizes penetration testing as a Markov decision process and uses network information to obtain the reward, which guides an agent to choose the best response actions to discover hidden attack paths from the intruder's perspective.

In [13] Comprehensive study of software testing: Categories, levels, techniques, and types, the author presents a comprehensive study of software testing methods. An explanation of Testing Categories is presented first, followed by Testing Levels then Testing Techniques.

3. Penetration Testing and Its Benefits

From the Business Standpoint

Millions of dollars are spent to enforce security mechanism in any application. These mechanisms safeguard the organizations against failures, help manage keep up the ethical reputation, maintain the trust of the customers and shareholders and keeping the corporate image high. Even a remote possibility of compromising the client data can be severely damaging. Dent to the reputation and customer confidence can throw an organization out of business [14]. Penetration testing inculcates the required awareness regarding information security at every level of the organization. Penetration testing is a proactive measure to identify the security loopholes so that they may be plugged to prevent security breaches.

From the Operational Standpoint

Penetration testing helps in measuring the impact and the possibility of the vulnerabilities. This enables the organization to prioritize and execute remedial measures for the reported vulnerabilities [15]. This helps in moulding the information security structure to fine tune it against identified risks and eliminate any cascading effects of security risks. Security testing is a very time-consuming process and involves thorough knowledge and efforts to handle the complex testing landscape. Penetration testing thus helps the organizations improve and test patches to proactively remove risks.

Types of penetration testing:

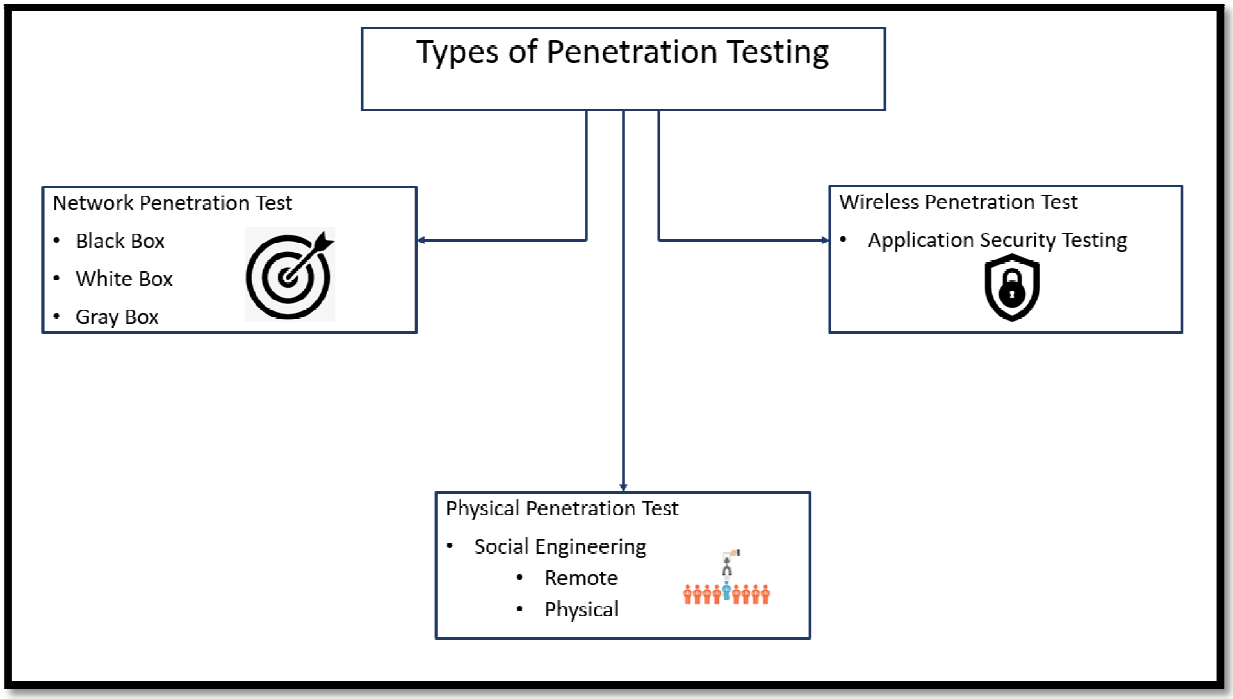

Figure 3: Types of Penetration Testing

Network Penetration Testing:

Network Penetration Testing: In this testing, the physical landscape of the system is put to test to identify the vulnerability and risks. In this, a tester explores thenetwork environment for security loopholes in design, implementation, or operation of the network. Devices like computers, routers, modems and other remote installations are included in the testing scenario.

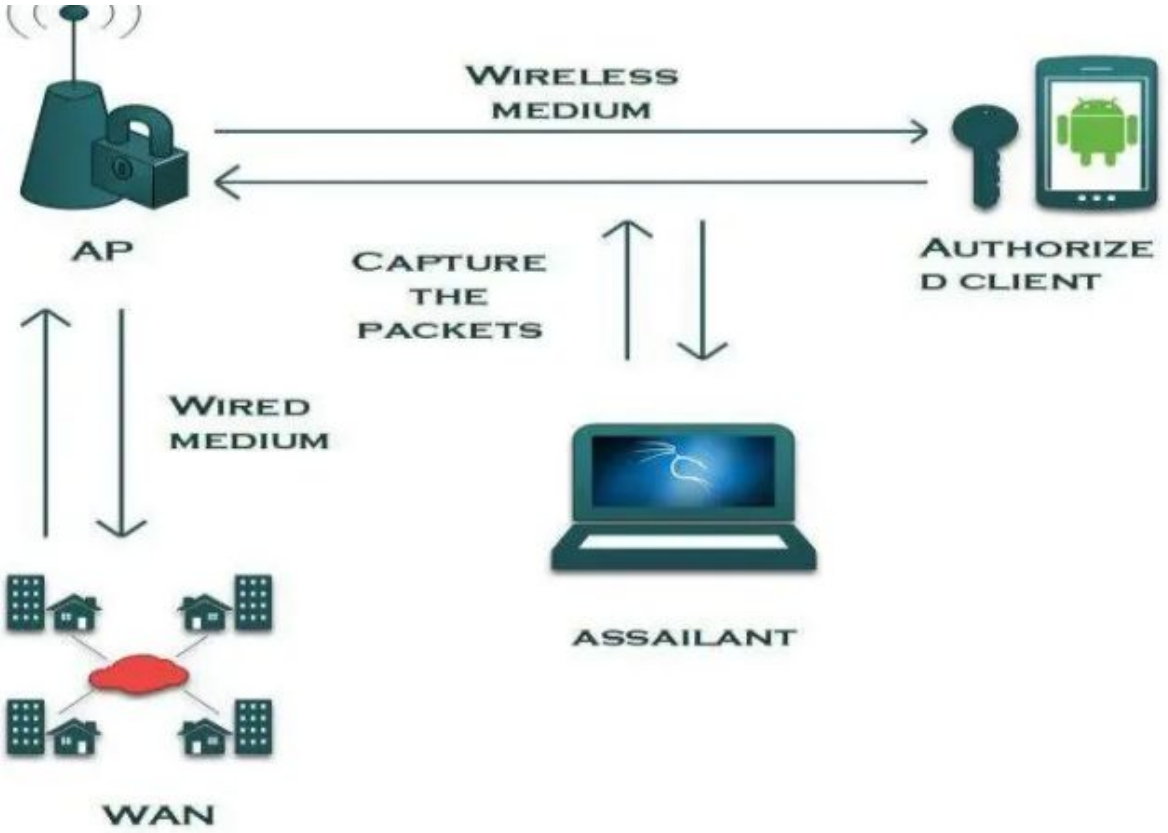

Figure 4: Network Penetration Testing

Wireless Penetration Testing:

Now a days, all the companies host browser-based applications. The security integrity of these applications is the target area in Wireless penetration testing [16]. The attack domain could be anything ranging from the application or the web services to the web user interfaces. The attacking mechanism simulates the steps adopted by the hackers to drill a hole into the security infrastructure to gain access to the organizations' internal systems.

Social Engineering Penetration Testing:

In social engineering type of penetration testing, the focus is on people and processes rather than the hardware or the technology associated with any network infrastructure. The testing methods include attacks like phishing, vishing, smishing, impersonation, USB drops, dumpster diving, tailgating and the like.

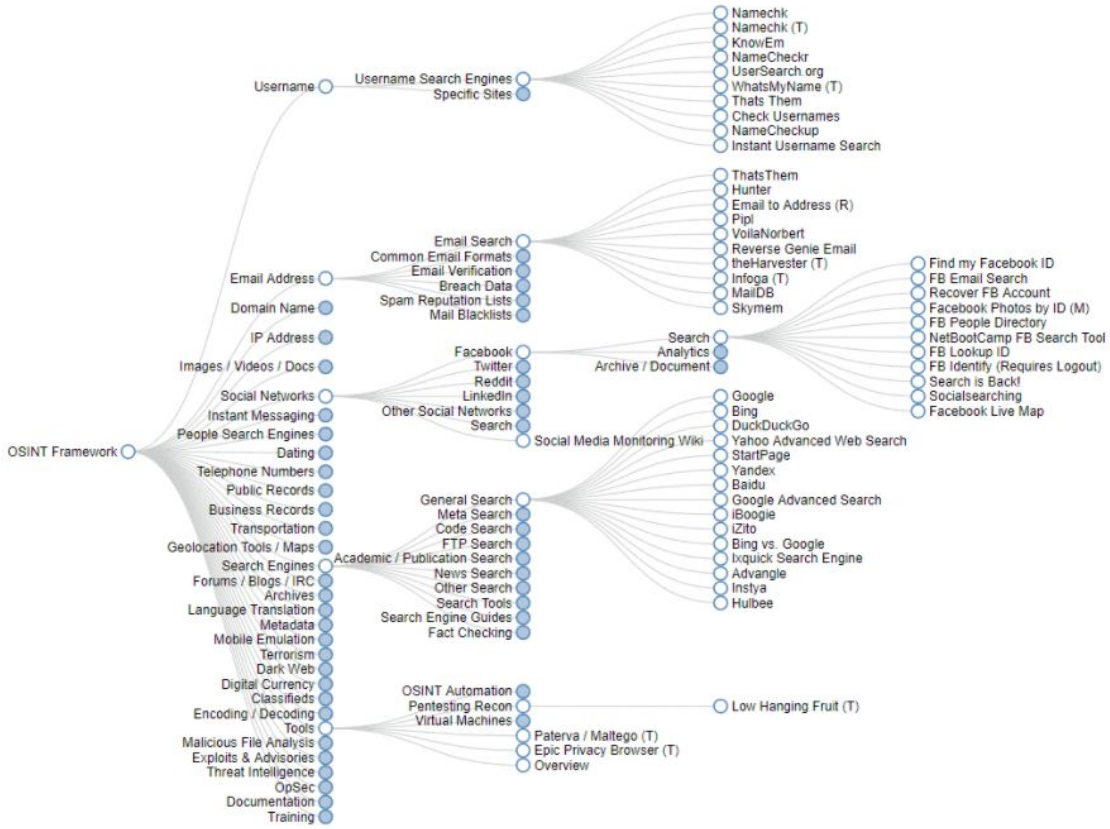

Figure 5: Social Engineering Penetration Testing

4. Attack Patterns

The security experts and the developers come from different worlds with regard to their thoughts. Security professionalsfocus on the security of thesystem [17], while developers want to build a working system, unfortunately the non-functional goal of security comes quite at the bottom of a developer's priority list. There is always a tussle going on between the developers and the security folks where the security folks complain of the developers ignoring the security aspect of the application whereas the developers debate that they need to take care of numerous aspects and not just the security. This is where the attack patterns build the bridge. An attack pattern is the general knowledgebase required to execute a specific attack so as to exploit particular types of security weaknesses [18][19]. These are the approaches adopted by hackers to exploit the application. They are tuned to be used and comprehended by the developers who may miss the securityfine print. While the primary focus area is security, however, the attack patterns also aim to bring to forth the strengths and weaknesses of various approaches to enable the developers factor the gathered information to make intelligent and well-informed decisions between security and other goals. Attack patterns provide the required foundation in the form of worked solutions and guidance to implement the security parameters in any application.

The attack patterns approach has pervaded the entire software engineering domain.For example, may it be the Java APIs or the Microsoft Foundation Classes, all use the patterns itemized in Design Patterns. Attack patterns twist the problem-solution paradigm of design patterns in a destructive rather than constructive context [20]. Attack pattern is a generic representation of a

deliberate, malicious attack that may occur in specific scenarios with an aim to educate the developers' community on how software can be exploited in reality and how they may be avoided. Businesses and governments have always preferred to hide the attack information for they may dent the organizations' reputation or the client confidence in them, or even worse, the attackers may try similar attacks again on the same organization or on another organization [21]. Thus, a substantial amount of attack information is off the books but still increased public interest and awareness has helped in documenting numerous of them in books, blogs, newgroups or the internet. The security engineers can now use this data in a more streamlined way to strengthen information system security and its endurance.

Attack patterns are mainly of structural types. Following are some of the examples of structural attack patterns:

Pattern Name	Brief Description
Account Lockout	Passwords are the key to authenticate any remote user. This makes password guessing a hot area to crack weak or poor passwords. Account lockout, although, are an efficient way to prevent automated password guessing but sometimes, this technique locks out the genuine user also.
Authenticated Session	This technique allows the user avoid reauthentication on every page request, thereby allowing the user access multiple pages with a single authentication. This pattern incorporates user authentication into the basic session model.
Client Data Storage	In the current scenario, it is somewhat necessary to store data on the client machines using the cookies, hidden fields or URL parameters. However, this always leaves a room of suspicion as the data is at the clients' disposal. Encryption techniques used in storing the data can safeguard against data tamper.

Documenting the patterns is necessary to describe how a type of attack is executed. The abstraction mechanism is to be depicted to describe the relationship among certain patterns based on the nature and its applicability. The subject matter of the attack pattern, the pattern type and the audience to use, form the blueprint of any exploit. A pattern repository or database is required to maintain the information regarding attack patterns. The pattern repository should contain the following information about an attack pattern so that it can be objectively used without an iota of doubt. A lucid and comprehensible attack pattern repository is one which enunciates each pattern in understandable form for the developers, readers and entire security fraternity [24]. An attack pattern in a pattern repository should ideally include the below information:

Name of the Pattern: The pattern identifier
Prerequisites: The pre-required conditions and the characteristics of the target software
Description: Attack description mentioning the actions to be taken
Related Vulnerabilities: The specific vulnerabilities which the attack leverages
Attack Procedure: The attack vector used, like data injection etc
Consequence: The end objective which is to be achieved
Mitigations: Actions which could mitigate the attack
Relevant Context: The technical aspect in which this attack pattern is pertinent

Although not mandatory it is always nice to have additional information about the attack pattern as follows:

Examples: Demonstrative exploits instances to help understand the nature, context and variability of the attack in practical terms.

Related Attack Pattern: Which other attack patterns affect or are affected by the attack pattern under consideration.

Related Design Patterns: All the design patterns which are susceptible to these attack patterns or any design pattern which is fool proof against the attack pattern under consideration.

Related Security Patterns: Any security pattern that may provide the required resistance to this kind of attack.

Probing Technique: Techniques used to scout a potential target to identify the security loophole.

Indicators: Activities, events or conditions that may indicate the imminence or occurrence of this kind of attack.

Disguising Methodology: Any technique that may be used to obfuscate this kind of attack.

Finally, it is an important step that how these patterns are constructed to avoid and discover the security threats. The security requirements are to be framed with attack patterns along with the application development process [25]. Here the requirement is initially defined in terms of system engineering and then combined with security requirement which finally refers the pattern repository for a specific pattern. The definition phase is about understanding the requirements followed by pattern relevance to the requirement. The requirement is defined in terms of various patterns that are applicable. Then each of the patterns is defined in terms of requirement that is associated. The output of the definition phase is set of all requirement-attack pattern pairs that are identified. The design phase is about designing the patterns along with the requirement. The patterns are designed with various solutions for each of the attack problem. It is inevitable that attack patterns are destructive rather than constructive in nature [26]. From the pattern repository various solutions are compared to the corresponding pattern problem and the right problem-solution pair is designed. The outcome of the design phase is evaluating all the problem-solution pair for respective pattern associated to the requirements.

5. Conclusion

The penetration security engineering is an open field with much more work to be done for the industry. Much of the work done in attack pattern is all about how security requirements are treated along with the system requirements. The security community is expecting more form the researchers and academicians in this direction. The proposed paradigm discussed in this paper is all about and how these attack patterns are integrated with the security requirements.To make this paradigm as standard in the industry one must need supporting tools and techniques and also more support towards terminology the security community. In future, I would like to extend my

work use various penetration testing tools available, to analyse various repositories how well the penetration testing tools map with the attack patterns.

References

[1] AkashdeepBhardwaj, Syed Bilal Hussian Shah, Achyut Shankar, MamounAlazab, Manoj Kumar &Thippa Reddy Gadekallu, *"Penetration testing framework for smart contract Blockchain"*, Peer-to-Peer Netw. Appl. (2020).https://doi.org/10.1007/s12083-020-00991-6. September 2020

[2] Ghanem, M.C.; Chen, T.M. *"Reinforcement Learning for Efficient Network Penetration Testing" Information* 2020, *11*, 6. https://doi.org/10.3390/info11010006

[3] SujitaChaudhary, Austin O'Brien, ShengjieXu. *"Automated Post-Breach Penetration Testing through Reinforcement Learning"*, 2020 IEEE Conference on Communications and Network Security (CNS) DOI: 10.1109/CNS48642.2020.9162301 August 2020.

[4] ValentinaCasola, Alessandra De Benedictis, MassimilianoRak*"A methodology for automated penetration testing of cloud applications"*, International Journal of Grid and Utility Computing (IJGUC), Vol. 11, No. 2, March 2020

[5] Dean Richard, McKinnelaTooska, DargahiaAliDehghantanhab, Kim-KwangRaymondChooc*"A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment"*, Computers & Electrical Engineering https://doi.org/10.1016/j.compeleceng.2019.02.022, Volume 75, Pages 175-188, May 2019,

[6] Shao-Bo Li, Jing Yang, Zheng Wang, Shu-De Zhu, Guan-Ci Yang *"Review of Development and Application of Defect Detection Technology"*, ActaAutomaticaSinica, Volume 46 , Issue 11 : 2319-2336(2020) https://doi.org/10.16383/j.aas.c180538, December 2018.

[7] Ahmad Salah Al-Ahmad, HasanKahtan, FadhlHujainah, Hamid A. Jalab. *"Systematic Literature Review on Penetration Testing for Mobile Cloud Computing Applications"*, IEEE Access (Volume: 7), DOI: 10.1109/ACCESS.2019.2956770, November 2019.

[8]VarshaR.Mouli, K.P.Jevitha, *"Web Services Attacks and Security- A Systematic Literature Review",* 6th International Conference On Advances In Computing & Communications, ICACC 2016, September 2016.

[9]XiongWenjun, RobertLagerström*"Threat modeling – A systematic literature review",* Division of Network and Systems Engineering, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, https://doi.org/10.1016/j.cose.2019.03.010 March 2019.

[10] E. Lisova, I. Šljivo and A. Čaušević, *"Safety and Security Co-Analyses: A Systematic Literature Review,"* in IEEE Systems Journal, vol. 13, no. 3, pp. 2189-2200, Sept. 2019, doi: 10.1109/JSYST.2018.2881017 September 2019.

[11]AnkurChowdhary, Dijiang Huang, JayasuryaSevalurMahendran, AbdulhakimSabur*"Autonomous Security Analysis and Penetration Testing",* The 16th International Conference on Mobility, Sensing and Networking, Tokyo, Japan, September 2020.

[12]SowmyaMyneni, AnkurChowdhary, AbdulhakimSabur, Myong Kang
"DAPT 2020 - Constructing a Benchmark Dataset for Advanced Persistent Threats" The First International Workshop on Deployable Machine Learning for Security Defense at SigKDD 2020, July 2020.

[13]Tian-yang Zhou, Yi-chaoZang, Jun-hu Zhu, Qing-xian Wang *"NIG-AP: a new method for automated penetration testing",* Frontiers of Information Technology & Electronic Engineering 20(9):1277-1288 September 2019.

[14] Mubarak Albarka Umar *"Comprehensive study of software testing: Categories, levels, techniques, and types"* International Journal of Advance Research, Ideas and Innovations in Technology, November 2019.

[15] Dang Jingxiong*"Analysis of Cyber Security Threat Environment and Information Security System of Financial Industry Under New Situation",*Manipal International University, 71800 PuteraNilai, Negeri Sembilan, Malaysia January 2020.

[16]MusbahAbobakerMusbah*"Information Security Management Requirments Model Through Data Life Cycle"* The International Journal of Engineering and Information Technology (IJEIT), VOL.6, NO.1,2019

[17] Tomasz Stefaniuk, *"Training in shaping employee information security awareness entrepreneurship and sustainability issues"*ISSN 2345-0282 (online) http://jssidoi.org/jesi/ 2020 Volume 7 Number 3 http://doi.org/10.9770/jesi.2020.7.3(26) March 2020

[18] Agnes Åkerlund and Christine Grobe*"Integration of Data Envelopment Analysis in Business Process Models: A Novel Approach to Measure Information Security"* 6th International Conference on Information Systems Security and Privacy, January 2020

[19]Yubao Wu *"The Effect of Information Security Education Based on Cyberspace Security"* 7th International Education, Economics, Social Science, Arts, Sports and Management Engineering Conference (IEESASM 2019).

[20]Emad Ahmed Farouk, Salah Ahmed Elgendy, Mahmoud Adel Mahmoud *"A Quick Survey on Information And computer Security"* Journal of Computer Science and Information Systems, Volume 1, Issue 2, May 2019.

[21]NileshBhingardeve and Seeza Franklin *"A Comparison Study of Open Source Penetration Testing Tools"* International Journal of Trend in Scientific Research and Development (IJTSRD) ISSN No: 2456-6470, Vol 2, Issue 4, June 2018.

[22]Abikoye, O.C., Abubakar, A., Dokoro, A.H. et al. *"A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm."*EURASIP J. on Info.Security 2020, 14 (2020). https://doi.org/10.1186/s13635-020-00113-y August 2020.

[23] Craig A Horne, Sean B. Maynard, AtifAhmadAtif Ahmad *"A Theory on Information Security: A Pilot Study"* 14th Pre-ICIS Workshop on Information Security and PrivacyAt: Munich, Germany, December 2019.

[24]ElhamRostami, Fredrik Karlsson and Shang Gao*"Requirements for computerized tools to design information security policies"* Available online at www.sciencedirect.com September 2020.

[25]Yogesh P. Surwade and Hitendra J Patil*"Information Security"* Conference: National Seminar on "Library as a Knowledge and Skill Development Center"At: Aurngabad January 2019.

[26] Margit Scholl *"Information Security Awareness"* The 22nd World Multi-Conference on Systemics, Cybernetics and Informatics July 2018.