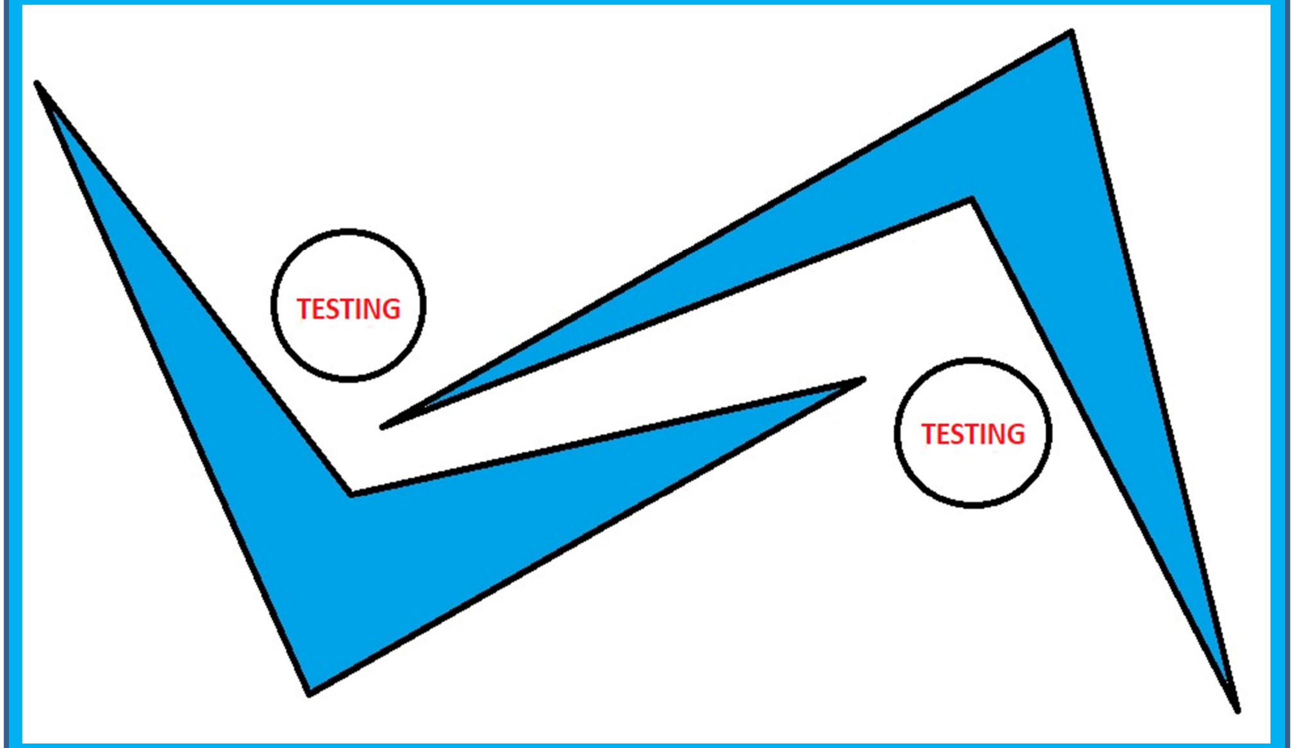
Tools & Best Practices of Penetration Testing Vs. Bug Bounty



Kavinda Herath

Sri Lanka Information Technology – Research Unit

ACKNOWLEDGEMENT

The theme which was selected in this booklet is highly demanded subject matters. I highly Appreciate Dr, Lakmal for being given his full support and guidance to succeed in this booklet. Writing on this book is quite challenging in terms of still digging deep into the developing subject matters however most researchers intended their afford on this without hesitation near future. Dr. Lakmal has been given me a task as this dissertation to be accomplished here is my achievement thank you, sir. Without a team of the SIIT, this is still being a dream for me no drought on that. Take this opportunity to show my humble gratitude to every one of the Sri Lanka Institute of Information Technology at Malabe and Kollupitiya campuses.

CONTENTS

Contents ……………………………………………..………………….………….……...….….iii
List of Figures …………………………….……………………………..…………….……….iv
List of Tables …………………………………………………………………......…..…….……iv
Chapter 1: Introduction …………………………………………….……….……….…………..1
1.1 Background Introduction ……………………………………………………………1

Chapter 2: Penetration Testing ……………………………………………….………..………..4
2.1 Vulnerabilities ………………………………………………………….…………...…4
2.2 Penetration Testing Fundamentals ……………………………………...…….….……4
2.3 Penetration Testing Stages …………………….…………………………….…………4
2.4 Reporting Phase ……………………….……………………………………………….6
2.5 Vulnerability Assessment And Penetration Testing Tool ………..……………………8

Chapter 3: Bug Bounty ………………………………………………………….………….…..10
3.1 Introduction to Bug Bounty ………………….………...……………………….……10
3.2 Bug Bounty Process ………………………………...……………….………………..12
3.3 HackerOne ……………………...……………..………….…………………..13
3.4 Rewards ……………………………..………………………………………………..13
3.5 Kali Linux for bug bounty ……………………………………………………………14

Chapter 4: Attack ………………………………………………….……………………………..15
4.1 Brute Force Attack.……………………………...………………………………………15
4.2 MITRE ATT & CK framework ……………………………………………………….16
4.3 Attack Navigator Tools on YouTube presentations ……………………………………..16
4.4 APT attack techniques ………………………………………………………………..16
4.5 SonarQube …………………………………………………………………………….17

Chapter 5: Hacker Boxes ……………………………….…….…………………….…………..19
5.1 Hacking Tools…..………………...……………...………………………………………19
5.2 Hacking Tools ……………......……………………………………………………….20
5.3 Bug Report ……………......…………………..…...……………………………………21

Chapter 6: Penetration testing for Cryptographic Vulnerabilities …………………………………..22
6.1 Common Cryptographic Vulnerabilities ……………………………………………….22
6.2 Finding Vulnerabilities from CryptoGuard …………………………………..……….22
6.3 CrySL Tool ……………………………………………………………….………..23

Chapter 7: Evaluation of Modem Pen test ………………………………..……………………….25

Conclusion ……………………..…………………………………………………...…………….25
Appendix ………………………………………………………………..……………………..27

LIST OF FIGURES

Figure 2.1 Penetration testing lab requirements ……………………………………..……...…….6

Figure 2.2 Penetration testing lab requirements ……………………..……………………………6

Figure 2.3 Type of Vulnerabilities according to Industrial Control System ……………………..…9

Figure 3.1 Bug Bounty Variants ………………………………...……………………......12

Figure 3.2 Bug Bounty Lifecycle …………………………………………………...…….13

Figure 4.1 CSRF attack …………………………………………………………..……….15

Figure 4.2 ATP attack progression ………………………………………...……………..17

Figure 4.3 SonarQube cycle ………………………………..……………………………..17

Picture 4.4 SonarQube analysis of the code ………………………………………………18

Picture 4.5: Error mechanism of SonarQube ……………………………………………..18

Figure 5.1 Common tools of hack box ………………………………………………………19

Figure 6.1 Cryptographic misuse detection from CriptoGusrd ……………………………..22

Figure 7.1: List of web application scanners …………………………………………..……25

Figure 7.2: Scan Barrier Support Commercial ……………………………………..……..26

Figure 7.3: SSDLC Feature Support Commercial ……………………………………..…26

LIST OF TABLES

Table 2.1: Testing Tools and Common Vulnerabilities ……………………………….…….8

Table 4.1: MITRE AT& CK attack for enterprise ………………………………………………16

Table 7.1: Path Traversal Commercial ……………………………………………………….27

Table 7.2: RFI Commercial …………………………………………………..…………27

Chapter 1: Introduction

Offensive security is a proactive approach for prevention and detection from the security threats breeches into of the vulnerabilities particularly valuable asserts such as system application. The valuable truth statement for ever would be "Prevention is better than cure". Vulnerabilities are highly prevalent to exploit the applications even though tough security arrangements are implemented. The security arrangement of the Initiatives of the cording is a must eliminating the risk of the vulnerabilities from the attacks. The loopholes could be everywhere obviously due to the improper mechanisms of the security practices. Penetration testing gives a huge impact on vulnerability assessment as a technical framework. It covers a plethora of scope among the cyber security domain to control the security breaches.

Penetration is a systematic approach of testing mechanism applicable to either software or application in any platform such as open-source or windows. Technical specialists are accredited to proceed can be handled the penetration testing in the given piece of software or application. Pretesting testing tools are in high demand due to the availability of the standard products which are supported to scan the weakness of the software program.

A bug bounty is the other most comprehensive evaluation technique is intended to implement security vulnerabilities in the crowdsource at any given time. Vulnerability disclosure programs are popular best practices among companies for encouraging and knowledging about the vulnerability discovered in the world. Particularly, bug bounty best practices are expert individuals interested in personal rewards whereas penetration testing is a group event with several strategies and scopes. This document is emphasized several penetrations and bug bounty techniques along with living specimens gaining knowledge that are interested in further research.

1.1 Background Introduction

Penetration testing is the ideal security assessment of the relevant application such as web, mobile or API analyzing and progressively attacks to check the security posture. Several sophisticated tools are available to perform several penetrating testing to explorer the existing issues and vulnerabilities merged with software applications transmission via insecure media. The tools in which compatible are high demanding and expensive for penetration testing. The penetration testing process is the most critical phenomenon of the security aspect in the enterprise applications or rather operating systems that are currently established.

Continuous alerting on weaknesses of the systems might be the most suitable precaution that can practice as a detecting mechanism rather than waiting until devastating circumstances to attempt on the issues. However, multiple factors depend on the successful penetration testing process to align along with ongoing applications by least interruption with the enterprise goals such as business continuity plan as governs perspective. Perhaps the most critical factors are considered as time, finance and resources to be managed to overcome the project proposal. Outsourcing is given a high impact on the company while consideration of the consequences that could have been faced.

Detection of the vulnerabilities and the prevention mechanisms both are equally important end of the day regardless of the circumstances. Several stages of the system design life cycle were contributed to sorting several vulnerabilities. Multiple techniques are used to mitigate the weakness of the application before deployment. The weakness of the systems could be endless until performed a zero-day attack. Penetration testing is an endless process to keep optimization of customer satisfaction.

Penetration testing is the most suitable technical mechanism that uses to find errors in the source code. This document is intended to explorer further identification of the cryptographic vulnerabilities that can be exploited intruder activities that can be harmed into the applications. Malicious applications such as several malware functions are intended to exploit cryptographic vulnerabilities that can exist with enterprise applications such as ERP systems or process dependency software programs. The contents of this booklet mainly emphasized issues and vulnerabilities in applications that can be caused to exploitation intentionally and unintentionally.

A sophisticated dynamic brute force attacks were breached unspecified vulnerabilities. It may emphasize the importance of a continuous bug finding process or penetration testing in scheduled and random sequences to avoid uncertain circumstances. It would be a challenge to find potential parameters to overcome the vulnerabilities of the web application. Penetrating testing is making a huge impact on security breaches.

For quick research with immediate result oriented both penetration testing and bug bounty concepts are more practical to use. The dynamic approach would be more convenient than the static method to find out the vulnerabilities and the security breaches. Expectation becomes a reality of the intruders if they could gain overwhelming privileges from the source being exploited. Some of the vulnerabilities are well known such as buffer overflows in which user-level application programs.

Kali Linux tools are the more sophisticated and result-oriented open-source kernel merged with penetration testing tools for best practices ever found. Kali is the most suitable tool to gain knowledge indeed for penetration testing and bug bounty practices.

In contrast, Offensive security approach would be the vulnerability disclosure mechanism will be changed exponentially. Know about more in the offensive security: https://www.offensive-security.com/. Vulnerability databases and their source identification are quite interesting segment of the offensive security. Here is the link of vulnerable database for further evaluation: https://www.exploit-db.com/.

Chapter 2: Penetration Testing

2.1 Vulnerabilities

Vulnerabilities are some sort of components of source code in the software program intended to exploit implicit or explicit security policies.

The disclosure of the vulnerability source code might be the challenge the software developers could be faced. Instead of source code, several other factors are potential to malfunctions caused for exploitation behavior of a set of conditions present in a system as policies violation. .

2.2 Penetration Testing Fundamentals

Penetration testing is a part of a system development life cycle or DevOps process before final deployment. As a policy, scheduled and unscheduled vulnerability testing must be enhanced proper security prevention mechanism. Penetration testing is no more the newest technique that could apply to the industry due to the dedication of the pen testers who have been engaged with several years on the particular enhancement of the proper initiatives to support in this industry. Especially open-source implementation could be the huge and critical phenomenon of the testing to clarify the several vulnerabilities that have been identified particularly in the Linux kernel. It shows that how critical the implementation still intended to find some vulnerability that can be compromised several security breaches intentionally or unintentionally.

Zero-day vulnerabilities and SQL injection attacks are exploited frequently as the facts published by authorities on their websites. Another way it will be emphasized is that penetration testing is a must for a continuous process to apply any system in their security framework. Internal and external penetration testing are two different categories whereas internal vulnerabilities handled by the team itself inside the establishment and external could be the most crucial aspect of the overall attention with expecting a security risk of exploitation possibilities into the system. Figure 2.1 depicted that the basic requirements of the penetration testing initial preparation and the clarification of the item list to be kept.

2.3 Penetration Testing Stages

Currently, the process of pen testing could be the systematic approach whereas applicability of well suited predesign technics and using sophisticated tools that are enhancing the efficiency and the reliability of the results. As shown in figure 2.2 Vulnerability Assessment and Penetration Testing are divided into 9 sequential processing steps. The scope is decided the general procedure of the testing plan which is most suitable to applied like a black, grey or white box. Then, to be collected the relevant information to proceed such as the operating system, network, and IP address in the reconnaissance step.

To identify the vulnerabilities, the tester should precede the various vulnerability assessment techniques in this phase. As a result of the vulnerability test, the tester makes a plan for pen-testing. Evaluate the plan with a client by emphasizing the effect and the outcome before penetrating the victims system. As a result of the testing, the tester increases the privilege. After the final analysis, the tester recommended a procedure for resolving the vulnerability from the system. Documentation is a must for all the activities to take the necessary action by the management. Finally, clean up and restore the system in the previous state as it was before the venerability assessment test process was started.

There are several vulnerabilities that exist in a real practical environment to discuss further access control vulnerability, Boundary condition vulnerability, Input validation vulnerability, Authentication Vulnerabilities, Configuration Weakness Vulnerabilities, and Exception Handling Vulnerabilities, etc

- **Pre-Engagement Phase**

The most primitive stage whereas starting the preparation of the pen testing initiatives with proper discussion with the business partner indeed with several authorities to face the potential information gathering. In this stage should be clarified exactly what are the basic granted provisioning and the company policies conflict with exploitation and consequence outcomes how to affect the Business Continuity Plan (BCP). Further, the possible downtime and the time frame are the crucial factors to be discussed at this stage. Finally, the payment and the non-discloser agreement should negotiate before the commencement of the aligned task on this phase.

- **Information-Gathering Phase**

There are important factors to be collected on the system whereas the related internal and external integrated intangible and tangible source of the company particularly documented for further convenience on this phase. Suppose, it could help to list out the appropriate suited tools to select for further testing. Correct and accurate information helps to start penetration testing with high enthusiasm.

- **Thread Molding**

Threat molding is based on the information gathering phase where gathered the source of details about the existing system based on the advantages of the intruders instead of the exploitation into the system and penetrating the potential provisioning paths that can be possible.

- **Vulnerability Analysis Phase**

In this phase, tester attempt to discover the system vulnerabilities before that can be advantage the exploitation phase.

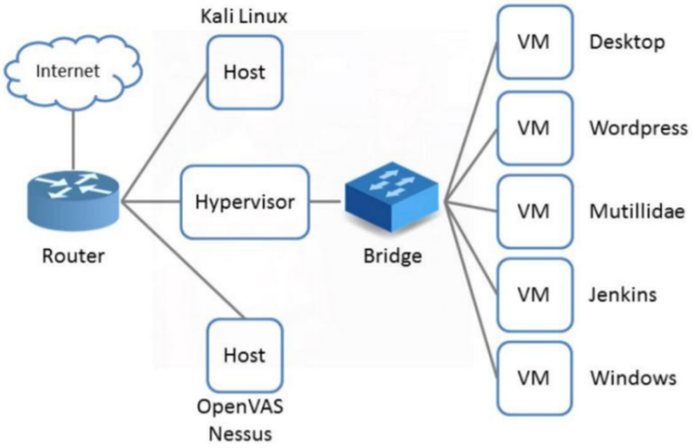


Figure 2.1 Penetration testing lab requirements

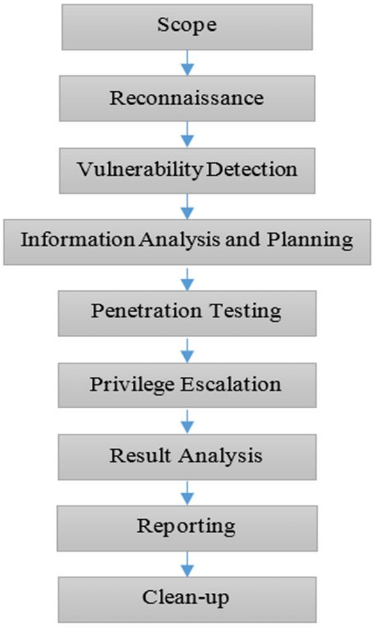


Figure 2.2: Vulnerability Assessment and Penetration Testing Life cycle

2.4 Reporting Phase

Tester summarized the findings and documented for standards of the pen testing further emphasized on the Penetration Testing Execution Standard (PTES) at http://www.pentest-standard.org/

Pre-engagement Before starting the penetration testing it could be very important that interact with the client on the overall scope and other critical factors everyone is on the same page about the penetration testing. Miscommunication among the tester and the client would be a major impact could lead to a simple vulnerability scan much more intrusive. Ask several questions about the client's business and the status of the criticality of applications, downtime, online vendors, etc. Discuss the reliable method of sharing confidential credentials among the testers. Scope What IP addresses or hosts are in scope and what are not? Limitations of the actions should be taken by the tester or acceptable downtime any acceptable.

Vulnerability assessment could be part of penetration testing or pen-testing. Pen testing is a collection of the process which is done sequentially. Vulnerability assessment is a process of scanning the loophole of the system or application. Loopholes could be the weakness of the backdoor attacker to attack the victim. Several vulnerabilities may exist in the systems.

- Access control vulnerability
- Boundary condition vulnerability
- Input validation vulnerability
- Authentication Vulnerabilities
- Configuration Weakness Vulnerabilities
- Exception Handling Vulnerabilities

Penetration testing scope and the procedure would be the same somewhat attackers are being engaged while exploiting the system. However, the attempted procedure could be the preplanning authorized manner instead of irrespective harmfulness agenda. In penetration testing, the tester has authority to do penetration testing and he intently exploits the system and finds out possible exploits. On penetrating testing is not only discovering the vulnerabilities, but it could also be assessing what attackers might gain after successful exploitation. The pen testers' scope could vary from one client to another. Some of them are having excellent security posters while others could allow them to breach through vulnerabilities and finally gain access to them.

There are two penetration testing categories while considering the access perimeter known as internal and external. Penetration testing could overlook a malicious employee or attacker already breached into the system and the externally attempting to the intruder who is coming from the Internet.

2.5 Vulnerability Assessment And Penetration Testing Tool

Penn testing and vulnerability tools are very important elements to identify the loopholes and weaknesses of the systems. There is a wide range of tools that are available on the Internet to assist penetration testing and security assessment, particularly free tools, which are not being recommended to use due to the many malware attached with them. Parameters of the tools which are most appropriate for testing purposes could be popular among the users, the accuracy of the result, and quantifiable measures.

Kali Linux is a Debian-based Linux distribution system that comes with many powerful pre-installed security tools. Copy of Kali Linux install in a VMware virtual machine is a highly recommended tool for penetration testing. The characteristics of the kernel of the Kali Linux are very unique and more receptacle for security breaches as primary features.

Testing Tools & Common Vulnerabilities

TESTING TOOLS	COMMON VULNERABILITIES
Acunetix Web Vulnerability Scanner	SQL injection flows
Burp Suite Pro	cross site scripting – XSS /
Core Impact	broken authentication
IBM AppScan	insecure direct object references
NTOSpider	cross site request forgery – CSRF
Paros Free	security misconfiguration
MileScan ParosPro Desktop	insecure cryptographic storage
Qualys Web Application Vulnerability Scan	sensitive data exposure
Skipfish	failure to restrict URL access
w3af	missing function level access control
ZAProxy	using component with known vulnerabilities
	LDAP Injection

Table 2.1: Testing Tools and Common Vulnerabilities

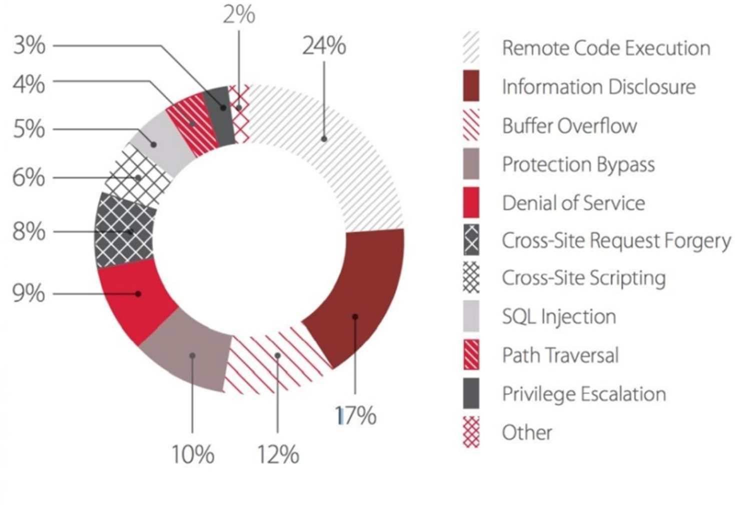


Figure 2.3 Type of Vulnerabilities according to Industrial Control System

Figure 2.3 depicted the ratio of attack vectors that can be potential to exploitation shown as a percentage. Remote code execution vulnerability is shown 24 percent of dominated among the others. Information disclosure and Buffer overflow are the other two vulnerabilities frequently at higher rates. Testing criteria could be intended to the values of figure 2.3 and the huge impact of the results.

Chapter 3: Bug Bounty

3.1 Introduction to Bug Bounty

It would be a controversial phenomenon that bug bounty is a great option to eliminate the vulnerabilities of the sophisticated applications being developed by several developers to fulfill their industrial requirements. Furthermore, third-party applications are available as standard web applications merged with many malware programs which are harmful to the systems. Then, selecting the testing tools are shows a major impact on the success of the testing process.
There are several platforms used for vulnerability disclosure assessment such as Open Bug Bounty (OBB) platform and the older volunteer-driven XSSPosed platform. Most community-based bug bounty platform is OBB. However, both methods are important to the theoretical framing of the bug bounties.

For security aspects, the Bug Bounty process has been drastically increased over the last couple of years. To eliminate the security breaches of the company applications, it is a must continuous implementation of the bug bounty process of the system. It shows the importance of identifying the existing vulnerabilities to prevent exploitations of the threats into the system and control the likelihood of the risk. Empirically thousands of vulnerabilities being disclosed of many applications continuously are caring with bug bounty process of many enterprises worldwide.

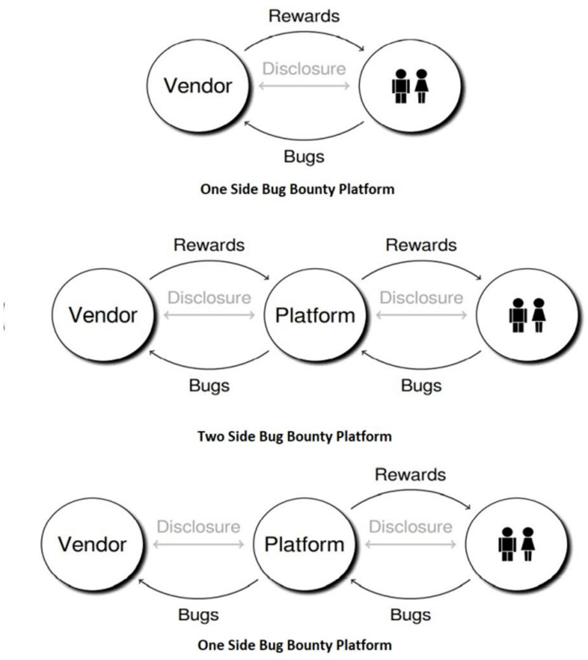
Bug bounty process is a popular technique for finding exploitation of vulnerabilities. It might be controversial phenomenon mitigation the vulnerabilities of the sophisticated applications while deployed in the initial stage by the developers. Furthermore, third-party applications are merged with many malware programs that are harmed for your genuine applications. Then, selecting the testing tools could be a major impact and depend on the progress of the test proceeding for the entire process successfully.

A bug bounty is an empirical approach rather than theoretical assumptions. However, the success of the mitigation techniques depends on several factors mainly such as tools, knowledge and previous experience, etc. Undoughtly, bug bounty is one of the most popular vulnerability detecting techniques used for individual rewards. It might be a controversial phenomenon that how much bug bounties are contributed to be protected the security systems in the enterprise applications.

Individual performance of the bug bounty is the most popular method rather than a group. Emprical implimentation is a best practices of the bug bounty among the testers. As gained from the previous experience with successful results might be moved on to further attempt as a framework. However, still, bug bounty is might have controversial factors to be stabilized in the business perspective expectation as enterprise context. Web vulnerabilities are given huge impact in the marketplace apparently by challenging the bug hunters' perspectives. Several individuals who are gained expert knowledge are being engaged in a successful attempt at rectification of the web vulnerabilities for their compensation or rewards.

Currently, there are two main existing segments of the bug bounty programs. As illustrated in figure 3.1 one side platform is the most shredded self-compensation approach whereas enterprise-level bug bounty practices are further subdivided as two side bub bounty and one side bug bounty platforms. However, reality would be identified the direct practices of the bug bounty appeared in the real world after 2010. Earlier initiatives were not gained widespread attraction in the software industry. Two side platforms are the modified version of the business models which are widespread among the developer at present. Later it was a definite initiative of several social media platforms along with Face Book, Google, and Microsoft etc.

Bug Bounty process has been drastically increased the practices among the ethical hackers for security aspect since last couple of years. Currently, it is initial testing requirement to protect the systems carried out the bug bounty process frequently as prevention mechanism of the security breach of the existing applications. Identification of the existing uncertainty circumstances of the system is a prevention mechanism of zero days' vulnerabilities or threats to control the probability of the risk in an important assert in the company. Empirically thousands of vulnerabilities being disclosed of many applications continuously are caring with bug bounty process of many enterprises worldwide. In here are emphasized appropriate pathways to correct provision to theoretical approach of finding vulnerabilities of the software applications.



One Side Bug Bounty Platform

Two Side Bug Bounty Platform

One Side Bug Bounty Platform

Figure 3.1 Bug Bounty Variants

3.2 Bug Bounty Process

This review mainly emphasized the theoretical points where necessary to carry on the practical scenario off bug bounty process and contribution elaborate in bullets as follows:

- Bug bounty platform align with theories and procedures along with appropriate network and the platform economy
- In terms of the number of vulnerabilities gained from the bug bounty not for consideration while allocation the compensation. The quality of the factor of the main parameter could necessary.
- The Discovery of automated tools in the bug bounty context precisely influences website targeted.
- Automated tools might be the solution to overcome the knowledge gap between productivity and the participants.
- The vulnerability gained is not mainly consideration even though large in size whereas the impact of the system disseminated is coordinated and communicated to vendors.
- However low-impact patching of vulnerabilities is often disseminated through bug bounties.

- Well, preparation before attempting would be an important factor affecting the patching time.

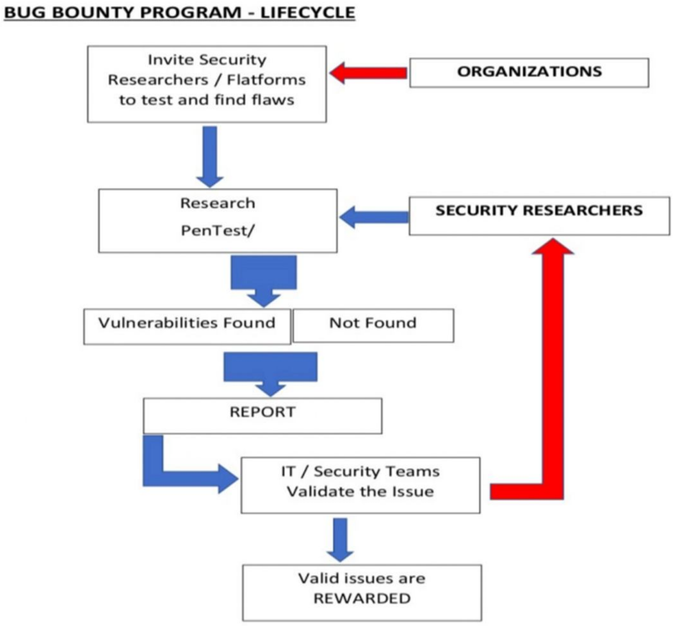

BUG BOUNTY PROGRAM - LIFECYCLE
Invite Security Researchers / Flatforms to test and find flaws
ORGANIZATIONS
Research PenTest/
SECURITY RESEARCHERS
Vulnerabilities Found
Not Found
REPORT
IT / Security Teams Validate the Issue
Valid issues are REWARDED

Figure 3.2 Bug Bounty Lifecycle

3.3 HackerOne

HackerOne is the proactive legal approach of hacking rather than reactive methodologies for high rewards instead of the criticality of the circumstances. The security assessments are exposed to explicit their product for rectifications of the existing vulnerabilities by the vendors. Hackers might have very high rewards of the criticality of the vulnerability that could be exploited. The bug bounty towards common platform is demarcated between vendor and the hackers facilitating to exploits the vulnerabilities such as Google and the Mozilla.

3.4 Rewards

Bug bounty rewards. Momentary rewards are popular discloser whereas outline payments are gone up to even several thousand dollars however it might not be raised proper acknowledgment on vulnerabilities rather than exploitation demonstration. The non-momentary approach could not be directly dependent on the cash however subsequently emphasized several related acknowledgments such as the security hall of fame in the bug bounty program. There are several benefits gaining the participants in the non-momentary approach such as collaborating and learning. The process of cost calculation could be challenged either any rewards due to the several instability circumstances happening along with the size of the software, the scope of the vulnerability rectification, and harmfulness of the criticality of the system damage.

3.5 Kali Linux for bug bounty

The reward is a concept of a better understanding of the bug bounty process. Monetary and non-monetary are the two methods of accepting Linux kernel as a navigator of the process handled by the processor. Linux kernel is system software that is rapidly upgrading and implementing the Linux kernel is a navigator of the process handled by the processor. Linux kernel is system software that is rapidly upgrading and implementing the most popular source code among the developers. Due to the high demand, many intruders are interested to breach the kernel to dominate their activities through loopholes that are existed. Security of the kernel is a crucial part of the open-source Linux system at present. The dynamic approach would be more convenient than the static method to find out the vulnerabilities and the security breaches. Expectation becomes a reality of the intruders if they could gain overwhelming privileges from the source being exploited. Some of the vulnerabilities are well known such as buffer overflows in which user-level application programs. Linux kernel (http://www.kernel.org/) is undergoing huge development annually as statistic shows that in every year such as approximately one-eight of the kernel code.

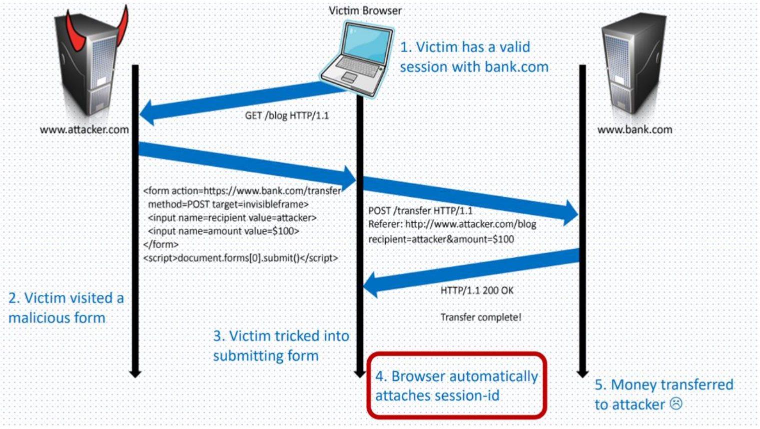
Due to the rapid implementation of the source code of the Linux kernel that could be complicated to handle in a standard systematic approach which is expected as the regular framework. Source code replication could be a major potential approach that moving from one to many destinations same existing vulnerabilities. The complexity of the code development is a huge impact on the compliance aspect of security-related implementation indeed. Dynamic bug finding methods are more practically potential rather than the manual approach. On the other hand, Linux Security Model (LSM) is a systematic approach to the access control mechanism where thorough attention on the security hardening to protect the compromised Linux Kernel. LMS has further emphasized the impact and performance of the security models in a Linux kernel.

Chapter: 4. Attack

The attack defines as unauthorized security breaches unintentionally intended to valuation aspects of confidentiality, integrity, and availability. Several vulnerabilities in a system are databases have given a huge impact on the business process. Nevertheless, vulnerabilities or bugs are common in the majority of systems even though millions of users could be integrated with them. To get some experience with popular vulnerable exploited applications click on the link to reach into the source code: https://www.exploit-db.com/exploits/35230

4.1 Brute Force Attack

Brute force is one of the most prevalent attacks that can be threatened due to the cryptographic vulnerabilities in the computers connected to the network. The machine learning evaluation process emphasized the attack vectors precisely on brute force attacks such as SSH protocol in the network. Based on the collected real-world source from the network recently to be applied the machine learning algorithm. Effectiveness of machine learning highly impact the comprehensive study indeed detecting brute force attack inflow level of the SSH protocol. The empirical approach is a highly successful detection method of the brute force attack that the machine learners with high percentages comparatively. The data transferring over the public network is a huge challenge due to the malicious activities intended to function. Encryption mechanisms are ensured that threat by intruder activates over the network whereas intended to the proper cryptographic algorithm which was designed proper standard and policies. Weak passwords with simple character combinations could be huge common vulnerability initiatives to intend into the brute force attacks.


Victim Browser
1. Victim has a valid session with bank.com
www.attacker.com
GET /blog HTTP/1.1
www.bank.com
2. Victim visited a malicious form
3. Victim tricked into submitting form
4. Browser automatically attaches session-id
5. Money transferred to attacker ☹

```
<form action=https://www.bank.com/transfer
  method=POST target=invisibleframe>
  <input name=recipient value=attacker>
  <input name=amount value=$100>
</form>
<script>document.forms[0].submit()</script>
```

```
POST /transfer HTTP/1.1
Referer: http://www.attacker.com/blog
recipient=attacker&amount=$100
```

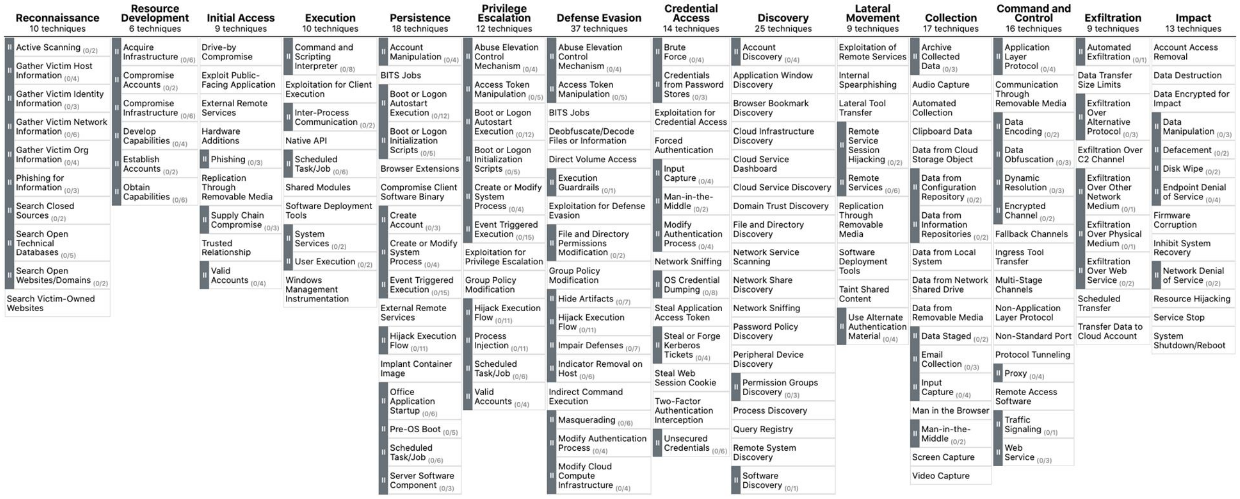
```
HTTP/1.1 200 OK

Transfer complete!
```

Figure 4.1 CSRF attack

Cross-site request forgery (CSRF) attack is very frequent happening attack who intended to compromise since individual online money transaction from the bank. Force authenticated user to send a request along with web application by the attacker which is currently authenticated. The figure 6.1 emphasized the steps how compromised and finally dominated the attacker.

4.2 MITRE ATT & CK framework

MITRE AT & CK framework is a model of knowledge based on cyber diversity behavior and the adversaries' attack variations with targeted vectors. It has now existed with three iterations such as ATT & CK for Enterprise, ATT & CK for Mobile, and Pre-ATT & CK. For more content further on MITRE ATT & CK with several live specimens that are available on the link:https://www.mcafee.com/enterprise/en-us/security-awareness/cybersecurity/what-is-mitre-attack-framework.html



Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
10 techniques	6 techniques	9 techniques	10 techniques	18 techniques	12 techniques	37 techniques	14 techniques	25 techniques	9 techniques	17 techniques	16 techniques	9 techniques	13 techniques
Active Scanning (0/2)	Acquire Infrastructure (0/6)	Drive-by Compromise	Command and Scripting Interpreter (0/8)	Account Manipulation (0/4)	Abuse Elevation Control Mechanism (0/4)	Abuse Elevation Control Mechanism (0/4)	Brute Force (0/4)	Account Discovery (0/4)	Exploitation of Remote Services	Archive Collected Data (0/3)	Application Layer Protocol (0/4)	Automated Exfiltration (0/1)	Account Access Removal
Gather Victim Host Information (0/4)	Compromise Accounts (0/2)	Exploit Public-Facing Application	Exploitation for Client Execution	BITS Jobs	Access Token Manipulation (0/5)	Access Token Manipulation (0/5)	Credentials from Password Stores (0/3)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (0/3)	Compromise Infrastructure (0/6)	External Remote Services	Inter-Process Communication (0/2)	Boot or Logon Autostart Execution (0/12)	Boot or Logon Autostart Execution (0/12)	BITS Jobs	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Data Encoding (0/2)	Exfiltration Over Alternative Protocol (0/3)	Data Encrypted for Impact
Gather Victim Network Information (0/6)	Develop Capabilities (0/4)	Hardware Additions	Native API	Boot or Logon Initialization Scripts (0/5)	Boot or Logon Initialization Scripts (0/5)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Infrastructure Discovery	Remote Service Session Hijacking (0/2)	Clipboard Data	Data Obfuscation (0/3)	Exfiltration Over C2 Channel	Data Manipulation (0/3)
Gather Victim Org Information (0/4)	Establish Accounts (0/2)	Phishing (0/3)	Scheduled Task/Job (0/6)	Browser Extensions	Create or Modify System Process (0/4)	Direct Volume Access	Input Capture (0/4)	Cloud Service Dashboard	Remote Services (0/6)	Data from Cloud Storage Object	Dynamic Resolution (0/3)	Exfiltration Over Other Network Medium (0/1)	Defacement (0/2)
Phishing for Information (0/3)	Obtain Capabilities (0/6)	Replication Through Removable Media	Shared Modules	Compromise Client Software Binary	Event Triggered Execution (0/15)	Execution Guardrails (0/1)	Man-in-the-Middle (0/2)	Cloud Service Discovery	Replication Through Removable Media	Data from Configuration Repository (0/2)	Encrypted Channel (0/2)	Exfiltration Over Physical Medium (0/1)	Disk Wipe (0/2)
Search Closed Sources (0/2)		Supply Chain Compromise (0/3)	Software Deployment Tools	Create Account (0/3)	Exploitation for Privilege Escalation	Exploitation for Defense Evasion	Modify Authentication Process (0/4)	Domain Trust Discovery	Software Deployment Tools	Data from Information Repositories (0/2)	Fallback Channels	Exfiltration Over Web Service (0/2)	Endpoint Denial of Service (0/4)
Search Open Technical Databases (0/5)		Trusted Relationship	System Services (0/2)	Create or Modify System Process (0/4)	Group Policy Modification	File and Directory Permissions Modification (0/2)	Network Sniffing	File and Directory Discovery	Taint Shared Content	Data from Local System	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Search Open Websites/Domains (0/2)		Valid Accounts (0/4)	User Execution (0/2)	Event Triggered Execution (0/15)	Hijack Execution Flow (0/11)	Group Policy Modification	OS Credential Dumping (0/8)	Network Service Scanning	Use Alternate Authentication Material (0/4)	Data from Network Shared Drive	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
Search Victim-Owned Websites			Windows Management Instrumentation	External Remote Services	Process Injection (0/11)	Hide Artifacts (0/7)	Steal Application Access Token	Network Share Discovery		Data from Removable Media	Non-Application Layer Protocol		Network Denial of Service (0/2)
				Hijack Execution Flow (0/11)	Scheduled Task/Job (0/6)	Hijack Execution Flow (0/11)	Steal or Forge Kerberos Tickets (0/4)	Network Sniffing		Data Staged (0/2)	Non-Standard Port		Resource Hijacking
				Implant Container Image	Valid Accounts (0/4)	Impair Defenses (0/7)	Steal Web Session Cookie	Password Policy Discovery		Email Collection (0/3)	Protocol Tunneling		Service Stop
				Office Application Startup (0/6)		Indicator Removal on Host (0/6)	Two-Factor Authentication Interception	Peripheral Device Discovery		Input Capture (0/4)	Proxy (0/4)		System Shutdown/Reboot
				Pre-OS Boot (0/5)		Indirect Command Execution	Unsecured Credentials (0/6)	Permission Groups Discovery (0/3)		Man in the Browser	Remote Access Software		
				Scheduled Task/Job (0/6)		Masquerading (0/6)		Process Discovery		Man-in-the-Middle (0/2)	Traffic Signaling (0/1)		
				Server Software Component (0/3)		Modify Authentication Process (0/4)		Query Registry		Screen Capture	Web Service (0/3)		
						Modify Cloud Compute Infrastructure (0/4)		Remote System Discovery		Video Capture			
								Software Discovery (0/1)					

Table 4.1 MITRE AT& CK attack for enterprise

4.3 Attack Navigator Tools on YouTube presentations

https://github.com/mitre-attack/attack-navigator

YouTube on MITRE AT& CK attack Nevigator

https://www.youtube.com/watch?v=pcclNdwG8Vs&t=300s

4.4 APT attack techniques

Advanced Persistence Attack (APT) is used best practices in new sophisticated attack techniques to gain access into the system whereas remaining prolong and a potentially long period for destructive activities valuable sources. Some APT attacks funded by government agencies are used as cyber warfare weapons. It might be a new trend of many conflicts between international communities to show their powers to the world. APT attack progression is broken into three main steps as infiltration, expansion, and extraction as emphasized in more details figure 4.2 and furthermore to click on the link: https://www.imperva.com/learn/application-security/apt-advanced-persistent-threat/

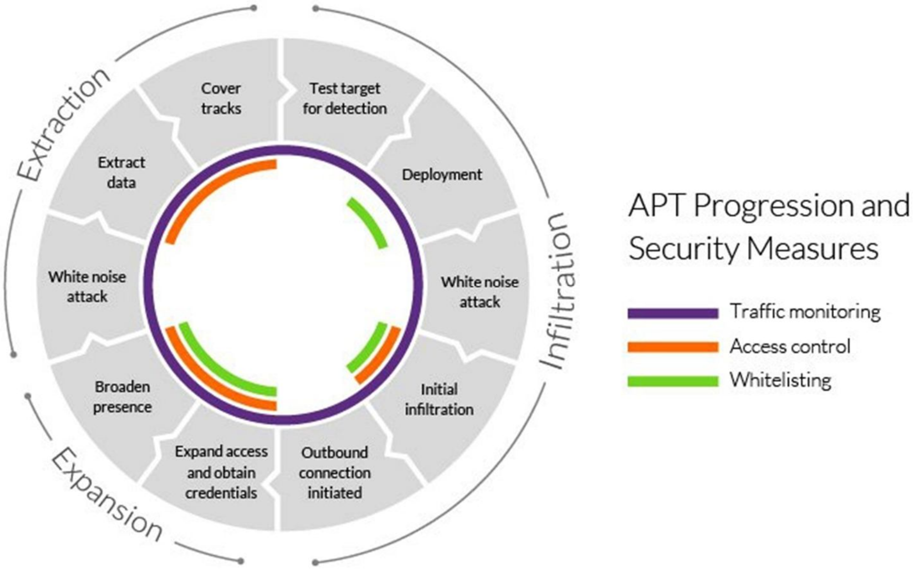


Figure 4.2 ATP attack progression

To become a security expert along with theoretical and practical competency, OSCP might be the suitable pathway to achieve success. Click the link for more details: https://www.offensive-security.com/ Offensive Security has been dedicated to penetration testing for the last couple of years with colorful results.

4.5 SonarQube

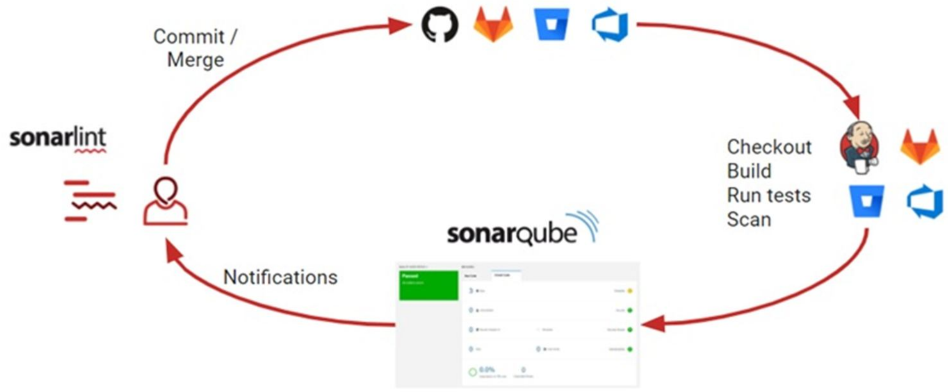
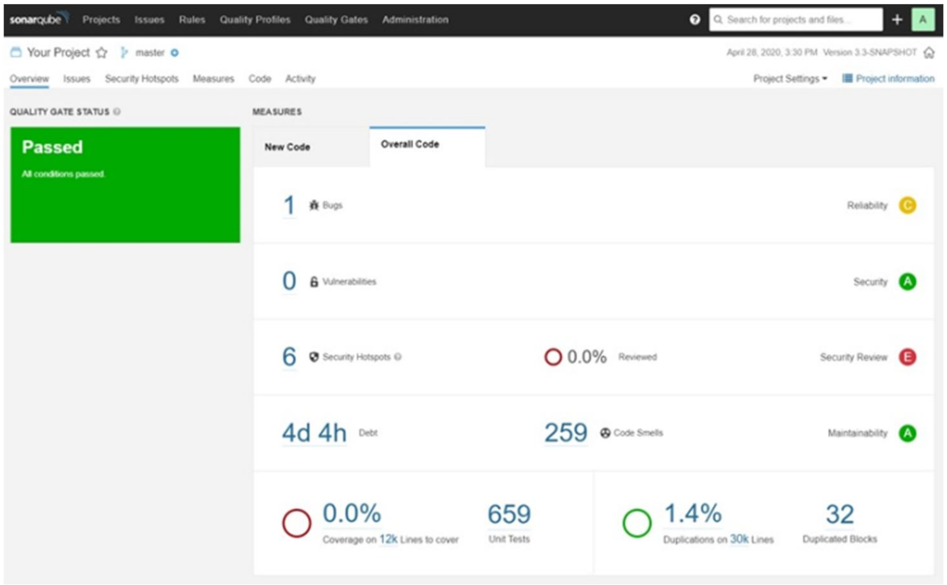
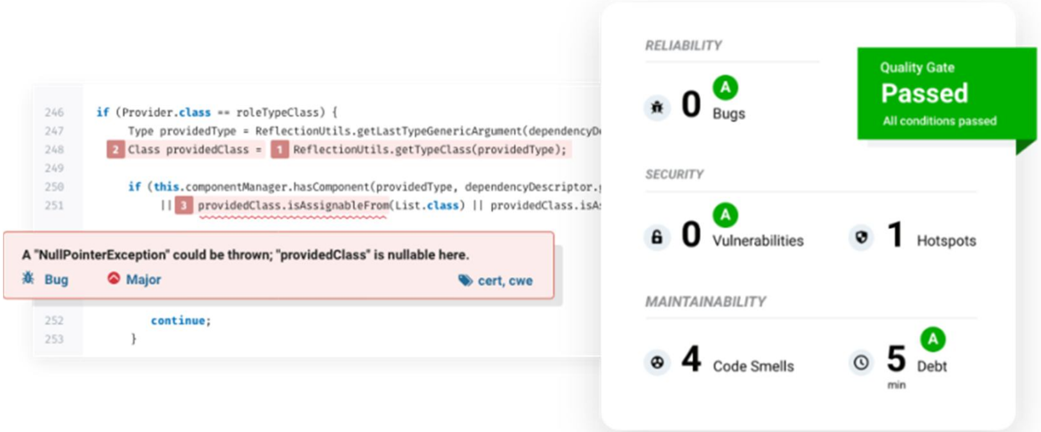


Figure 4.3 SonarQube cycle

SonarQube is an automated tool which is detected errors, misconfigurations, and vulnerabilities in the source code. Figure 4.4 has emphasized the first analysis report of the code. SonarQube is a well-recommended penetration testing tool in the marketplace. To reach more on SonarQube click the link: https://www.sonarqube.org/



Picture 4.4 SonarQube analysis of the code



```
if (Provider.class == roleTypeClass) {
    Type providedType = ReflectionUtils.getLastTypeGenericArgument(dependencyDe
  Class providedClass = ReflectionUtils.getTypeClass(providedType);

    if (this.componentManager.hasComponent(providedType, dependencyDescriptor.
        || providedClass.isAssignableFrom(List.class) || providedClass.isA
```

A "NullPointerException" could be thrown; "providedClass" is nullable here.

```
        continue;
    }
```

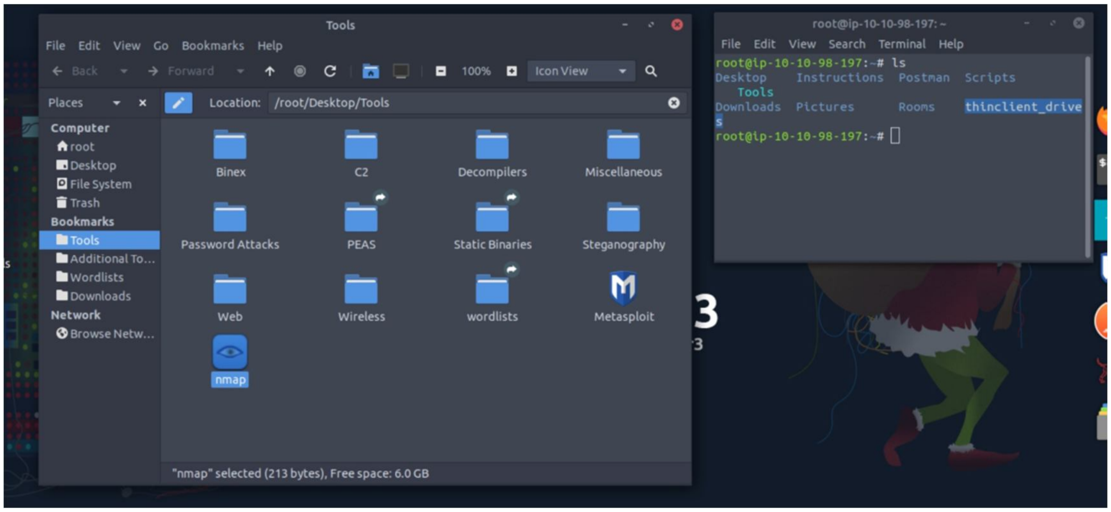
Picture 4.5: Error mechanism of SonarQube

Chapter 5: Hacker Boxes

The hacker boxes are virtual platforms that act as a facilitator including with necessary tool kits to hack the intended system. The hackers must show the capabilities of exploitation to the systems with their expertise theoretical and empirical knowledge to find the vulnerable vectors to breach. There are main three analyzing tool boxes are popular as white box security analysis (SAST), dynamic black box security analysis (DAST), and interactive white box security analysis (IAST)

5.1 Ethical Hacker

A hacker is a skilled person who has capable of handling tools to exploit the vulnerabilities in the systems. Ethical hackers or else known as white hat hackers are finding the report on system vulnerabilities from bug bounty programs in today’s security ecosystem. The role of an ethical hacker was dedicated to digging the targeted vectors to find the vulnerabilities to reduce the risk of the company's business continuity plan. Bug bounty programs are fully organized with rewards based on benefits to that individual who ware participated. Several other non-ethical hackers could be found who are damaged or engaging with harmful activities in the organizational systems. Some of them are categorized as a cracker, grey hat hacker, scrip kiddies, hacktivist, and phreaker, etc.



```
root@ip-10-10-98-197:~# ls
Desktop    Instructions  Postman  Scripts
   Tools
Downloads  Pictures      Rooms    thinclient_drive
s
root@ip-10-10-98-197:~# 
```

Figure 5.1 Common tools of hack box

The common tools generally exist in all the hacker boxes as shown in figure 4.1. The majority of hacker boxes that are used for industrial purposes could be quite expensive. The empirical approach of the attack more benefited the context rather than the theoretical way of applying hacking techniques. The hacking framework of the hacker boxes could be equal format every one of them except the number of tools and their accuracy. Tools are the most crucial part of the box to gain maximum benefits from the hackers. Some of Hacker Boxes will be emphasized herewith to show how it works**.**

5.2 Hacking Tools

Some of the popular and crucial hacking tools are Nessus, Acunetix, Netsparker, Intruder, Nmap, Metasploit, Aircrack-Ng, Wireshark, Ettercap, Maltego, Nikto, Burp Suite, John The Ripper, Angry IP Scanner

The hacking process is performed to identify potential threats in the computer or network then it will be ethical hacking. Ethical hacking is called different terminologies in terms of functions red teaming, penetration testing, and intrusion testing. Several online free courses are available to commence the bug hunting from the initial stage with appropriate practicals. Here is one of the most popular testing tool kits for beginners: https://tryhackme.com/dashboard

Well knows bug bounty commercial platforms are popular at present in the communities of cyber security in the world. Most of them were interested to know prominent web vulnerabilities and attacks exploited for their customers. Hackenproof is well known bug bounty team find from the link https://hackenproof.com/hacken/hackenproof

5.2.1 Nessus

Nessus is the world’s most widely-deployed vulnerability assessment solution. Nessus quickly and accurately identifies vulnerabilities, configuration issues and malware in physical, virtual and cloud environments to help you prioritize what to fix first. Combine Nessus with Kali Linux to build a superior pen testing toolkit that provides deep insight into your network systems.

Nessus Installation Steps

Step 1: Purchase Nessus and obtain an Activation Code. In here I got the testing key to activate the package

Step 2: Download Nessus Click the link here:

https://www.tenable.com/downloads/nessus?loginAttempted=true

Select the package: Debian 6, 7, 8, 9 / Kali Linux 1, 2017.3 AMD64 Download the package and Open the download folder

Step 3: Install Nessus Run the package https://localhost:8834/ Create new panel password Enter the silence key step

Step 4: Configure and use Nessus

- apt update apt upgrade
- dpkg -i package name which is down loaded
- /etc/init.d/nessusd start
- update-rc.d nessusd enable
- https://localhost:8834

5.2.2 Netsparker

Netsparker is a one of the automated security scanner which is dedicated to find web vulnerabilities throughout the SDLC (System Development Life Cycle). Instead of the vulnerability scanning remediation will be precede reducing the processing time. The link for get more on this services herewith: https://www.netsparker.com/plp/scan-website-security-issues/?

5.3 Bug Report

When you are find a bug or vulnerability report must be prepared to disclose the findings. It contents details of where was found, how to reproduced it, proof of concept supporting details. These details are highly valuable to carry out the potential vulnerability assessment. Rectification process is faster than it could be to eliminate more replications and other harmfulness to the system. The report must be summarized as follows:

- Summery title
- Target
- Technical

Chapter 6: Penetration testing for Cryptographic Vulnerabilities

6.1 Common Cryptographic Vulnerabilities

Several cryptographic vulnerabilities in web applications are fallen into well-known categories whereas frequently identified since processing the security scanning phase of the development process. Good practices of the development and the operation of the lifecycle could be eliminated the well-known vulnerabilities. In aspect of the security vulnerabilities, different scope and practices would be used to find cryptographic issues in the web applications or system.

6.2 Finding Vulnerabilities from CryptoGuard

CriptoGurd is a static analyzing tool that can be used to detect cryptographic issues and vulnerabilities in the Java application and the Android mobile apps. Static analyzers are agile to use for developers and penetration testers to the identification of cryptographic misuse rather than the automated systems. CryptoGuard was the most effective and efficient static code analyzer for Java working precisely. The most frequent issues in cryptography that can be identified are insecure internet protocol and weak passwords. Some cryptographic misuse is frequently identified from the developers' end as insecure protocols, hard-coded keys, and weak program hashers. CryptoGuard has been used as a code scanner for cryptographic misuse precisely for since last decade. In this document was emphasized that several examples of evaluation identification of such vulnerabilities in the coding. Figure 6.2 realized the data flow of the CriptoGuard analyzing strategies that were implemented in the process sequentially. Indeed, several analyzing steps process the source code plus build a script to evaluate and detect the cryptographic vulnerabilities.

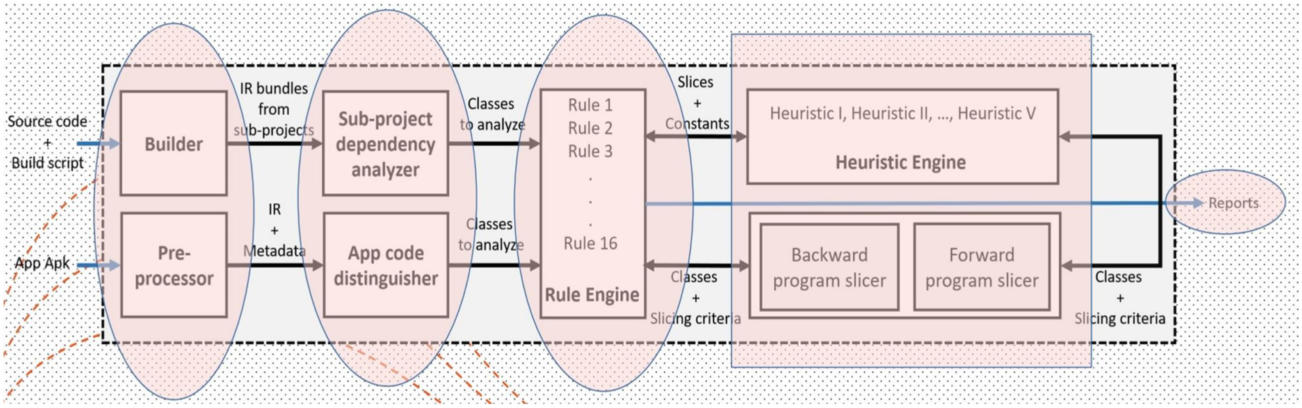


Figure 6.1 Cryptographic misuse detection from CriptoGusrd

Agility is the most prominent characteristic of the development process along with different strategies applied by the developers whereas increasing the likelihood of other unexpected circumstances happening particularly intended too much vulnerability due to the technique or strategies being used to speed up the process. Software complexity might be intended to be done huge misuse in coding whereas weaken secure architecture directly impact most frequent vulnerabilities. The status of the current landscape, Java application development projects are followed as good developers' ecosystems to implementation process to minimize as such complexity circumstances on software development process such as Intergraded Development Environment (IDE) supported developed sophisticated Java programs.

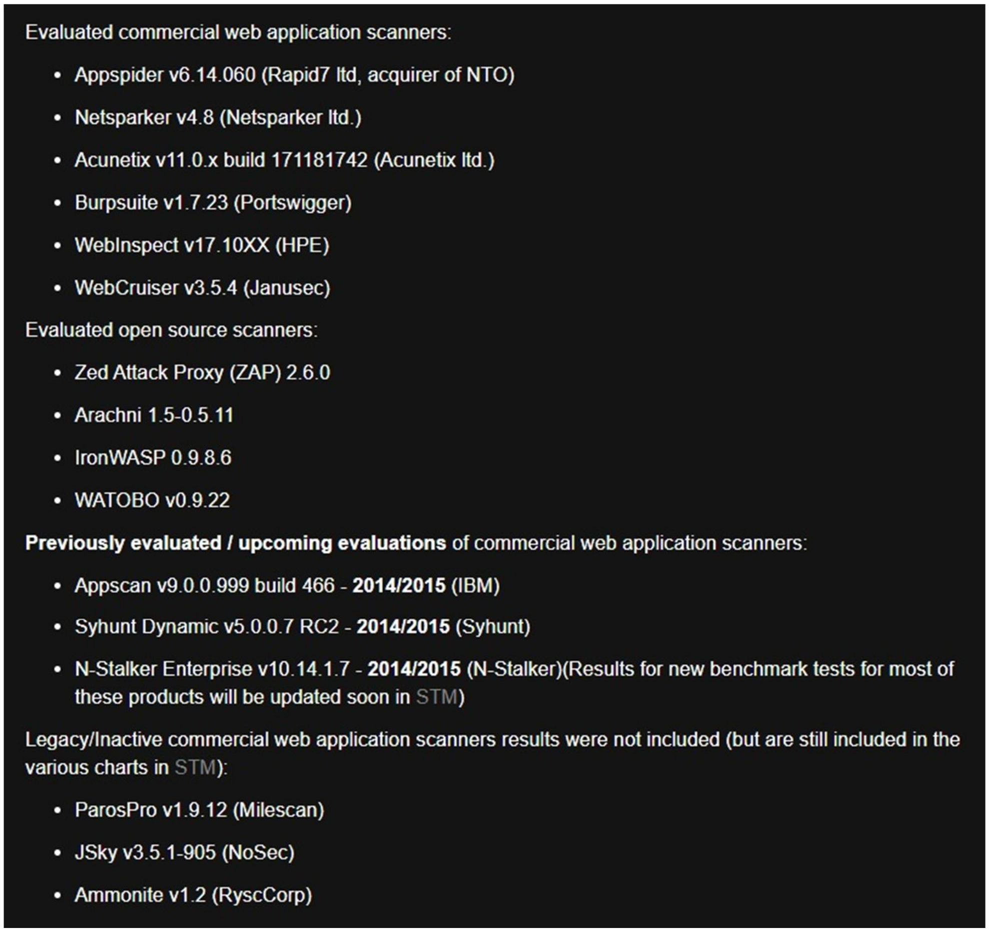
Since the completion of the CreptoGurd analysis process, it might be hosted into the cloud for ensuring with Security Assurance Marketplace (SWAMP). To ensure the exhausting and comprehensive security of the program which was developed by the Department of Homeland Security (DHS) is a SWAMP. Not only that, they create developers to test the source code since rectifying the issues and aspects of the security deploy the necessary testing tools. Many analysis tools depend on the open-source platform with truly possible to scan only Java codes. OWASP tools are prevalent platforms regardless of the language intended to best perform source code analysis. Static white box security analysis (SAST), dynamic black box security analysis (DAST), and interactive white box security analysis (IAST) security testing tools of OWAPS are designed based on proper technique and algorithm.

6.3 CrySL Tool

Finding cryptographic API misuses could be a critical point that screening millions of lines of code with many technical challenges. False positives due to phantom methods and false positives due to data structures are the main technical challenges of false positives. The security aspect of the API misuse on Java or Android programs basically should be designed by the cryptographic experts not be program developers. CrySL has sorted the cognitive gap between cryptographic experts and the program developers. CrySL is a demand-driven static analysis method that helps automatically checking a given Java or Android app for compliance with the CrySL-encoded rules. A set of extensive rules have been designed on CrySL for the JCA and evaluated. According to the facts, it shows that 95 percent of cryptographic API at least one misuse. It has been shown that cryptographic vulnerabilities of web applications are existed empirically to be further evaluated. Especially Java applications have been misused cryptographic libraries as frequently since the development process by the developers. There are extensive rules intended to enable CrySL in the JCA and empirically prove that by analyzing current Android apps.

Mistakes in cryptography would be the common phenomena frequently happening as a human being of the process of software implementation even though adapted the standard framework. The rectification is a must on every portion of the software before launching to the public especially the security aspect to eliminate the uncertain circumstances which could be faced by an indecent party. Statistic has been shown in the CEV database as 17% of the bugs are in cryptographic libraries and 83% are misused of cryptographic libraries. It clearly stated that traditionally many organizations are attempted to do mistakes intentionally or unintentionally in cryptographic or ciphertext in software applications.

Chapter 7: Evaluation of Modem Pen test



Evaluated commercial web application scanners:

- Appspider v6.14.060 (Rapid7 ltd, acquirer of NTO)
- Netsparker v4.8 (Netsparker ltd.)
- Acunetix v11.0.x build 171181742 (Acunetix ltd.)
- Burpsuite v1.7.23 (Portswigger)
- WebInspect v17.10XX (HPE)
- WebCruiser v3.5.4 (Janusec)

Evaluated open source scanners:

- Zed Attack Proxy (ZAP) 2.6.0
- Arachni 1.5-0.5.11
- IronWASP 0.9.8.6
- WATOBO v0.9.22

Previously evaluated / upcoming evaluations of commercial web application scanners:

- Appscan v9.0.0.999 build 466 - **2014/2015** (IBM)
- Syhunt Dynamic v5.0.0.7 RC2 - **2014/2015** (Syhunt)
- N-Stalker Enterprise v10.14.1.7 - **2014/2015** (N-Stalker)(Results for new benchmark tests for most of these products will be updated soon in STM)

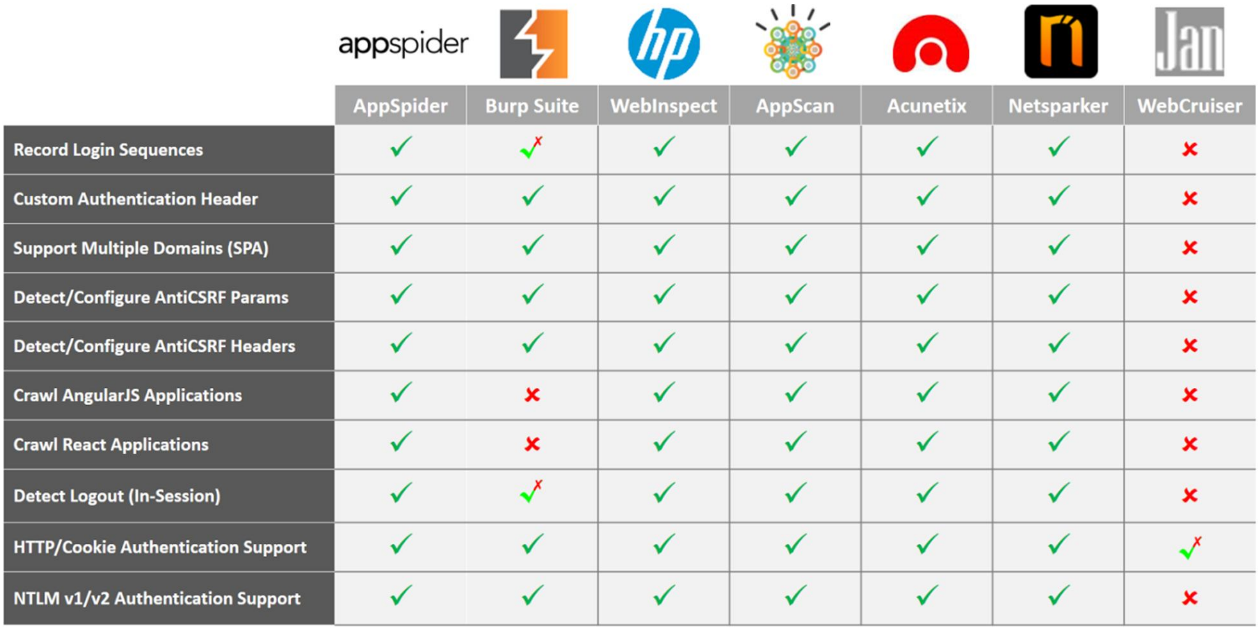
Legacy/Inactive commercial web application scanners results were not included (but are still included in the various charts in STM):

- ParosPro v1.9.12 (Milescan)
- JSky v3.5.1-905 (NoSec)
- Ammonite v1.2 (RyscCorp)

Figure 7.1: List of web application scanners

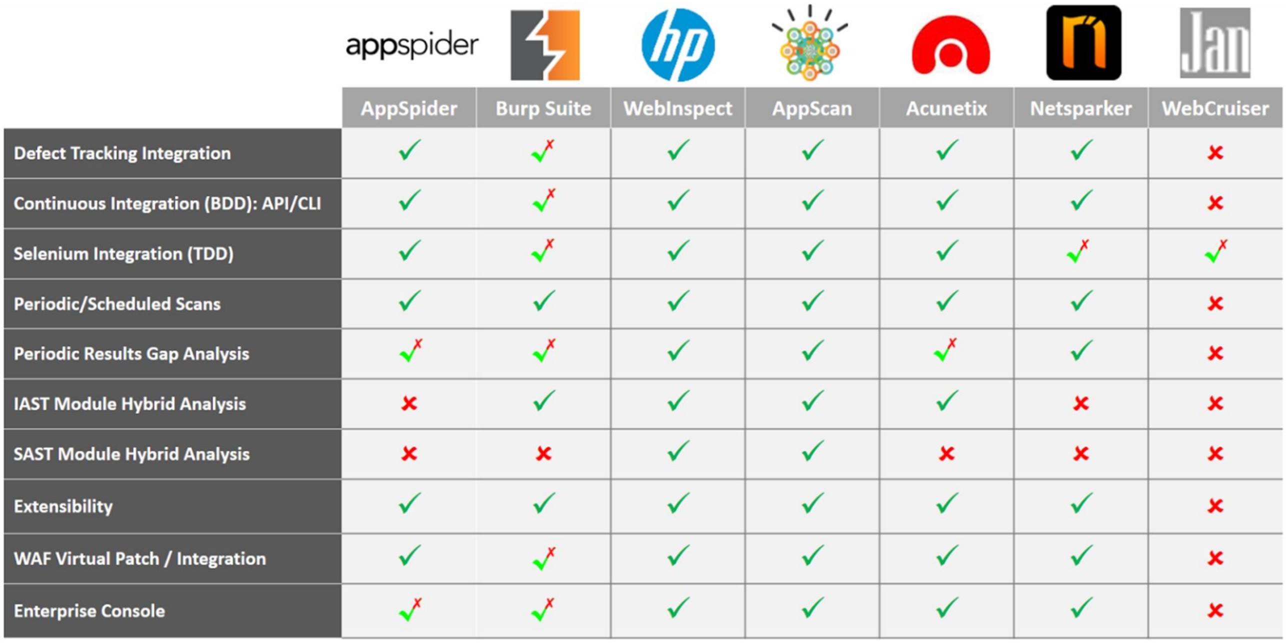
Evaluation of the web applications in SDLC is common at present. Several scanning tools such as figure 7.1 bring used by the developers however the report of the applications against tools represented as follows: for further details click the link: https://sectooladdict.blogspot.com/

The link shows that exactly voluble information for development clues aspect of the vulnerability scanning tools. Implementation of the DAST (Dynamic Application Security Testing), SAST (Static Application Security Testing) and IAST (Interactive Application Security Testing) solutions for SDLC (System Development Life Cycle) process will be taken many years particularly for finance, hi-tech and telecommunication companies.



	AppSpider	Burp Suite	WebInspect	AppScan	Acunetix	Netsparker	WebCruiser
Record Login Sequences	✓	✓✗	✓	✓	✓	✓	✗
Custom Authentication Header	✓	✓	✓	✓	✓	✓	✗
Support Multiple Domains (SPA)	✓	✓	✓	✓	✓	✓	✗
Detect/Configure AntiCSRF Params	✓	✓	✓	✓	✓	✓	✗
Detect/Configure AntiCSRF Headers	✓	✓	✓	✓	✓	✓	✗
Crawl AngularJS Applications	✓	✗	✓	✓	✓	✓	✗
Crawl React Applications	✓	✗	✓	✓	✓	✓	✗
Detect Logout (In-Session)	✓	✓✗	✓	✓	✓	✓	✗
HTTP/Cookie Authentication Support	✓	✓	✓	✓	✓	✓	✓✗
NTLM v1/v2 Authentication Support	✓	✓	✓	✓	✓	✓	✗

Picture 7.2 Scan Barrier Support Commercial



	AppSpider	Burp Suite	WebInspect	AppScan	Acunetix	Netsparker	WebCruiser
Defect Tracking Integration	✓	✓✗	✓	✓	✓	✓	✗
Continuous Integration (BDD): API/CLI	✓	✓✗	✓	✓	✓	✓	✗
Selenium Integration (TDD)	✓	✓✗	✓	✓	✓	✓✗	✓✗
Periodic/Scheduled Scans	✓	✓	✓	✓	✓	✓	✗
Periodic Results Gap Analysis	✓✗	✓✗	✓	✓	✓✗	✓	✗
IAST Module Hybrid Analysis	✗	✓	✓	✓	✓	✗	✗
SAST Module Hybrid Analysis	✗	✗	✓	✓	✗	✗	✗
Extensibility	✓	✓	✓	✓	✓	✓	✗
WAF Virtual Patch / Integration	✓	✓✗	✓	✓	✓	✓	✗
Enterprise Console	✓✗	✓✗	✓	✓	✓	✓	✗

Figure 7.3 SSDLC Feature Support Commercial

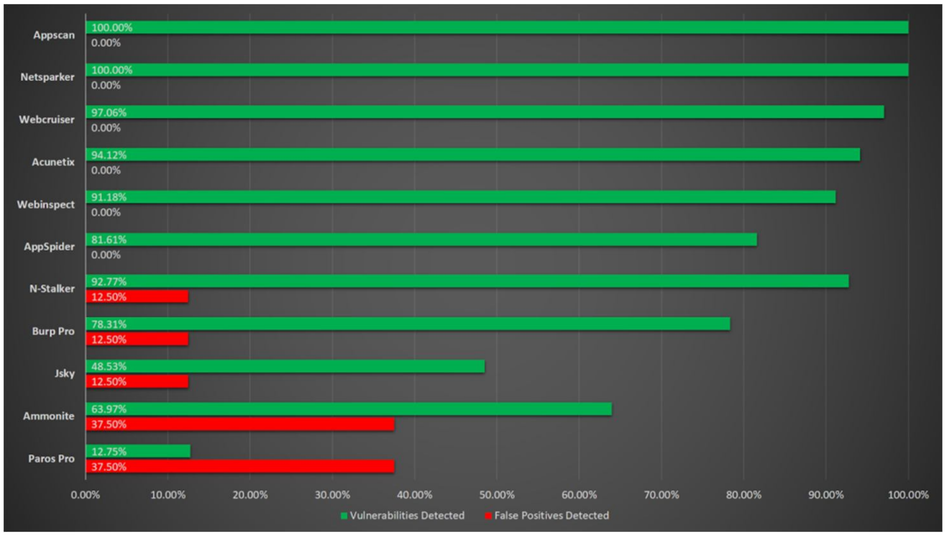


Table 7.1 Path Traversal Commercial

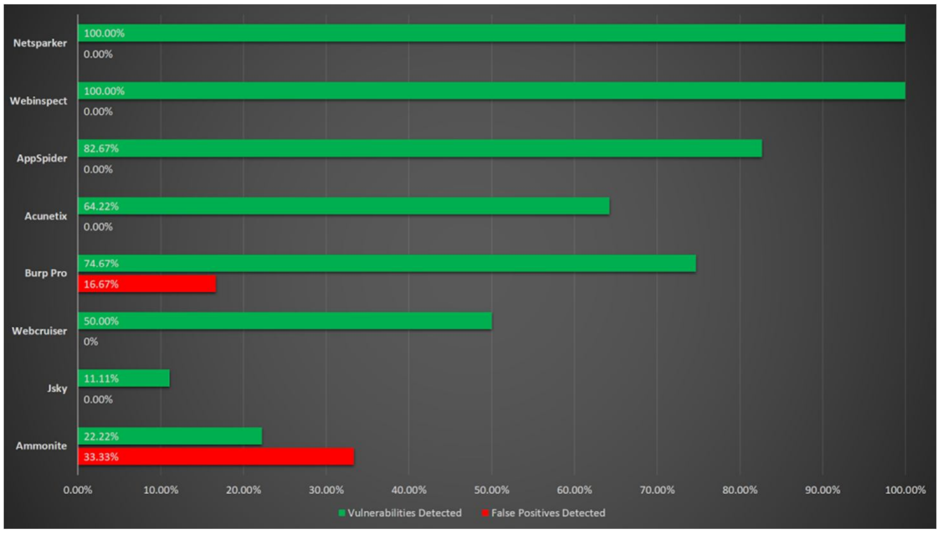


Table 7.2 RFI Commercial

Conclusion

Vulnerability assessment and penetration testing are critical factors in current applications and software systems regardless of the task and the flat form which is running. Penetration testing, bug bounty and vulnerability detection techniques are categorized as the same domain in terms of uncertainty weakness of the software program or related activities of the deployment of the application in the online environment. In contrast, Penetration testing was emphasized precisely far more advanced predesigned vulnerability detection mechanism rather than the bug bounty. The important characteristics of the penetration testing could be the criteria and the potential approach that can be applied to proceed is very formal and informative best practices comparatively well suited to enterprises. I would say that bug bounty is an informal attempt on the evaluation of the error detection mechanism for their personal rewards without formal procedure or documentation approach. Penetration testing is a group task along with sequential practices which was planned previously.

Testing tools are the most crucial elements of the penetration testing phenomenon whereas it will be depend on the expected output. Due to the limitation of the tools, they are very expensive and competitive. Thereby, overall penetration testing scenarios make huge financial dependencies to proceed. Bug bounty could be the individual non procedural disclosure attempt with mutual understanding for both ethical hacker and the organization. However, pub bounty is a process of task oriented non official open challenges. Empirical approach could be widely successful approach for both penetration testing and the bug bounty process.

Exploitation of the software applications for threats is a common several times a year in terms of vulnerabilities in year. Recommendation of the penetration testing has to be done at least once a year for online web applications.

Vulnerabilities can’t guess in particular application or cording thereby, it could be an unexpected security breaches during the live development. Therefor vulnerability assessment is important even though system running smoothly to avoid the zero day vulnerability. Pen testing is complicated process and expensive phenomenon which is not get rid of form the business when run the business continuity planning.

Penetration testing is multi-processing well plan sequence of testing mechanism with correct scheduled and doing with trusted team who are dedicated and skill full. There are certain agreement with company and the testers before attempt to carry the testing process such as business continuity plan and the disaster recovery procedures.

Kali Linux was well tested open source operating system inbuilt with security such as bug prevention and detection mechanisms. Kali is dedicated open source operating system for penetration testing support with many powerful tools such as Nessus. Several features of Nessus were emphasized with this document somewhat identification of the several machines that are connected in the same network. Finally, it shows that how to eliminate the existing antivirus programs while doing the penetration testing. This successful project, which is help to learn many of new concepts and it, helps how to approach of real pen testing attempt next time

Comparatively, cryptographic vulnerability detection methodology and the tools are quite different than others in terms of practices. CriptoGuard like tools are particularly well known to applied security engineers rather than the general bug bounty or pen testing teams. Therefore, the scope of the testing is a very important factor to emphasize and negotiate for with the company before commencing a disclosure agreement for further persisting this industry

Consistency of the testing process is a must to mitigate the security risk of the system specially some of threat such as zero day vulnerabilities. DevSecOps security practices are well suited precaution against the zero day vulnerability attacks. In contrast, pretty sure that there should be security professionals to established with a team for continuous alerted 24/7 to mitigate the security risk near future.

Appendix:

Most Prominent Password Recovery Software

https://hashcat.net/hashcat/

Global Cyber Security updates

https://www.csoonline.com/asean/

Data Exfiltration: How Data Gets Out

https://www.csoonline.com/article/2135266/data-exfiltration--how-data-gets-out.html?nsdr=true