Manual and Automated Penetration Testing. Benefits and Drawbacks. Modern Tendency

Yaroslav Stefinko, Andrian Piskozub, Roman Banakh

Abstract - **In this paper, we describe penetration testing, as a methodology for information security. Pentesting is used for proactive defence and information systems protection. Special operational systems on UNIX core, developed scripts, utilities and applications are suggested. Most pros and cons of manual and automated pentest are given.**

Keywords – **Penetration Testing, Information Security, Proactive Defence, Ethical Hacker, Exploitation, Hacking, Data Breach, Cyber Attack, Metasploit, Python.**

I. Introduction

Cyber attacks have become one of the greatest threats to the world of business and economics. Amount of damage has been growing every day and more companies or institutions have become victims of attacks or data breach performed by black hats. Hence, companies are looking for best way to protect their systems and essential information. Most popular way is to probe their systems via penetration tests by certified ethical teams, which can proactively defend computer systems.

Penetration test is a security evaluation process for network or computer systems that simulates an attack by an ethical hacker (pentester). The testing process usually involves simulating different types of attacks on the target system. This type of testing provides an organized and controlled way to identify security shortcomings. The resources and time required for comprehensive testing can make penetration testing cost intensive. Consequently, such tests are usually only performed during important milestones.

The most important distinction between a hacker and a penetration tester is that penetration test is done with a license and a signed contract with an organization or company, and the output is provided as a report. Such tests may be legal if the exfiltrated data was not personal, but this legality may be questionable depending on the legal jurisdiction where the test takes place. Some words about ethics we will discuss as well in next sections.

II. Pentesting Background

Pentesting is used to search for vulnerabilities that might exist in a system. The goal of penetration test is to increase data security. Security information and weaknesses that are specified in penetration test are considered confidential and shall not be disclosed until complete resolution of defects[2].

Penetration testers attack systems to evaluate their security in the face of realistic threats. These attacks take the form of authorised penetration tests that probe a system's defences; these defences are then breached to evaluate the impact of any weaknesses; the results of these tests are used to improve a system's security, making them resilient to further attacks.

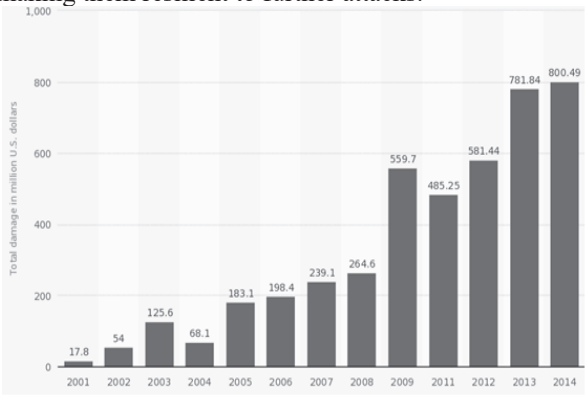

Fig.1. Amount of monetary damage caused by reported cyber crime to the FBI(ICS) from 2001 to 2014 (mln.US dollars).Source - statista.com.

The impact of these attacks can affect the company in many ways such as data loss, denial of service, liability claims, and loss of productivity. The serious nature of the impacts has resulted in the growth of an entire industry, which addresses these security issues [7]. Network firewalls and vulnerability scanners are two types of common security solutions available today. These solutions address specific concerns: a firewall setup will prevent an unauthorized access into the system, whereas vulnerability scanners will spot the potential vulnerabilities in the system. However, these tools cannot ensure the identification of all the different modes of attacks. Penetration testing is an important, additional tool for addressing the common security issues. A penetration test not only identifies the existing vulnerabilities, but also exploits them. The goal of penetration testing is to improve or augment the security posture of a network or a system [5]. Hacking a system requires technical prowess, creativity, and ingenuity to find unexpected ways of appropriating it [4]. Penetration testing requires all of this, with the added constraint that finding and exploiting vulnerabilities should neither harm the system nor encroach on the dignity of those affected by it.

Penetration testing is performed by compromising the servers, wireless networks, web applications, and other

Yaroslav Stefinko, Andrian Piskozub, Roman Banakh – Lviv Polytechnic National University, S.Bandery Str., 12, Lviv, 79013, UKRAINE, E-mail: jarik.bit@gmail.com, piskozub@polynet.lviv.ua, banakh.ri@gmail.com

potential points of exposure to identify and analyse security issues [1]. This enables fool proofing against these issues to prevent future attacks. Penetration testing can be conducted either manually or through an automated program.

There are numerous benefits of penetration testing from the business as well as technical perspective. Some of the principal reasons for adopting penetration testing are presented below:

Security Issues

Security issues such as malware attacks, network intrusion, and data theft can result in service interruption and unreliable system processes. This could lead to potential loss of customer loyalty and affect the company's market value. Penetration testing can work to avoid such occurrences by weeding out persistent as well as unexpected threats.

Protect Information

Companies use different security mechanisms to safeguard information like access control methods, firewalls, cryptography, intrusion detection systems, etc. [4]. However, with new attacks being discovered everyday it is difficult to protect user/system information constantly. Penetration testing could address these concerns by simulating a variety of attacks simultaneously.

Prioritize security risks

The use of penetration testing as a standard security practice not only helps understand the security issues but it can also help prioritize these issues. The issues identified during the testing can be prioritized on the basis of severity. Also, these efforts can lead to efficient budget allocation for information security issues.

Financial Loss

Penetration testing helps to mitigate the loss of revenues/capital due to service downtime arising from malicious attacks. It can also prevent or reduce fines/lawsuits resulting from security malpractices.

Traditionally, pentesting should be included as a standard process within the security testing roadmap. Organizations prefer to perform penetration testing prior to a product release or a major upgrade. However, it is also advisable to conduct this testing in the following situations:

- New infrastructure is added
- Software is installed
- System updates are applied
- Security patches are applied
- User policies are modified.

Rules of Engagement

Every penetration test you do would comprise of a rules of engagement, which basically defines how a penetration test would be laid out, what methodology would be used, the start and end dates, the milestones, the goals of the penetration test, the liabilities and responsibilities, etc. All of them have to be mutually agreed upon by both the customer and the representative before the penetration test is started. Following are important requirements that are present in almost every ROE:

- A proper "permission to hack" (ROE) and a "nondisclosure" (NDA) agreement should be signed by both the parties.
- The scope of the engagement and what part of the organization must be tested.
- The project duration including both the start and the end date.
- The methodology to be used for conducting a penetration test.
- The goals of a penetration test.
- The allowed and disallowed techniques, whether denial-of-service testing should be performed or not.
- The liabilities and responsibilities, which are decided ahead of time. As a penetration tester you might break into something that should not be accessible, causing a denial of service; also, you might access sensitive information such as credit cards. Therefore, the liabilities should be defined prior to the engagement.

If you need a more thorough documentation, refer to the "PTES Pre-engagement" white paper [6].

Most financial organizations would like to be compliant to world information security requirements, which are described in The Payment Card Industry Data Security Standardn(PCI DSS) and The Unified Compliance framework (UFC) standards. UFC contains the following general reference to penetration testing (Unified Compliance, 2010): "The organization will perform penetration testing on all defined major, general support, and key minor application systems at least yearly or after any material changes."

There is also specific industry guidance. For example, the Payment Card Industry Data Security Standard contains the following reference to penetration testing [7]: "Once the threats and vulnerabilities have been evaluated, design the testing to address the risks identified throughout the environment. The penetration test should be appropriate for the complexity and size of an organization. Another section of PCI standard references that the goal of the penetration test is to gain access (PCI, 2008): "The goal of penetration testing is to determine if unauthorized access to key systems and files can be achieved"[7].

From the methodology side, there are public testing methodologies that are available. Various organizations have published free frameworks that will facilitate successful in-house penetration testing:

- The Open Source Security Testing Methodology Manual (OSSTMM), currently version 2.2 and version 3 Lite are free;
- Open Web Application Security Project (OWASP) Testing Guide;
- NIST Special Publication 800-42: Guideline to Network Security Testing;

- PTES. The Penetration Testing Execution Standard.
- Information Systems Security Assessment Framework (ISSAF)

In the NIST penetration test methodology, the penetration test consists of four phases: *planning, discovery, attack, reporting*. Despite this, we can also use BackTrack pentesting methodology. It is predecessor and founder of **Kali Linux** – modern penetration testing operational system for ethical hacking.

The BackTrack testing methodology, presented in this section, comprises both the black-box and white-box approaches. Either of these approaches can be adjusted according to the given target of assessment. The methodology is contains stages that should be followed in order to accomplish a successful assessment. These include *Target Scoping, Information Gathering, Target Discovery, Enumerating Target, Vulnerability Mapping, Social Engineering, Target Exploitation, Privilege Escalation, Maintaining Access, and Documentation and Reporting*.

Whether applying any combination of these steps with black-box or white-box approaches, it is all left up to the penetration tester to decide and choose the most strategic path according to the given target environment and its prior knowledge before the test begins. We will explain each stage of testing with a brief description, definition and its possible applications[1].

The illustration for the BackTrack (Kali Linux) testing process is also given below:

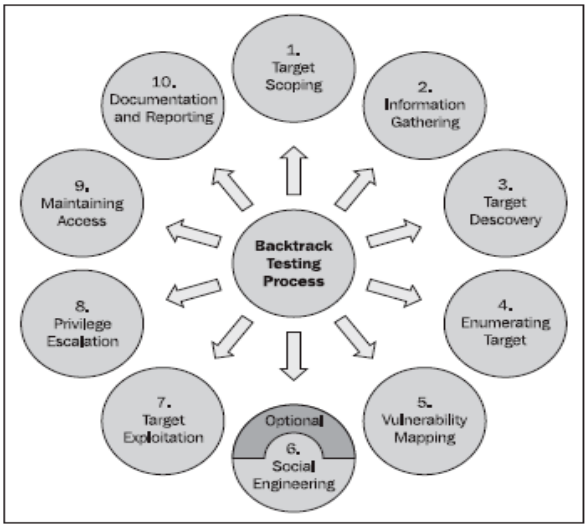

Fig.2 Penetration test process from Backtrack community.

III. Comparison of Manual and Automated Penetration Testing

Until recently, penetration testing has been a very complex *manual* process that could be performed by only advanced security specialists with many years of relevant experience. Testers typically write their own exploits, learn to master tools available in the public domain, and perform many tedious, time-consuming tasks. Comprehensive, manual penetration testing usually requires an extensive team of professionals possessing diverse skill sets, which most organizations cannot afford to maintain in-house or contract on a frequent basis.

Usually, for manual pentest, ethical hackers use **Metasploit Framework** command line utility - "msf" with adding some options or suitable exploits from Metasploit community database.

The all-inclusive nature of testing employed in penetration testing makes it a very complex process. Manual penetration test usually is not effective in terms of time and money, so its automatic version is considered. But manual pentest have benefits in long term efficiency, because of universality. Web scanners are used for performing the automatic web penetration test. For example, Nessus firstly crawl the target, then attack to the results of the previous phase and finally report vulnerabilities in the target.

A commercial-grade *automated* penetration testing solution is typically produced by a team of experienced security experts(tiger-team) and developers who complete sophisticated vulnerability research, build safe, cutting-edge exploits and then combine them into a simple, easy-to-use package. By thoroughly testing across networks, endpoints, web applications and email users, an automated penetration testing solution can provide a clear, comprehensive view of an organization's security posture.

For example, one of the most useful and user-friendly command line tool for exploitation in Kali Linux 2.0 is Social engineering toolkit(SET), is a menu driven tool in used to build different client-side tricks.

It includes various social engineering attack options that can be deployed from the same interface. It is written in Python and the menu-driven functionality makes it easier to build the attack. The social engineering toolkit helps execute a complex attack with less efforts and time and also allows us to test various social engineering scenarios in a practical way. It was previously impossible to execute these in a timely manner. The social engineering toolkit can be found in Kali Linux 2.0 at Applications | Exploitation tools. Once the terminal window is up, you will be presented with the menu shown in the following screenshot. The prompt at the terminal displays set and it waits for your input:

Automated testing is a safe and simple way to perform all the tasks related to penetration testing. Also, since most of the tasks are automated, the tests can be less time-consuming than manual testing. The ease of reproducibility of the tests is also a big benefit, compared to the customized approach in manual testing.

Fig.3. Social engineering toolkit for automated exploitation.

Scripting languages like Python, Perl, Ruby and even bash are becoming more popular for penetration testing and can be easily compatible with Metasploit or directly from Linux shell, for example, pentester can use Python script for brute-forcing credentials for ssh connection:

attacker# python sshBrute.py -H 10.10.1.36 -u root -F pass.txt

[-] Testing: 123456
[-] Testing: 12345
[-] Testing: 123456789
[-] Testing: password
[+] Password Found: alpine

For example, we can easily create malicious server. We will use the Metasploit framework in order to quickly create infected server and page hosted at http://10.10.10.112:8080/exploit. For this example we chose the exploit ms10_002_aurora, the very same exploit used during Operation Aurora against Google:

attacker#msfcli
exploit/windows/browser/ms10_002_aurora
LHOST=10.10.10.112 SRVHOST=10.10.10.112
URIPATH=/exploit
PAYLOAD=windows/shell/reverse_tcp
LHOST=10.10.10.112 LPORT=443

After professional combinations with python scripts and msf console, we can manually create own exploits for future using in pentests. Below is a summary of the key points of differentiations between the two types:

TABLE 1

COMPARISON OF MANUAL VS. AUTOMATED TESTING

	Manual	Automated
Testing Process	Manual, non-standard process; Labor and capital intensive; High cost of customization;	Fast, standard process; Easily repeatable tests;
Vulnerability/ Attack Database Manage-ment	Maintenance of database is manual; Need to rely on public database; Need re-write attack code for functioning across different platforms;	Attack database is maintained and updated; Attack codes are written for a variety of platforms;
Reporting	Requires collecting the data manually;	Reports are automated and customized;
Cleanup	The tester has to manually undo the changes to the system every time vulnerabilities found;	Automated testing products offer clean-up solutions;
Training	Testers need to learn non-standard ways of testing; Training can be customized and is time consuming.	Training for automated tools is easier than manual testing.

An in-house penetration testing solution enables us to ensure more consistently high levels of security by regularly monitoring our networks, endpoints and web applications between consulting engagements.

IV. CONCLUSION

Penetration testing allows organizations to assess vulnerabilities proactively, using real-world exploits, allowing them to evaluate the potential for their systems to be subverted through hacking and malware schemes in the same manner that attackers employ. Manual penetration tests are still more popular and useful, due to much different vulnerabilities in security sections, for example in human factor. Experienced ethical hackers are used to writing own scripts or even automate one of the stages, in order to proceed quickly and find more security leaks in target systems. This automation process can be improved by using scripting languages.

REFERENCES

[1] A.Z.Piskozub. Using penetration testing in computer networks and systems for increasing level of protection / *Materials of the third international scientific and practical conference FOSS* Lviv 2013. – Lviv, 2013.

[2] Y.Y.Stefinko, A.Z.Piskozub.Using free and open-source operational systems for penetration testing in educational purposes / *Journal of The National University "Lviv Polytechnic", "Computer systems and networks*".– 2014. – № 806. – p.258-263.

[3] Y.Y.Stefinko, A.Z.Piskozub, R.I.Banakh. Penetration testing with Metasploit and scripts. *Materials of the IV-th International Scientific Conference "Information protection and security of information systems"* .-2015.- p.87-89.

[4] D.Kennedy, J.O'Gorman. "Metasploit.The penetration tester's guide". - *No starch press, San Francisco*, 2011. 332p.

[5] Jason Andress, Ryan Linn. Coding for Penetration Testers. *Elsiever - London*, 2012, 321p.

[6]http://www.pentest-standard.org/index.php/Pre-engagement

[7] https://www.pcisecuritystandards.org

[8] Bishop, M. "About Penetration Testing". - *IEEE Security & Privacy*, December 2007, p.84-87.