*AUTOMATED SOFTWARE TESTING AND DEVOPS,
STOCKHOLM, SWEDEN 2022*

Securing the pipeline

Continuous penetration testing

Valdimar Björnsson

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Abstract

The history of penetration testing can be traced back to the Spring 1967 Joint Computer Conference where concerns of computer security where arising for the first time as a result of the Time-sharing computing paradigm. The importance and utility of penetration testing was quickly recognized and since then it has been conducted by government bodies, cyber security experts and reformed hackers. Recent tools have introduced the possibility of automating the process of penetration testing but so far it is impossible to automate rigorous coverage without occasional manual intervention. It is possible to use the Zed Attack Proxy running on a persistent server to run automatic penetration tests in a CI/CD pipeline.

Contents

1

1.1 Introduction

A penetration test, also known as a pen test or ethical hacking, is an authorized simulated cyber attack against a computer system to identify exploitable vulnerabilities, strengths and sources of risk. Penetration testing has many forms, network (external, internal), wireless, web application, social engineering, etc. It is a difficult problem to manually repeat penetration testing in a system that is rapidly being changed by multiple developers. Some known attack vectors can be automated and built into the development pipeline to avoid the tedious task of manually repeating the attack. This essay will motivate the need for automated, continuous penetration testing, discuss its limitations and introduce possibilities of including it in the development pipeline.

1.2 The inception of penetration testing

Many renowned computing experts gathered at the Spring 1967 Joint Computer Conference to explore system security challenges introduced by the new prevailing Time-sharing paradigm of computing. This is where Willis Ware, Harold Petersen, and Rein Turn of the RAND Corporation, as well as Bernard Peters of the National Security Agency (NSA), all used the term ”penetration” for an attack against a computer system for the first time. As a result of this conference computer penetration would be formally identified as a key threat to online computer systems.[6] [7] To gain a better understanding of computer system vulnerabilities, the US government and its contractors organized tiger teams. These teams of crackers were to simulate computer penetration to assess system security. They were both government and industry-sponsored and aimed to break through computer system defenses in order to find and remedy security gaps. [10]

According to a number of RAND experts the penetration test exercises all provided multiple benefits that supported their continued usage. ”A penetrator seems to develop a diabolical frame of mind in his search for operating system weaknesses and incompleteness, which is difficult to emulate.” Therefore RAND advised that penetration techniques be studied further for their utility in assessing system security. [6]

Since then penetration testing has been used in an attempt to secure computer systems against external attackers.

1.3 Conducting a penetration test

A penetration test usually begins with identifying the target within a company's system, followed by an assessment of available data and finally an attempt to reach the target. The subjected system of a penetration test can be a white or a black box, and the test can be conducted internally or externally. The most rigorous form of penetration testing is known as covert penetration testing.

A penetration test usually follows the same fundamental steps:

1: Reconnaissance - Acquiring the necessary data to attack the target system.

2: Scanning - Use existing tools to scan for open ports and gain more information about the target system

3: Gaining access - The attacker can utilize a payload to exploit the target system using the information obtained during the previous steps.

4: Maintaining access - The attacker attempts to remain in the target system for as long as feasible in order to collect as much data as possible.

5: Covering tracks - To stay anonymous, the attacker must erase all evidence of hacking the victim's machine.

[9]

The system owner should be notified of any security problems discovered during the penetration test. Penetration test reports may also examine the organization's possible risks and provide countermeasures to mitigate the risk. The fundamental purpose of a penetration test is to uncover vulnerabilities that could be exploited by a a malicious hacker and to tell the client of such vulnerabilities as well as recommending strategies to cover these vulnerabilities. [1]

Different types of penetration testing

White box pen test

The hacker will be given certain security information about the target company ahead of time.

Black box pen test

The hacker will not be given any security information aside from the name of the target company.

External pen test

The hacker takes on the company's external-facing technologies, such as its website and external network servers. In many cases the hacker may not be allowed to enter the company's building.

Internal pen test

The hacker conducts the test using the company's internal network. This type of test can help you figure out how much damage an intruder can do once he has breached the company's firewall.

Covert pen test

During this test almost no one in the firm, including the IT and security specialists who will be responding to the attack, is aware that it is taking place. This is done to closely simulate the environment of a real attack. To avoid any complications with law enforcement, it is vital for white hat hackers to have a formal written copy of the scope and all information of the test written down ahead of time.

[2]

1.4 Testing within companies

Many firms began to use penetration testing as a means of securing their systems against genuine attackers. The ideal strategy was to employ a third-party attacker. Someone who has little to no prior knowledge of how the systems defenses because they may be able to uncover security flaws that the system's engineers missed. As a result, outside contractors are frequently hired to conduct the tests. Since they are employed to hack into a system with explicit permission and for the goal of strengthening security, these contractors are commonly referred to as ”ethical hackers” or ”white hat hackers”.

Many ethical hackers are experienced programmers with advanced degrees and cyber security certifications. However some of the top ethical hackers do not have formal education in the field of penetration testing and are instead self taught. In fact some are ex-criminal hackers who now utilize their skills to assist address security weaknesses rather than exploiting them. Depending on the company and the type of penetration test they want to do, the best candidate to conduct a pen test can differ substantially. For example some may have skills in different technical fields while others are experts at social engineering.

However the process of finding and employing a suitably skilled ethical hacker is expensive and time consuming. Each test may take a long time and in the current landscape of software development, quality manual penetration testing cannot be performed between each iterative change to a system. To assist with this problem some automated testing tools have been developed, this is discussed in the following section. [8]
[3]

1.5 Automated testing

Tools that automate penetration testing are built on commonly successful attacks. Since there's not a need for a human to manually conduct the test, automated penetration tests are significantly faster and they can produce results in as little as a few seconds to a few minutes. Unlike manual penetration testing, automated security testing does not probe deeper into a vulnerability to uncover ways to attack it. Instead, it simply lists the discovered vulnerabilities. The results are then scrutinized by a human to eliminate false positives. Therefore there is still a manual element to complete penetration testing when tests fail.

Where automated tests are quick and easy to use, manual penetration testing is an excellent way to assess the severity of a vulnerability exploit once it is discovered. In the following table, the methods are compared.

Table 1.1: Differences between automated and manual penetration tests

Automated Pentest	**Manual Pentest**
Automatically detect vulnerabilities with a testing tool.	Manual detection of vulnerabilities performed by human tester with cyber security experience.
Time frame is often minutes to hours	Time frame is often days to weeks
Efficient and low effort.	High effort, requires preparation and coordination.
Makes sure new changes do not introduce obvious vulnerabilities. – These are common vulnerabilities such as privilege escalation / file access problems / SQL injection.	Can detect obscure, hidden vulnerabilities. – These are often specific to the system introduced by coding flaws related to the business logic.
Provides an overview of vulnerabilities.	Provides deeper insight into vulnerabilities.
Does not exploit vulnerabilities to see the effects	Exploits the vulnerabilities to see the effects and assess the risk.
Can be done automatically every time a change is introduced.	Cannot be done frequently without severely disrupting development.

Automated penetration testing excels at spotting common vulnerabilities and can do so quite fast, however it cannot be relied on completely. Even though automated testing tools will surely outperform the amateur developer in discovering vulnerabilities, it can’t test uncommon and complex vulnerabilities as reliably as an expert can. And it can only handle those vulnerabilities it has encountered before, there is no creativity. Since automated penetration testing is based on deterministic algorithms, it produces consistent findings under identical circumstances. Therefore an automated test does not cover potential vulnerabilities as well as an expert and in terms of compliance standards, it is insufficient. A manual penetration test performed by a human can uncover business logic issues, coding flaws, and gaps that automated scanners can't yet detect. As a result, manual penetration testing lives on. [4]

1.6 Implementing automatic pen tests

The Open Web Application Security Project (OWASP) is a non-profit organization dedicated to the creation of open-source technologies. OWASP's creators designed and maintain ZAP, a free and open-source PEN testing tool. ZAP is the acronym for Zed Attack Proxy. It's possible to employ ZAP's automated capabilities in a CI/CD pipeline, (it can also be utilized manually). The tool can be used as a stand-alone program with a graphical interface or as a background process. Docker is supported as well as a variety of operating systems. ZAP can be used in many ways, such as an intercepting proxy, a spider or a scanner. ZAP will need to run on a persistent server but the application or script that will communicate with ZAP for the penetration test does not need to be on the same server as ZAP.

ZAP mimics a hacker by acting as a ”man in the middle proxy.” It intercepts user requests and can manipulate them in order to alter the behavior of the targeted application. ZAP acts as a ”middleman” between the two systems to make this possible. ZAP can also be connected to an existing proxy server. ZAP can then be integrated into a CI/CD pipeline, causing the pipeline to fail if the application contains any obvious vulnerabilities .

There are some configurations that need to be made in order to run ZAP.

The spider to be used which can either be the classic spider or the AJAX spider which is slower but more thorough. These spiders ”crawl” the application, issue requests and collect their responses.

Aside from that, one has a choice of two different scanning types. Passive scanning is a secure method of verifying your applications security because it does not alter the responses. However, active scanning entails launching a full fledged attack on your application. The replies that the application sends back to the client are altered by ZAP then ZAP looks for more comprehensive weaknesses that it may exploit and may put your application at risk.

Requests and responses are recorded. In case something goes wrong with a request, it is indicated in the log. The overall number of warnings is tracked by ZAP, which divides them into five risk categories ranging from false positives to high risk. One should manually investigate the request and response of a specific item if a risk indicates a vulnerability. The response header and body are displayed, as well as the piece of data that triggered the alert. To effectively analyze this the investigator should have a thorough understanding of the application.
[3] [5]

1.7 Conclusion

Penetration testing is an important tool to aid the development of secure software. It is possible to use tools that automate the process but they cannot fully replace security experts. There is a need for manual intervention to gain a thorough understanding of the risk a vulnerability poses and to rigorously test a systems security. For optimal security, automated penetration tests should occur frequently, ideally with each change to a system, while manual penetration tests should be done periodically when big changes are made.

References

[1] URL: https://www.sciencedirect.com/topics/computer-science/open-source-security-testing-methodology-manual.

[2] URL: https://www.cloudflare.com/learning/security/glossary/what-is-penetration-testing/.

[3] URL: https://stelligent.com/2016/04/28/automating-penetration-testing-in-a-cicd-pipeline/.

[4] URL: https://www.getastra.com/blog/security-audit/automated-penetration-testing/.

[5] URL: https://amazic.com/penetration-testing-think-and-act-like-an-attacker/.

[6] Hunt, Edward. "US Government Computer Penetration Programs and the Implications for Cyberwar". In: *IEEE Annals of the History of Computing* 34.3 (2012), pp. 4–21. DOI: 10.1109/MAHC.2011.82.

[7] MacKenzie, D. and Pottinger, G. "Mathematics, technology, and trust: formal verification, computer security, and the U.S. military". In: *IEEE Annals of the History of Computing* 19.3 (1997), pp. 41–59. DOI: 10.1109/85.601735.

[8] MacKenzie, Donald A. *Mechanizing proof: Computing, risk, and Trust (Inside Technology)*. MIT Press, 2001.

[9] *Penetration test*. Apr. 2022. URL: https://en.wikipedia.org/wiki/Penetration_test.

[10] Russell, Deborah and Gangemi, G. T. *Computer security basic*. O'Reilly, 1992.

www.kth.se