Literature Study of Penetration Testing

Michele Fiocca
Email: micfi931@student.liu.se
Supervisor: Anna Vapen, {annva@ida.liu.se}
Project Report for Information Security Course
Linköpings universitet, Sweden

Abstract

The Literature Study of Penetration Testing project will address aspects regarding how much vulnerable computer systems are and what effort that is needed to break into a system where the access is restricted or the target is remotely located. Penetration testing is the art of using different tools to get unauthorized access to a victim's computer. Thus, the main task of a penetration tester is to find vulnerabilities or security flaws of new programs and systems to make the development team aware of required modifications in order to increase the security. This report also contains a review of the tools used in the different phases of penetration testing.

1. Introduction

Questions to be addressed in this paper include:

- What is a penetration test and why is it useful to perform it?
- Which type of penetration tests exist and what are their differences?
- What are today the most common tools for penetration testing?

Today the number of security problems seem to be sharply increasing due to increasing complexity of software. In this context, systems are becoming more open to attackers, but we don't know how they act and what they look for. So the intent of penetration testing is to put oneself in the attacker's shoes and learn tips and tricks of his activity. Penetration testing is the activity conducted by a penetration tester or auditor, sometimes also called ethical hacker. A group of many testers is called tiger team. This process consists of more phases and its goal is to detect weaknesses of the system under testing giving as much information as possible about flaws and vulnerabilities that have allowed unauthorized access. Because the list of potential insecurities is unknowable and hence innumerable, a penetration testing cannot prove security of the system, just as no doctor can prove that you are without occult disease; thus, it can just prove that the system is vulnerable.

The survey on penetration testing will start from historical background to come to the actual status. This project is mostly theoretical, so the method of work will be an in-depth research conducted by reading different scientific articles. The sources used in this paper are security portals, security organizations and IEEE.

2. Background

According to the Matt Bishop article [6], I will define what a penetration test is using a metaphor to give a better understanding of the problem. Imagine you have just bought an expensive brand new car and you want to prevent a theft. So you ask a friend, a policeman, to keep your car safe. He doesn't know your model, so he has to examine the car and do it as if he were a thief. Before the test begins you must decide what you mean by "steal", that is breaking into the car and drive it away, even though is later returned. Now the goal becomes more specific and the tester asks himself: can someone who hasn't a key drive your car away? The goal of the test isn't to absolutely prevent the theft, being almost impossible, but at least make it harder, limiting the thief's capacities, for example requiring more than three hours for stealing the car. Proved this, the tester must know as much information as the attackers know so that the test does not fail, and the best way to assure it is that he knows everything about the system. Here the penetration test starts, when the tester (the policeman) first has to gather information about the technical specification (for example the owner's manual or mechanic's guide) of the car and get information provided by you.

Afterwards, he must consider that usually attackers have more time to acquire information, but often the best defense is based on the attackers' lack of it. For example, attackers will have different skills and knowledge, going

from a joy rider who wants to try your new car, but does not know how to steal it, to a professional thief who will have easy access to it, through a mechanic from the store that sold you the car and could easily find a key.

The last part of the article is dedicated to consider what resources attackers have, giving the tester time and resources necessary to the task [6]. The example of the car can be brought to the world of computer science by imagining that you are the chief developer of a new software product and want to test its security and resistance to attacks. So you ask a penetration tester to do it for you.

In today's world, communication requirements are becoming more and more a living manner, so we are seeing a steady rise towards technologies that could ensure the three most important requirements, which are confidentiality, integrity and availability, according to the C.I.A. model. In a nutshell, the Confidentiality explains that messages sent from a source to a destination must be immune to eavesdropping attacks, that is nobody should sniff and record any data sent over the channel between the two party. Then Integrity explains that nobody should alter the content of the messages, trying to modify, insert or delete non authorized content. The Availability of a system should not be compromised by any malicious attack (Denial of Service attacks are the most common), whose purpose is to prevent the system to operate in the correct manner, often causing interruption of service [9].

Many steps forward have been made in the field of security since Internet has become widespread and has begun to manage people's lives. Hence, commercial companies increasingly need new technologies to protect their internal networks preventing unauthorized access aimed to sniff sensitive data, like social security number, password, credit card number and so on. Security mechanisms mostly used today are firewall filtering, clear separation in sub domains, VPN (Virtual Private Network that provides tunneling and cryptography), DMZ (Demilitarized Zone, a portion of a network that separates a purely internal network from an external network as is defined in [5]), end-point authentication that ensures confidentiality, and Intrusion Detection Systems (IDS). IDSs are systems able to prevent and detect any unwanted intrusion through a deep packet inspection aimed to find any matching with a signature database (each signature is a set of rules pertaining to an intrusion activity) previously wrote by skilled network security engineers. If a packet matches a signature in the database, a threat is found and an alert is generated. All these measures seem to be quite sufficient to assure a good data protection, but every day new vulnerabilities are discovered. In this context, system administrators need mechanisms for testing these technologies [9].

2.1 Historical background of penetration testing

Penetration testing was among the first activities performed when security concerns were raised many years ago [3]. The basic process used in penetration testing is simple: attempt to compromise the security of the mechanism undergoing the test. In earlier years, computer networked operating systems, with their access control mechanism, were the most suitable components for penetration testing, because o.s. is the core component of a machine and so more exposed to security threats [3].

The major advances in penetration testing technologies have been made by the IT Security practitioners, while indeed software tools were written with the IT Security Professionals as its principal users [3].

The earliest penetration testing processes were highly ad hoc and manually intensive, while later automatic processes started to be utilized for clearly reduce cost [3].

2.2 Needs for penetration testing

Testing is important for companies that want to guarantee the best product before distribute it. The results are used to find out security flaws and to patch them before it will be too late. Usually however companies lack of time and resources, and consequently penetration testers have reduced amount of resources. This brings testers to widely adopt automatic tools, as it is demonstrated by the continue releasing of platforms finalized to automate this process, *SolarSword* being one of them [7].

2.3 Relevant cases of penetration testing

In this section we will see a real-life example of penetration testing that involved the civilian government agency FBI. The example is taken from an article in Computerworld [2] that speaks about the penetration tester Chris Goggans that has been working as a penetration tester since 1991. One of his latest exploit was against the FBI. It only took him six hours to break into a crime database without permission. This is how he acted: he discovered a series of unpatched vulnerabilities in the civilian government agency's Web server, used a hole in the Web Server to pull down usernames and passwords that were reused on a host of enterprise systems, therefore he got Windows domain administrator privileges gaining full access to almost all Windows-based system in the enterprise, including workstations used by police officers. Finally, remotely controlling them he found programs on their desktops that automatically connected the workstations to the FBI's

crime database. This vulnerability could have been eliminated through a clear separation of domains such as between the police network and the enterprise network.

3. Penetration testing

There are two categories of penetration tests: black box and white box testing. The meaning of these two types of testing differs accordingly to the resources given to the tester, that is, in the first nothing of the system is known or just one resource, for example an IP number. The latter instead implies that tester knows very well the architecture and implementation of the system under test [10].

A simplistic form of penetration testing is to use the popular Google Search Engine. In "Google Hacking for Penetration Testers" by Johnny Long [1] many tricks to get information from the engine are explained. In fact while it is far more than a security tool, Google's massive database is a good resource for security experts and penetration testers. It is possible to use it to discover preliminary information on services of a target company by using directives such as "site:target-domain.com", find employee names, browse sensitive information that they wrongly thought was hidden, view images from web camera considered protected, trace vulnerable software installations, map the network and more. Similarly, when a bug is found in another popular web application, Google can often provide a list of vulnerable servers worldwide within seconds, giving very hot information to a well trained attacker.

3.1 Understand results

A typical result of a well conducted test is to get access to protected resources, exploiting vulnerabilities found by scanning and probing any possible insecurity. Reports are then produced for examination and for measures to be taken, like patches or others activities such as Vulnerability Assessment Mitigation. This is a method periodically used by companies to assess effectiveness of internal and external protection mechanisms. It should be performed several times a year due to the new vulnerabilities discovered.

4. Evaluation

This section contains an evaluation of penetration testing tools chosen from the large amount of resources available on Internet. The comparison is only literature-based and will not rely on experiments. A particular overview of a new penetration testing platform based on Opensolaris will report the innovation in this field [7].

4.1 An Easy-to-Deploy Penetration Testing Platform

Three computer scientist students [7] have built and presented a new platform called *SolarSword* for penetration testing developed on Opensolaris. This platform overcomes the limitations of the previous platforms implementing an automatic, easy-to-deploy, environment that allows testers to work with restricted conditions of time, as it is common for many companies. The distributed system is based on a client/server architecture presenting a central control center coordinating operations and many testing clients FTP-connected with the server. The reason for this architecture has to be addressed to three assumptions made by the authors:

1) Test should be performed in a more automatic way.
2) Test should start from many different positions, therefore a quick deployment is needed.
3) The platform needs to be immune to attacks and likely not controlled by external attackers.

The template generator of the control center generates an initial testing plan that is interpreted into real testing scripts and sent to the distributed clients. These use the scripts to perform the initial tests for gathering information and then upload results to the control center. At this point, the control center knowing the vulnerabilities of the system tested, through an automatic analysis and decision-making module generates further scripts to exploit them. This module will receive results of scanning from clients and then will use the information as input, apply the attack tree or attack graph tools on it to generate possible attacking paths and decide what kinds of weapons to use, in order to perform the real exploitation tests. Afterwards clients upload result to the control center that generates the final evaluation report.
In the paper is also proposed a real test case study in which is exposed the use of the *SolarSword* platform with the security tools Nmap 4.03 and SPIKE 2.9. The network tested is a 100Mbps Ethernet with around 30 hosts. Three procedures are performed: gathering information of target network and finding active hosts, vulnerability scanning on selected host, performing a real vulnerability exploit with the security leaks found in the second step of this test. All tests are done in an automatic, controllable procedure, simple and easy to perform, without the need of introducing highly-qualified penetration testing specialists, as it is stated in the conclusion of [7].

4.2 Popular tools

This section gives an overview of penetration test tools based on "*Penetration Testing Tools*" by Kenneth R. van Wyk [4] and will answer to the following question: What tools could be used to perform a penetration test?

Tools presented in this paper are intended to serve as examples of particular types of tools, and the overview is not comprehensive of the large amount of commercial or open source software available on Internet. According to the article [4], tools for penetration testing could be divided in four categories, 1 and 2 are the most common types:

1. Port scanner
2. Vulnerabilities scanner
3. Application scanner
4. Web Application assessment proxy

4.2.1 Port scanners

Port scanners are pieces of software designed to search a network host for open ports. They are often used by attackers to compromise the security of a network system, and it is the first step for any unauthorized intrusion in a computer network system.

A popular tool for port scanning is Nmap, a "Network Mapper", which through port scanning is able to discover computers and services on a computer network, thus creating a "map" of the network. In addition Nmap may be able to determine various details about the remote computers including operating system version, device type, uptime, software product used to run a service, exact version number of that product, presence of some firewall techniques and, on a local area network, even vendor of the remote network card. The software takes as input an IP number of a subnet and performs a scan searching for active hosts, then scans open ports looking for running services behind them. In this manner a tester could discover vulnerabilities to penetrate and later exploit [11].

4.2.2 Vulnerability scanners

Vulnerability scanners are vital tools to a traditional penetration tester because they provide essential means of probing each and every available network service on the targeted hosts. Vulnerability scanners work from a database of documented network service security defects, trying to probe each defect on each available service of the target range of hosts. Flaws could be found in configuration weaknesses as well as in unpatched network software of the target systems. These defects are exposed to the tester tool and the tester could rapidly and exhaustively look for them. This type of scanner is generally able to scan any TCP/IP device present on a network researching any operating system level weakness. On the contrary it could not compromise the security of general purpose applications, because it does not have any knowledge base of how an unknown application works [4].

Nessus is a popular free (but not open source anymore) constantly updated tool that allow a tester to first retrieve version and type of the target host operating system, and then scanning for known weaknesses and trying to exploit them. Nessus doesn't require deployment of agents on the target systems to perform vulnerability scans and find all the main information allowing a bird's-eye view of the whole reality of interest. Nessus runs on Windows-based machines and is considered one of the best available vulnerability scanners with its large number of plugins. It boasts of including a client/server architecture with a graphical interface, a remote and local (authenticated) security checks and an embedded scripting language for writing your own plugins or understanding the existing ones. On Nessus's website [14], there is a demo of how the tool works through a thorough analysis of its features. Its key principle is the idea of having a scan policy, which permits the user to set parameters and variables for a more successfully scanning, such as scan options, credentials, plugins and advanced settings.

CoreImpact is an automated, comprehensive penetration testing commercial software produced by CoreSecurity. It has a regularly updated large online database of professional exploits, and is considered to be one of the most powerful tools for exploitation currently available, having the ability to force access to the system under test. It also has extra features such as establishing an encrypted tunnel through an exploited machine to reach and exploit other machines [11].

QualysGuard is a commercial web-based vulnerability scanner with reduced cost. These are some of the features of the product: completed and daily updated to monitor thousands of vulnerabilities, automatically providing direct links to controlled remedies, easy to install without additional software/hardware to maintain and update, provided with an inference-based scanning engine, easy to use Web interface, large vulnerability checks database and fast reports with ranking of the vulnerabilities.

ISS's InternetScanner is an application-level vulnerability assessment that started off in '92 as an open source scanner by Christopher Klaus and later developed by IBM. It has many of the previous scanners' features plus a dynamic assignment of performances by automatically increasing scanning speed and accuracy according to the OS of the target hosts [12].

Metasploit Framework was released in 2004, causing a storm in the security information world. It allows a tester to develop, test and use exploit code through an advanced open-source platform. The possibility to use the Metasploit Framework to integrate payloads, encoders, no-op generators, and exploits into an extensible model has made it a beacon for the large community of security researchers. Nevertheless it contains hundreds of exploits ready to be tested, and it also gives the possibility to write your own exploit in an easy way [13].

4.2.3 Application scanners

With respect to application scanners, these tools can perform black box testing and scan web applications to find vulnerabilities. They are more powerful than their network-based vulnerability scanner cousins. Practically these tools function in this simple way: they attempt to do probing of a targeted web application exercising an attack taken from a list of common and known attacks; the list include cookie manipulation, cross-site scripting (referred to as "XSS" attack), SQL insertion, buffer overruns, and so on. A web application passing all tests does not mean it is safe and resistant to any attack [4]. One tool is reviewed in this paragraph: AppScan produced by Watchfire.

This tool is only available for Windows systems and provides in-depth scans for many common vulnerabilities affecting web server and simulates attacks on application level to discover new security flaws. List of common attacks include "XSS" cross site scripting, splitting of HTTP response, cookie manipulation, buffer overflows and more [11].

4.2.4 Web Application Assessment Proxy

They are probably the most useful scanners among those listed above, because they work by interposing themselves between the tester's web browser and the web server of the target. In this way a tester can furthermore manipulate data exchanged by the two parties changing some critical values, for example cookies, that can result in an anonymous identification of the client. One tool suitable for this use is ParosProxy [4].

ParosProxy is a Java based web proxy for web application vulnerabilities assessment. It can sniff HTTP(S) packets and change entries such as cookies and form fields. In the program packet are included useful tools that extend the capabilities of the tool, the most important of them being a scanner for common web attacks [11].

5. Conclusions

After having dealt with penetration testing in a careful analysis, we can now answer to the questions given in the introduction. So we have seen that a penetration test is a complex activity performed by security experts aimed to discover vulnerabilities of a system, going from a computer network of a company to new software ready to be released. The result of such a test is analyzed and then countermeasures are taken to patch any possible weakness. There usually exist two types of penetration tests; white box tests in which system implementation is known to the tester and expected results derivate from a clear choice of an input, while a black box test is conducted only knowing a resource like an IP address of a host inside the network. We continued with the analysis of a new platform that has brought penetration testing towards a more automatic way of working. We have then evaluated a long list of tools sorted by category but more attention has been dedicated to vulnerability scanners through a more extended review of tools like Nessus, considered the world-leader in active scanners.

Security analyst Bruce Schneier is wondering in his blog [8]: *Is Penetration Testing Worth it?* The answer he gives at the end is no. Shortly, he says that penetration testing could be just a waste of time and money because, although it is a convenient activity, the result of testing is a document containing all the problems requiring attention, but most companies do not have the budget to fix it all. What can we say is that the increased complexity of software and systems has brought the attention towards security issues as long as the number of potential attackers is continuously growing.

References

[1] J.Long, "Google Hacking for Penetration Testers", e-book.

[2] "Six hours to hack the FBI (and other pen-testing adventures)"http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9087441", 2009-04-25.

[3] Kenneth R. van Wyk, "Adapting Penetration Testing for Software Development Purposes", 2007, Carnegie Mellon University.

[4] Kenneth R. van Wyk, Software Engineering Institute, "Penetration Testing Tools", 2007, Carnegie Mellon University.

[5] Matt Bishop, "Introduction to Computer Security", Addison-Wesley.

[6] Matt Bishop, "About Penetration Testing", Security & Privacy, IEEE.

[7] B. Duan, Y. Zhang, D. Gu, "An Easy-to-deploy Penetration Testing Platform", The 9th International

Conference for Young Computer Scientists, 2008. ICYCS 2008.
[8] Bruce Schneier Blog http://www.schneier.com/blog/archives/2007/05/is_penetration.html, 2009-05-01.
[9] James F. Kurose, Keith W. Ross, "Computer Networking – A top Down Approach", 4th edition, Addison Wesley Computing.
[10] Dr. Daniel Geer and John Harthorne, "Penetration Testing: A Duet", Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC '02).
[11] "Nmap - Free Security Scanner For Network Exploration & Security Audits" http://nmap.org/, 2009-05-08.
[12] "IBM – Features" http://www-935.ibm.com/services/us/index.wss/detail/iss/a1027213?cntxt=a1027208, 2009-05-08.
[13] "The Metasploit Project" http://www.metasploit.com/, 2009-05-08.
[14] "Tenable Network Security" http://www.nessus.org/nessus/, 2009-05-08.