

External footprinting security assessments

Combining the PTES framework with open-source tools to conduct external footprinting security assessments

Bruno Dinis

ISCTE-IUL/ADETTI-IUL

Av. das Forças Armadas, 1649-026, Lisbon, Portugal
dinis@ua.pt

Carlos Serrao

ISCTE-IUL/ADETTI-IUL

Av. das Forças Armadas, 1649-026, Lisbon, Portugal
carlos.serrao@iscte.pt

Abstract— One of the first phases, and one of the most important ones, in a security assessment activity (either legitimate or not) consists in the information gathering about a specific target. Information gathering, also recognized as *footprinting*, is the process of collecting all accessible information about that specific target. In a security assessment, the importance of this phase is clamorous and involves the examination, collection and classification of large volumes of data from the target network. The Penetration Testing Execution Standard (PTES), although still in an early and definition stage, provides the description of the processes that are necessary to conduct penetration-testing assessments in a generic and integrated manner. The focus of this article consists in the analysis of the PTES and its recommendations on what concerns *footprinting* processes and to provide some contributions in terms of the practical applicability of the PTES recommendations.

Keywords—component; penetration testing, pentests, network vulnerabilities, PTES, footprinting;

I. INTRODUCTION

Information gathering is an important stage of a systems security assessment. It allows testers to collect all the necessary information about a target or a set of targets that will allow them to have a full insight of the target prior to any security assessment. It is a crucial stage in a specific type of security assessment known as penetration testing. Penetration testers, also known as "ethical" or "white hat" hackers, are tasked with discovering information security vulnerabilities that may be exploited by an attacker (before it actually becomes exploited by him). Knowing our own vulnerabilities before our enemies has always been one of the best ways to protect against them [20].

One of the known penetration test frameworks is the community-driven emerging Penetration Testing Execution Standard (PTES). This framework, although still in early development stage, provides a structured methodology to a motivated community aimed to openly specify what penetration assessments are and which are the steps involved. Also, one of the objectives of PTES is to provide clients with a benchmark to determine the quality of tests carried out for them by contracted penetration testers. Throughout a conventional expression and scope to accomplish penetration tests, this framework aims to increase the global quality of penetration testing assessments and specially supporting

businesses to define what they need to perform a security assessment to their systems and what to expect from the different penetration testing phases.

This methodology is well divided into seven phases. It sets a normalized approach to the following defined set of stages: pre-engagement interactions, intelligence gathering, threat modeling, vulnerability analysis, exploitation, post-exploitation and reporting phases [1].

In this article, the authors will focus their attention in the second stage of this methodology - intelligence gathering - while providing some open-source tools and examples on how the PTES recommendations can be fulfilled to conduct external *footprinting* (a *footprinting* that is conducted from the outside of the evaluating targets). Although extremely exhaustive on the process description, the PTES often fails to provide a mapping between these processes and its practical applicability on the field. One of the objectives of this article is to provide a contribution to PTES while offering a mapping for the intelligence gathering process inside PTES and some of the tools that can be applied to accomplish this stage.

The article starts by providing an overview of the different stages that compose the PTES, while offering a small description of each of the stages. The second part of the article describes in more detail the *footprinting* processes in the PTES, while it provides more details about tools and methods that might be used to conduct this stage. Finally, at the end of the article some conclusions are presented.

II. STAGES OVERVIEW

The PTES establishes several stages where security audits need to be conduit and provides a set of guidelines that need to be followed to meet the objectives of each of the stages (or the expectations of potential clients) [1]. These stages and guidelines can be resumed in the following:

1. Pre-engagement. The first phase describes all the pre-engagement actions and scope descriptions. Scoping is possibly one of the most significant and frequently ignored sections of a penetration test. It provides an important guideline about what to consider or not on the test.

2. Intelligence gathering. In this stage, is specified the information collection necessary to produce an intelligible representation of the business and its own procedures. All the

data picked during this stage will contribute to guide all the evaluation of the possible organization vulnerabilities. It will also determine what kind of IT infrastructure is used, information about workers, products and operations.

3. Threat modeling. Based on the previous stages, this phase covers the fundamentals of the threat modeling. In a conventional description this segment outlines a risk modeling approach as mandatory for a precise implementation of the tests. This framework does not endorse a specific model, however compels that the standard used must be reliable in terms of its depiction of risks, their potentials, their limitations for the organization being analyzed, and the skill to constantly be requested to future penetration tests with the similar outcomes.

4. Vulnerability analysis. Also known as vulnerability assessment, this level describes the key subjects the vulnerability analysis must cover. It is a method that describes and categorizes the security vulnerabilities, while defining and sorting network or system resources that can provide different levels of importance to these resources. In addition, recognizing conceivable threats to each resource develops a plan to deal with the most severe possible problems. The major objective at this point is to compile a high value target list and attack vectors in order to determine the impact on the organization in successful attacks. Conceiving this target list will also define and fulfill security methods to diminish the costs if an attack occurs.

5. Exploitation. This represents the phase, where the attacker effectively exploits the target. However, it is mandatory to take attention on all the preceding tasks in order for this one to be fruitful. Exploitation is the triumphant manipulation or misuse of defenseless equipment, service or somebody to eventually gain access to data or information otherwise unreachable. This process involves the penetration tester to have persistently investigated and proved all potential threats to the target by effectively leveraging all earlier obtained knowledge of the target. This phase concentrates exclusively on establishing access to a service or machine by bypassing safety boundaries. If in the previous stage, vulnerability analysis were achieved accurately, this stage would be organized and accurate. The foremost effort is to detect the leading entrance point into the organization system and to recognize high value target assets.

6. Post exploitation. After the penetration tester has control over an objective target and approaches the post-exploitation stage, and the main goal is to stay undetected and got future control on the exploited systems. Starting the exploiting methods all over again is not a good practice. A standard procedure is covering used scripts to regain access when needed. Thus this technique prepares the exploited system to take advantage of it in the next time. Other method of post exploitation is attacking the system from the exploited machine, covering all the eventual tracks of the manipulative activities. The reason of this last phase is to verify if the target compromised is worthless or not and provide access of the system for further use. The significance of the compromised system is established by the sensitivity of the data stored on it and the valuable machines in further attacks to the network [1].

7. Reporting. To summarize his work the penetration tester should create a document where he uncovers all security vulnerabilities discovered during the system audition. This document is projected to outline the base principles for penetration testing and report all the vulnerabilities discovered to those who are responsible for that. Network and system administrators must be warned about infrastructure vulnerabilities, developers must be advised about weaknesses in their code design [2]. Don Williams refers that “*writing a penetration testing report is an art that needs to be learned to make sure that the report has delivered the right message to the right people*” [3]. This record concludes the penetration testing in the organization and should be taken seriously by the management and administration to mitigate future weaknesses and exposures in their systems.

III. FOOTPRINTING IN THE PTES

Reconnaissance and *footprinting* are similar concepts. Nevertheless it is essential to understand the differences between them. Reconnaissance can be defined as the initial phase of the security assessment activity where attackers will attempt to learn and collect as much as possible evidences about the target. This information is gathered to characterize the infrastructure under investigation. Although *footprinting* can be labeled as a passive and non-intrusive reconnaissance technique, tolerating the potential attackers to accumulate all potential information about the target without the requisite of employing forceful reconnaissance methods [4], can be quite dangerous and imprudent (this is usually the first step for an attack). However, according to the PTES framework, *footprinting* can be both passive and active. For instance, conducting a review of the website of an organization is a type of passive *footprinting*, while intensive scanning of machines and networks can be considered as an example of active *footprinting*. They both can be evasive and silent techniques and it is important to know when to apply them. The PTES framework defines the *footprinting* technique as being part of the second overall phase of the penetration testing procedure - the information-gathering phase. Divided into two different spheres, the *footprinting* could be performed externally or internally. In the external mode, the framework identifies the customer external ranges, conducts passive reconnaissance, active *footprinting* and also establishes an external target list. On the other hand there is also the internal *footprinting* that covers three main topics: the passive reconnaissance, customer internal ranges identification and the active *footprinting*.

As it was previously referred the focus of this paper is on external footprinting that follows the best practices described by the PTES, and on the mapping of open-source tools and techniques that allow the implementation of such good practices. In the following sections of this paper, some examples of the execution of external *footprinting* are provided [13].

These examples were performed in a controlled and authorized network environment using a virtualized laboratory. In accordance to the law, in most countries, some of the tests and techniques when used in a non-authorized manner in non-authorized targets are considered a serious crime.

A. External footprinting

During an external penetration test, it is necessary to perform an assessment on all Internet accessible assets (all the organization assets that are accessible from the exterior of the organization using an Internet connection). This way, an evaluation of the target security from an outsider perspective is conducted. The external *footprinting* process of intelligence gathering stage comprises collecting answers from the objective target created from an exterior point of view. The purpose is to accumulate as much information about the target as possible. The PTES framework covers the techniques that are presented in the following sections.

1) Identifying customer external ranges

The intelligence gathering stage is critical for the penetration testing execution. The main objective during the security audit is fixing all the possible target hosts. With these hosts in mind the audit can be launched.

2) Passive reconnaissance

From the external point of view, the passive reconnaissance is based on the search of publicly available information about the target on the Internet. This is sometimes also known as OSINT (open-source intelligence) [14]. This process of information collection about an intended target occurs silently without any knowledge of the victim. Any valuable search engine (such as Google, for instance) can support security auditor's work offering lots of background information about almost any target [5][15]. Since the web address (or IP address) is public, loads of sensitive information can be found while performing this kind of passive exploration.

Conducting a search in the chronologic information hosted in the website of the target can also prove to be quite useful. The public online service "Internet Archive" [6] saves website information since 1996 and it makes possible to seek information about the target since that date. For instance, it is possible to search in this archive for a specific website URL and it will reveal how many times the website has been crawled since it was launched by the first time [6].

Usually, passive reconnaissance can also embrace physical inspection of an organizational structure, seeking over wasted machines or other equipment in an effort to locate hardware that may include valuable information. Other techniques may also include dumpster diving or simply eavesdropping the network.

3) Active footprinting

Such as the passive counterpart, active *footprinting* consists in collecting information about an intended target system. Also known as active reconnaissance, this concept usually includes port scanning in order to discover flaws in the target system, like vulnerable ports or services that could be used to compromise the firewall or routers and leverage the exploits. Nmap is an open-source network mapper tool [16] that can be used for network discovery and security auditing. Nmap is capable of scanning network ranges. In the following example, Nmap can be used to find which are the active assets on a network, which are the ports that are open or closed on these assets also, which are the services that are running on such

ports. Nmap can be used with different combinations to extract even more information from the targets.

```
nmap 192.168.100.0/24
```

```
Starting Nmap 5.00 (http://nmap.org)
```

```
Nmap scan report for 192.168.100.1
```

```
Host is up (0.016s latency).
```

```
Not shown: 996 closed ports
```

```
PORT STATE SERVICE
```

```
23/tcp open telnet
```

```
53/tcp open domain
```

```
80/tcp open http
```

```
5000/tcp open upnp
```

```
Nmap scan report for 192.168.100.2
```

```
Host is up (0.036s latency).
```

```
All 1000 scanned ports on 192.168.100.2 are closed
```

```
Nmap scan report for 192.168.100.3
```

```
Host is up (0.000068s latency).
```

```
All 1000 scanned ports on 192.168.100.3 are closed
```

```
Nmap done: 256 IP addresses (3 hosts up)
```

```
scanned in 22.19 seconds
```

The method of exploiting the target can then be carried out once the auditor has established a way to access the system from an outside perspective. Banner grabbing is another well-known technique for conducting external *footprinting* of a target system. This technique allows the security auditor to find out which service or application is up and running and on which port. This is the process in the audit that simulates an attacker attempts to locate an application version installed in victim's machine and find any known exploits or vulnerabilities for that specific combination (application and version). With this information, an attacker could exploit known vulnerabilities using specific exploits [7]. There are online public databases of vulnerabilities, indexed by service/software and version that allow access to millions of identifiable vulnerabilities (for instance the Common Vulnerabilities and Exposures (CVE) database) [17].

The example that is provided next allows the security auditor to learn which service the SNMP server is using. Using a simple telnet session in all the reported open ports and waiting for the response banner will provide valuable information. For example, this is the response banner of a SMTP server running PRTG. If the services and versions are revealed during the penetration test and if there are known

vulnerabilities for these services/versions then they could be easily exploited by an attacker.

```
$ telnet 192.168.100.250 25
Trying 192.168.100.250...
Connected to 192.168.100.250.
Escape character is '^]'.
220 snmp.website PRTG
```

Another technique that is mentioned by PTES framework is SNMP Sweeps [18]. Using this allows the discovery of huge volumes of information about a specific system or actually it also allows the compromise of a remote device. When interrogating through SNMP service, there is a MIB (Management Information Base) that permits to request information to the target machine. Metasploit [12] is a free penetration testing software and offers a catalogue of default MIBs database. Applying them, would probe the target machine for extra info. Its auxiliary module could be applied to the following example, showing what IP addresses are using default community strings, which can be easily exploited.

```
msf auxiliary(snmp_login) > set RHOSTS
RHOSTS => 192.168.100.0-192.168.100.255
msf auxiliary(snmp_login) > set THREADS 10
THREADS => 10
msf auxiliary(snmp_login) > run
[*] >> progress (192.168.0.0-192.168.100.255
0/30208...
[*] 192.168.100.50 'public' 'APC Web/SNMP
Management (...)'
[*] Auxiliary module execution completed.
```

Zone Transfers are a vital source of data about the network. This technique is used to allow backup DNS servers to synchronize with their primary servers by querying simple requests for the DNS records for a specified domain. For each of the requests it is provided all the known information about a single domain, although it can be used to gather information about the servers (huge leakage of information). Zonetransfer.me is a domain registered to show how DNS zone transfer works, explaining the security problems concerned [8] and how an attacker can harvest information about an organization DNS.

```
dig axfr @ns12.zoneedit.com zonetransfer.me
<<>> DiG 9.7.3-P3 <<>> axfr @ns12.zoneedit.com
zonetransfer.me
(1 server found)
global options: +cmd
zonetransfer.me. 7200      IN NS
ns16.zoneedit.com.
zonetransfer.me. 7200      IN NS
ns12.zoneedit.com
```

```
zonetransfer.me. 7200      IN A
217.147.180.162
```

The Non-Delivery Report/Receipt (NDR) or the SMTP bounce back is a notification about a mail system. It is used typically to get extra information about the email system, getting the headers and sometimes the infrastructure details by composing a faulty email to the target domain and getting the report on this delivery failure. This is an issue that can simply be simulated by creating a bogus email address within the target's domain. Using this method it would be possible to uncover the email server IP.

Here's an example of a common email provider header:

Delivery to the following recipient failed permanently:

badaddress@website.com

Technical details of permanent failure:

We tried to deliver your message, but it was rejected by the server for the recipient domain "Real IP Email Server Address Displayed"

The error that the other server returned was: 550 badaddress@website.com: Recipient address rejected: User unknown in local recipient table.

Another method mentioned in the PTES is DNS Discovery. Normally mentioned as "whois" technique and widely used on Internet, this routine allows queries to remote databases for domain registration information. After collecting all the information needed using the above methods, the auditor could query the DNS using some open-source tools.

WHOIS information for: website

[Querying whois.dns.com]

[whois.dns.com]

Domain Name: website

Creation Date (dd/mm/yyyy): creation date

Expiration Date (dd/mm/yyyy): expiration date

Status: ACTIVE

Titular / Registrant

Full Company Address

Email: website@website.com

Billing Contact

Full Company Name

website@website.com

Tech Contact

Tech Contact Name
techcontactname@website.com

Name server Information

Name server: Webserver Name type Webserver
URL

Name server: Webserver Name type Webserver
URL

Name server: Webserver Name type Real IP
Address

If the Domain Name Server points the address to his associated IP address, the Reverse DNS method can be handled to achieve valid server names by trying the server with several IP addresses to check if it returns any outcomes. If it does resolve the name then the results are returned. There are some available examples on how to how to configure the Reverse DNS [9].

host 66.40.65.49

49.65.40.66.in-addr.arpa domain name pointer
www.ntchosting.com

After discovering the initial information, a DNS brute force technique [19] can also be used in the information gathering stage according to the PTES framework. Forcing the queries to the DNS server, this tool seeks for misconfigurations about the DNS server to allow users to perform Zone Transfers as it was mentioned before.

Web Application discovery is also a procedure meant to uncover installed web applications on the target system, and vulnerabilities that could be exploited by identifying those applications. To verify which is the OS running on the target system (OS fingerprinting), the auditors can benefit from several open-source tools (or even some web-sites) that are available on the Internet. For instance, the Netcraft website as a service called "What is that site running" that identifies the operating system and the server software that a given web site is running. To use this tool, it is simply necessary to enter the target URL name on Netcraft webpage and all the details are revealed. Besides operating system detection this company offers more vulnerability analysis tools [10].

Search for "website"

OS	Server	Last Change	IP
Linux	Apache	22-Jun-2013	A.B.C.D

The final technique that is enumerated on the PTES framework is the Virtual Host Detection & Enumeration. This technique is used to discover the host names related to a given IP address. HostMap is an open-source discovery software designed to detect hostnames and virtual hosts in the network and performs vulnerability evaluations and penetration testing. This tool assists the auditor on the exploration of several techniques to specify and enumerate all the hostnames related with a specific IP address [11].

4) Establishing external list

The final result of the information-gathering phase is creating external inventory with all the information collected

and of all the uncovered vulnerabilities. This is one of the areas where the PTES framework still lacks on information. The mapping of all the versions of the applications/services/assets gathered, consequence of the exploitation of some of its vulnerabilities, it is critical. This is the information that probably an attacker would use to compromise some of these assets. With this information, an attacker could label all the existing patch levels for all those found applications, interrogating the system with random vulnerability scanners.

An attacker can also look for weaker applications installed, find a breach in its security and then implement the way to exploit it. Once inside the infrastructure he can leverage the privileges and seek for storage setup evidences, virtualization platforms and virtual machines running on the systems infrastructure. The attacker can identify the lockout threshold to perform the attack identifying weaker ports and outdated systems. Finding the lockout threshold of a validation service will permit the attacker to guarantee that some hacks do not deliberately lock out valid users during the testing [1].

IV. CONCLUSIONS

The evolving Penetration Test Execution Standard (PTES) is a collaborative effort, supported by a wiki-based system, meant to offer a penetration testing security service a set of well-defined and standardized practical guidelines. Although still in an early stage, the *footprinting* phase holds a critical significance in the information-gathering phase. In the article all the different seven phases were briefly described, although the article focus was in the *footprinting* process. *Footprinting* reveals the work of information gathering that needs to be carried out before any penetration testing is actually conducted. In this article, some methods based on open-source tools that are capable of gathering information about a target testing system, were also presented. These methods can reveal potential vulnerabilities and attack points on the target's infrastructure. The usage of these open tools will ensure the applicability to the PTES framework, complementing its information and covering one of its weakest aspects.

As it was demonstrated in this paper, the information gathering stage is of extreme importance because it ensures that the next stages on the penetration testing are actually executed in a proper manner, targeting the most relevant vulnerabilities and addressing the proper targets – therefore minimizing the time and effort of the testing itself.

The penetration test methods that were described in this paper are extremely important to secure and mitigate the risk of compromising the security so they are essential to conduct proper penetration testing in a way to conduct exhaustive security audits. An organization should be conscientious that it is important to know its own weaknesses and vulnerabilities before an attacker, in order to be able to raise the necessary fences to keep intruders out. Otherwise, it has little change to resist an attack [20].

REFERENCES

- [1] The Penetration Testing Execution Standard. (2012). Retrieved July 02, 2014, from http://www.pentest-standard.org/index.php/Main_Page

- [2] Wren, C., Reilly, D., & Berry, T. (2010). Footprinting: A Methodology for Auditing eSystem Vulnerabilities. In 2010 Developments in E-systems Engineering (pp. 263–267). IEEE. doi:10.1109/DeSE.2010.49
- [3] Williams, D. (2005). A Guide to Discovering Web Application Insecurities, Before Attackers Do. SANS Institute. Retrieved July 02, 2014, from <http://www.sans.org/reading-room/whitepapers/webserver/guide-discovering-web-application-insecurities-attackers-1557>
- [4] E-CQurity (2011). Footprinting Encored. Retrieved July 02, 2014, from <http://www.e-cq.co.th/wp/footprinting-encored.pdf>
- [5] Mansfield-Devine, S. (2009). Google hacking 101. Network Security, 2009(3), 4–6. doi:10.1016/S1353-4858(09)70025-X
- [6] Internet Archive: Digital Library of Free Books, Movies, Music & Wayback Machine.. Retrieved July 02, 2014, from <https://archive.org/>
- [7] Lanz, J. (2003). Practical aspects of vulnerability assessment and penetration testing. Rma Journal, 85(5), 52-57.
- [8] ZoneTransfer.me - DigiNinja. (2013). Retrieved July 02, 2014, from <http://digi.ninja/projects/zonetransferme.php>
- [9] How does the Reverse DNS work? (2013). Retrieved July 02, 2014, from <http://www.ntchosting.com/dns/reverse-dns.html>
- [10] Netcraft | Internet Research, Anti-Phishing and PCI Security Services. (2013). Retrieved July 02, 2014, from <http://www.netcraft.com/>
- [11] hostmap - the automatic hostnames and virtual hosts discovery tool. (2013). Retrieved July 02, 2014, from <http://hostmap.lonerunners.net/>
- [12] Jaswal, N. (2014). Mastering Metasploit (p. 378). Packt Publishing Ltd.
- [13] Faircloth, J. (2011). Penetration Tester's Open Source Toolkit (p. 441). Elsevier.
- [14] Glassman, M., & Kang, M. J. (2012). Intelligence in the internet age: The emergence and evolution of Open Source Intelligence (OSINT). Computers in Human Behavior, 28(2), 673–682. doi:10.1016/j.chb.2011.11.014
- [15] Billig, J., Danilchenko, Y., & Frank, C. E. (2008). Evaluation of Google Hacking. In Proceedings of the 5th Annual Conference on Information Security Curriculum Development (pp. 27–32). New York, NY, USA: ACM. doi:10.1145/1456625.1456634
- [16] Nmap - Free Security Scanner For Network Exploration & Security Audits. (n.d.). Retrieved July 02, 2014, from <http://nmap.org/>
- [17] CVE - CVE List Main Page. (n.d.). Retrieved July 02, 2014, from <https://cve.mitre.org/cve/>
- [18] Herrero, Á., Corchado, E., & Sáiz, J. M. (2005). Identification of anomalous SNMP situations using a cooperative connectionist exploratory projection pursuit model. In Intelligent Data Engineering and Automated Learning-IDEAL 2005 (pp. 187-194). Springer Berlin Heidelberg.
- [19] Sutton, M., Greene, A., & Amini, P. (2007). Fuzzing: brute force vulnerability discovery. Pearson Education.
- [20] Tzu, S. (2013). The art of war. Orange Publishing.