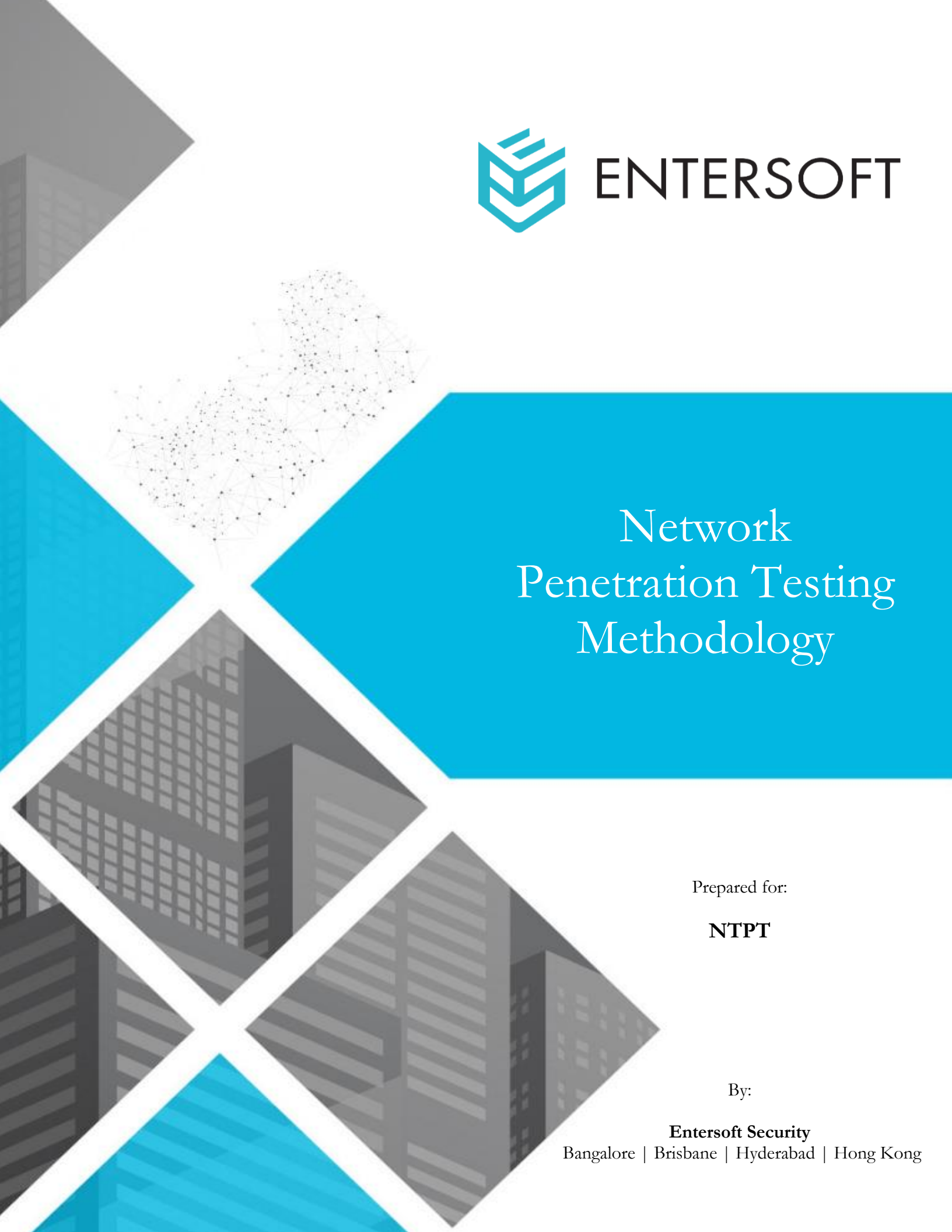

ENTERSOFT

Network Penetration Testing Methodology

Prepared for:

NTPT

By:

Entersoft Security
Bangalore | Brisbane | Hyderabad | Hong Kong

Contents

Revision History & Version Control

Revision History & Version Control		
Release Number	**Author**	**Comments/Details**
1.0	Hussain	Final Draft
Reviewed by	Sai Charan	

1 Introduction

This report document hereby describes the methodology for Network Penetration Testing.

This methodology helps in addressing the below

- Determines the security posture of the network
- Identify the vulnerabilities in the network
- Aid in understanding the risks associated with the vulnerable networks

1.1 Scope

The scope of the project will be defined and documented as part of the project engagement which is done once after the NDA has been signed.

Details collected as part of the Pre-sales

Details collected during the Walkthrough and Project Engagement Phase (after NDA is signed)

1.2 Standards and Guidelines

- OSSTMM
- NIST

1.3 Risk Calculation and Classification

The final risk value of the finding identified is arrived at by considering the likelihood of occurrence of an attack by exploiting the vulnerability and its impact on business.

Following is the risk classification:

		Impact				
		Minimal	Low	Medium	High	Critical
Likelihood	Critical	Minimal	Low	Medium	High	Critical
	High	Minimal	Low	Medium	High	Critical
	Medium	Minimal	Low	Medium	Medium	High
	Low	Minimal	Low	Low	Low	Medium
	Minimal	Minimal	Minimal	Minimal	Low	Low

Likelihood

The difficulty of exploiting the described security vulnerability includes the required skill level and the amount of access necessary to visit the element susceptible to the vulnerability. The difficulty is rated with the following values:

Critical: An attacker is almost certain to initiate the threat event.

High: An untrained user could exploit the vulnerability, or the vulnerability is very obvious and easily accessible.

Medium: The vulnerability requires some hacking knowledge or access is restricted in some way.

Low: Exploiting the vulnerability requires application access, significant time, resource or a specialized skillset.

Minimal/ Informational: Adversaries are highly unlikely to leverage the vulnerability.

Impact

The impact of the vulnerability would have on the organization if it were successfully exploited is rated with

the following values:

Critical: The issue causes multiple severe or catastrophic effects on organizational operations, organizational assets or other organizations.

High: Exploitation produces severe degradation in mission capability to the point that the organization is not able to perform primary functions or results in damage to organizational assets.

Medium: Threat events trigger degradation in mission capability to an extent the application is able to perform its primary functions, but their effectiveness is reduced and there may be damage to organizational assets.

Low: Successful exploitation has limited degradation in mission capability; the organization is able to perform its primary functions, but its effectiveness is noticeably reduced and may result in minor damage to organizational assets.

Minimal: The threat could have a negligible adverse effect on organizational operations or organizational assets.

2 Methodology

This section represents the procedure followed by the team to determine the security posture of the network. The goal of security testing is to identify the threats in the system and measure its potential vulnerabilities, so

the system does not stop functioning or is exploited. It also helps in detecting all possible security risks in the system and help developers in fixing these problems.

- Agreement (Signing NDA)
- Information Gathering
- Technical information gathering through Scanning.
- Gaining Access
- Maintaining Access
- Consolidate the test results and exploit the vulnerabilities and defining the severity.
- Reporting
- Patch Assistance
- Revalidation

2.1 Agreement

In this phase, there is a mutual agreement between the parties. The agreement covers high-level details such as methods followed and the exploitation levels. The testing team will never bring down the production server even if the testing is being done at non-peak hours. The testing team should never disclose any sensitive information about the vulnerabilities to any third party. A non-disclosure agreement must be signed between the parties before the test starts.

Once the penetration testing recommendations are complete, the tester should clean up the environment, reconfigure any access he/she obtained to penetrate the environment, and prevent future unauthorized access into the system through whatever means necessary.

2.2 Information Gathering

In this phase, the attacker gathers as much information as possible about the target. The information can be IP addresses, domain details, mail servers, network topology, etc. An expert hacker will spend most of the time in this phase, this will help with further phases of the attack.

2.3 Enumeration

Enumeration in information security is a computing activity for extracting usernames, machine names, network resources, and other services from a system. All the gathered information is used to identify the vulnerabilities or weak points in system security and then tries to exploit it.

2.3.1 Enumeration Techniques

- **Port Scanning**

Tools: Nmap, Zenmap, Net cat and Angry IP Scanner

The above-mentioned tools are used for network discovery and the services running for a security audit. They can be used for a single or multiple network.

Whois: It is used to gather information about the target like IP Range, host provider, email etc.

Traceroute: It is used to find the path from source to destination

- **Email Enumeration**

Harvester: It is used to enumerate emails in that domain by using search engines, social media sites etc.

- **Port Enumeration**

NSE Scripts and Metasploit tools are used for scanning, enumerating, exploitation and post-exploitation. Generally used for service and version enumeration and finding exploit for it.

2.3.2 Tools

FTP: Checks for anonymous login and used for connecting to a remote host using FTP.

SSH: It is used to connect to a remote host using ssh.

Telnet: It is used to connect to a remote host using telnet.

SMB Client: It is used to connect to a remote system for SMB service.

MYSQL: It is used to connect to the remote database

HTTP & HTTPS: SSL scan is used to test for the weak ciphers and vulnerable SSL version

Nikto: It scans web servers for dangerous files/CGIs, outdated server software and other problems like server configuration items such as the presence of multiple index files, HTTP server options.

DIRB: DirBuster is a multi-threaded java application designed to brute force directories and files names on web/application servers. We do have other tools as well for this which are 'gobuster, dirsearch and bfac'

Automated Scanners

Nessus: Nessus is a remote security scanning tool, which scans a computer and raises an alert if it discovers any vulnerabilities that malicious hackers could use to gain access to any computer you have connected to a

network. It does this by running over 1200 checks on a given computer, testing to see if any of these attacks could be used to break into the computer or otherwise harm it.

OPENVAS: OpenVAS is a full-featured vulnerability scanner. Its capabilities include unauthenticated testing, authenticated testing, various high-level and low-level Internet and industrial protocols, performance tuning for large-scale scans and a powerful internal programming language to implement any type of vulnerability test.

The scanner is accompanied by a vulnerability tests feed with a long history and daily updates. This Greenbone Community Feed includes more than 50,000 vulnerability tests

Metasploit: Metasploit is an open source framework used for almost all the phases in pen testing. It has a wide range of modules having different payloads according to different operating system. It helps in exploitation & post-exploitation phase and handling victim's session.

2.4 Scanning

This is the phase where the attacker will interact with the target with an aim to identify the vulnerabilities. An attacker will send probes to the target and records the response of the target to various inputs. This phase includes - scanning the network with various scanning tools, identification of open share drives, open FTP portals, services that are running, and much more. In case of a web application, the scanning part can be either dynamic or static. In static scanning, the application code is scanned by either a YTool or an expert application vulnerability analyst. The aim is to identify the vulnerable functions, libraries and logic implemented. In dynamic analysis, the tester will pass various inputs to the application and record the responses; various vulnerabilities like injection, cross-site scripting, remote code execution can be identified in this phase.

2.5 Gaining Access

Once the vulnerabilities have been identified, the next step is to exploit the vulnerabilities with an aim to gain access to the target. The target can be a system, firewall, secured zone or server. Be aware that not all vulnerabilities will lead you to this stage. You need to identify the ones that are exploitable enough to provide you with access to the target.

2.6 Maintaining Access

Ensure that the access is maintained i.e., persistence. This is required to ensure that the access is maintained even if the system is rebooted, reset or modified. This kind of persistence is used by attackers who live in the system and gain knowledge about them over a period, and when the environment is suitable, they exploit.

2.7 Exploitation

With a map of all possible vulnerabilities and entry points, the penetration tester begins to test the exploits on the vulnerabilities found within your network, applications, and data. The goal for the penetration tester is to

see exactly how far they can get into the environment, identify high-value targets, and avoid any detection. The penetration tester will only go as far as determined by the guidelines agreed as art of the project scope.

The below given are the step by step approach followed by Entersoft

1. Identifying the vulnerability and version of the service/daemons through enumeration. (Refer to the section 2.3.1 of Enumeration)

2. Focusing only on the version specific exploits in order to achieve the desired output.

3. Comparing and evaluating the relevant readymade available public exploits for the identified vulnerability. (OR) Writing a customized payload with the available theoretical information of this exploit.

4. Either using an exploit framework for ex: Metasploit or writing a shell script/PowerShell shall be used as a technique to execute the exploit on the vulnerability.

5. This exploitation technique shall be an iterative process observed thoroughly until we reach a logical and technical conclusion.

6. If successful the impact and likelihood will be into an consider before reporting this to the customer.

2.8 Risk Analysis & Reporting

Document how vulnerabilities are exploited as well as explain the techniques and tactics used to obtain access to high-value targets along with proof of concept (PoC). We will determine the value of the compromised systems and any value associated with the sensitive data captured to define the severity using the Common Vulnerability Scoring System (CVSS) V3.

2.9 Patch Assistance

Within the reports shared to the client we do deliver a sample code for fixing the identified bugs.

2.10 Revalidation

A validation test will be performed once after the client fixes the reported bugs and certify the network to be free from vulnerabilities. (certificate remains valid unless there are no changes been made to the tested network).

Reference Links:
https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers
http://www.tcpipguide.com/free/t_toc.htm
https://en.wikipedia.org/wiki/OSI_mode

https://en.wikipedia.org/wiki/IPv4

For Tools

https://www.ssllabs.com/ssltest/
https://www.lookout.net/test/clickjack.html (Used for checking Clickjacking)
https://securityheaders.com/ (Used for checking misconfigured secure response headers)