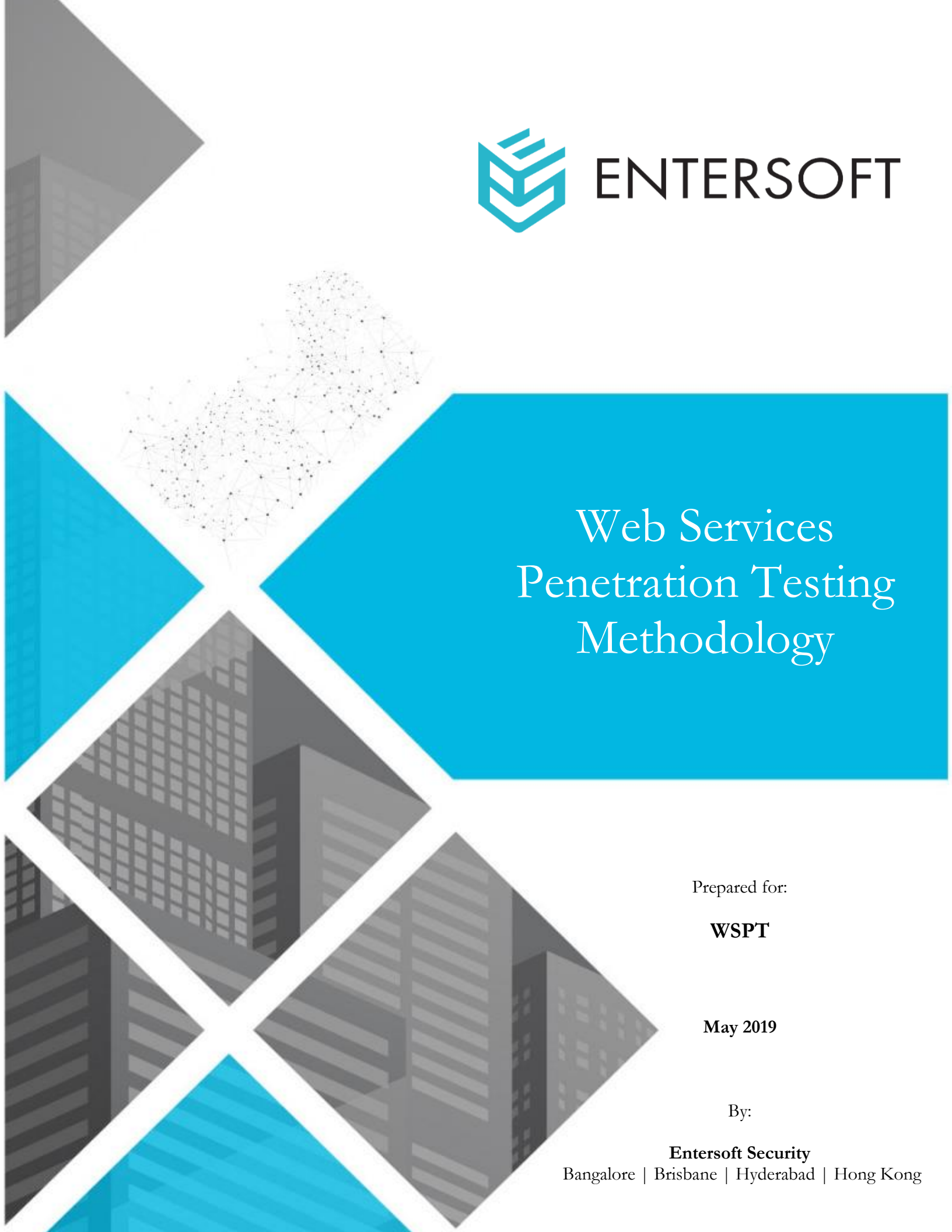

ENTERSOFT

Web Services Penetration Testing Methodology

Prepared for:

WSPT

May 2019

By:

Entersoft Security
Bangalore | Brisbane | Hyderabad | Hong Kong

Contents

1 Introduction....3
1.1 Scope....3
1.2 Standards and Guidelines....3
1.3 Risk Calculation and Classification....4
2 Types of Testing....6
2.1 White Box....6
2.2 Grey Box....6
2.3 Black Box....6
3 Methodology....7
3.1 Information Gathering....7
3.2 Scanning....7
3.3 Vulnerability Assessment....8
3.4 Exploitation....8
3.5 Reporting....8
3.6 Patch Assistance....8
3.7 Revalidation....9
4 Tools Used....9

Revision History & Version Control

Revision History & Version Control			
Release Number	**Date**	**Author**	**Comments/Details**
1.0	06.05.2019	Raju & Sagar	Final Draft
Reviewed by		Rony Konda	

1 Introduction

This report document hereby describes the methodology for Web Services Penetration Testing.

This methodology helps in addressing the below

- Determines the security posture of Web Services.
- Identify the vulnerabilities in Web Services.
- Aid in understanding the risks associated with Web Services.

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

Web Services
Presales Questionnair

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

Web Services
Onboarding Question

1.2 Standards and Guidelines

- OWASP Top 10
- REST Security Cheat sheet

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 Types of Testing

This section gives information on the types of testing.

2.1 White Box

The internal structure/design/login credentials of the Web API's being tested are shared with the tester. The tester using the below methodology determines the appropriate outputs.

Full Knowledge / Internals Fully Known

2.2 Grey Box

It is a combination of White and Black box testing wherein the internal structure is partially known to the tester.

Some Knowledge / Internals Relevant to Testing are Known

2.3 Black Box

Here the internal structure/design/credentials of the Web API's being tested are not known to the tester.

Zero Knowledge / Internals Not Known

3 Methodology

This section represents the structural procedure followed by the team to determine the security posture

- Information Gathering
- Scanning
- Vulnerability Assessment
- Exploitation
- Reporting
- Patch Assistance
- Revalidation

3.1 Information Gathering

The idea of this phase is to collect as much information as we can about web services. As part of the information gathering phase, we enumerate the API calls by accessing the application and also search for API documentation which might be available on the Internet for black box testing to understand the functionality of each endpoint. For white box testing we gather all the required materials and documentation to prepare a comprehensive test plan.

- Search engine queries to gather data about the personnel, systems, or technologies of the client.
- Domain name searches, WHOIS lookups, and reverse DNS to get subdomains, people's names, and data about the attack surface.
- Social Engineering to find out positions, technologies, email addresses
- Internet foot-printing looking for email addresses, social accounts, names, positions

3.2 Scanning

We will configure all the endpoints in automated scanning tools ex: ReadyAPI or Burp suite and initiate the scanning. After the scans are completed, we would sanitize the report to check if the reported vulnerabilities are false positives or false negatives.

3.3 Vulnerability Assessment

In this phase we prepare the relevant test cases from the information which has been gathered during the recon phase are co-related with OWASP guidelines. We will follow OWASP TOP 10 and OWASP Rest Security Cheat Sheet standards.

As per the OWASP guidelines we will focus on the below areas.

- Broken Authentication
- Broken Access Control
- Injections
- Input Validation
- Rate limiting

3.4 Exploitation

After the completion of the vulnerability assessment phase, we will map all possible vulnerabilities and entry points, we begin to test the exploits found within the client's network, applications, and data. The goal is to see exactly how far an attacker can get into the client's environment, identify high-value targets, and avoid any detection.

After the exploitation phase is complete, we document the methods used to gain access to the client organization's valuable information.

3.5 Reporting

In this phase we will report all the identifies vulnerabilities with detailed explanation along with proof of concept (PoC). We will define the severity using the Common Vulnerability Scoring System (CVSS) V3.

3.6 Patch Assistance

Within the reports shared to the client we do deliver a sample code for fixing the identified bugs which are in line with the language on which the application is built.

3.7 Revalidation

A Validation test will be performed once after the client fixes the reported bugs and certify the application as free from OWASP Top 10 vulnerabilities (validity of the certificate remains valid unless there are no code changes).

4 Tools Used

Automated Static Analysis

- Burp Suite tool
- ReadyAPI
- Drib
- Nikto
- Sqlmap
- Nmap

Manual Testing Tools

- Postman rest Client (REST)
- SOAPUI(SOAP)
- Burp Suite

Manual Testing Tools

- Postman rest Client (REST)
- SOAPUI(SOAP)
- Burp Suite