



# \$ Penetration Testing Methodology

- > Seow Chun Yong
- > Ensign InfoSecurity

# \$ whoami

- > **Seow** Chun Yong
- > Lead Penetration Tester @ Ensign InfoSecurity
- > SUTD Engineering Product Dev > Nuclear Engineering > Cybersecurity
- > Penetration Testing, Red Teaming, Vulnerability Research
- > Capture-The-Flag Organizer + Player
- > Car Hacking + Bug Bounty Programmes
- > ...





# \$ Security Testing in Software Development

- > Before code can go live, several functional and non-functional tests have to be performed to catch and fix any bugs that may be discovered before users start using the application.
- > Security Tests are tests that help identify security bugs/vulnerabilities.



# \$ ls -al Security\_Tests

- > Vulnerability Assessment
- > Penetration Testing
- > Red Teaming



# \$ man Pen\_Testing

- > Penetration Testing is testing where the tester assumes the role of a malicious actor to attempt to exploit vulnerabilities and weaknesses in a system.
- > Findings are remediated before the system can be pushed into Production.

# \$ diff Vuln\_Assessment Pen\_Testing

## > Vulnerability Assessment vs Penetration Testing

> A Vulnerability Assessment is a **non-intrusive test** that reports possible vulnerabilities with **a certain degree of confidence**.

Vulnerability Assessments do not typically validate findings, and can lead to **several false positives**.

> A Penetration Test on the other hand will attempt to exploit discovered vulnerabilities, and **assess the impact of these vulnerabilities** in the context of the system or application being tested. **It is not expected for a Penetration Test to have many false positives.**

# \$ diff Pen\_Testing Red\_Teaming

## > Penetration Testing vs Red Teaming

> A Penetration Test tests **the security of the application, system, or network**. It is concerned with how well the code and configurations are to ensure that it is not vulnerable to exploitation.

> A Red Team Exercise tests **the organization's ability to respond to an attack**. The Red Team assesses how well an organization is able to detect and respond to a simulated breach or attack, as well as how well they are able to resume normal business operations following such an attack.



# \$ ls -al Pen\_Testing/

- > Infrastructure

  - > Network

  - > Host Configuration

- > Application

  - > Web

  - > Mobile

  - > Thick Client

  - > API

  - > OT



# \$ ls -al Pen\_Testing/methodologies

- > Open Web Application Security Project (OWASP)
  - > Web Security Testing Guide (WSTG)
  - > Mobile Security Testing Guide (MSTG)
  - > Firmware Security Testing Guide (FSTG)
- > Penetration Testing Execution Standard (PTES)
- > PCI Penetration Testing Guide

[https://owasp.org/www-project-web-security-testing-guide/latest/3-The\\_OWASP\\_Testing\\_Framework/1-Penetration\\_Testing\\_Methodologies](https://owasp.org/www-project-web-security-testing-guide/latest/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies)

# \$ man Pen\_Testing/methodologies/PTES

- > Penetration Testing Execution Standard (PTES)
  - > Pre-engagement Interactions
  - > Intelligence Gathering
  - > Threat Modeling
  - > Vulnerability Analysis
  - > Exploitation
  - > Post Exploitation
  - > Reporting

[http://www.pentest-standard.org/index.php/PTES\\_Technical\\_Guidelines](http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines)

# \$ cat Pen\_Testing/methodologies/PTES/1

## > Pre-engagement Interactions

- > Also known as scoping

- > Scoping outlines the logistics of the test, expectations, legal implications, objectives, and goals that the client would like to achieve.

- > Questions to consider:

- > What are the system(s) in scope? (E.g. IP addresses, URLs)

- > Staging/UAT/Production environment?

- > Internet/Intranet?

- > Tests to be done during office hours/non-office hours?

- > Black/Grey/White Box Testing?

- > If credentials are provided, how many test accounts are provided?

- > For each test account, what permissions do they have?

# \$ cat Pen\_Testing/methodologies/PTES/2

## > Intelligence Gathering

> Data → Information → Intelligence

> Passive vs Active Scanning

> OSINT (i.e. Google-Fu)

> Questions to consider:

> What can I conclude about the system based on the information it is telling me?

> Is there any information about versions?

> Can I infer anything based on how the application behaves?

> Do I notice any highly customized functionality?



```
$ cat Pen_Testing/methodologies/PTES/3
```

- > Threat Modeling

- > Identifying and prioritizing threats to a system.

- > A Threat Model includes:

- > Attack surface of system

- > Privileges required to interact with various parts of the data flow

- > Security controls implemented

[https://cheatsheetseries.owasp.org/cheatsheets/Threat\\_Modeling\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html)



```
$ cat Pen_Testing/methodologies/PTES/3
```

- > Threat Modeling

- > Questions to consider:

- > Are my assumptions for my threat model reasonably valid?

- > How much of the scope does my threat model cover?

- > What are the test cases that would simulate such a threat actor in my threat model?

[https://cheatsheetseries.owasp.org/cheatsheets/Threat\\_Modeling\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html)



```
$ cat Pen_Testing/methodologies/PTES/4
```

- > Vulnerability Assessment

- > Identifying vulnerabilities within the system(s) in scope

- > Automated discovery

- > Known CVEs

- > Simple checks

- > Manual discovery

- > Anomalous / Unexpected behavior

- > Errors

- > Information disclosure



# \$ cat Pen\_Testing/methodologies/PTES/4

## > Vulnerability Assessment

### > Questions to consider:

- > What are the kinds of vulnerabilities I should be looking for?
- > Do my test cases cover the search for such vulnerabilities?
- > How much testing do I need to do for an engagement?
- > How much time should I spend searching for vulnerabilities?
- > What if the vulnerabilities are blind (i.e. they do not return a response to the tester)?



# \$ cat Pen\_Testing/methodologies/PTES/5

## > Exploitation

- > Successfully exploit identified vulnerabilities to assess **impact** and **severity** of vulnerabilities.
- > Exploitation performed by using Proof-of-Concept (PoC) exploits (as compared to fully-weaponized exploits).
- > PoC exploits perform the minimum requirement to prove that a vulnerability is exploitable. It should not cause damage to the system being tested.
  - > Staging/UAT: Most PoC exploits are acceptable.
  - > Production: Some PoC exploits are not recommended.



```
$ cat Pen_Testing/methodologies/PTES/5
```

- > Exploitation

- > Metasploit Framework (MSF)

- > Popular exploitation framework that facilitates exploitation of vulnerabilities

- > msfconsole (interactive) vs. msfvenom (command line)

<https://www.offensive-security.com/metasploit-unleashed/exploit-development/>

# \$ cat Pen\_Testing/methodologies/PTES/5

## > Exploitation

### > Questions to consider:

- > What counts as sufficient proof that the system is vulnerable?
- > What was agreed with the customer on the scope?
- > What evidence do I need to collect?
- > Do I need more information to be able to exploit a vulnerability?
- > How long should I try to exploit one vulnerability?
- > If I cannot exploit a vulnerability, does that prove that the system is not vulnerable?
- > Are there ways to bypass some of the security controls in place so that my exploit will work?



```
$ cat Pen_Testing/methodologies/PTES/6
```

> **Post Exploitation**

- > Comprises of activities to be performed post-compromise.
- > Highly dependent on the scope of the engagement.
- > Focus is on proving **impact in the context of the system**.
  - > Confidentiality: What kind of confidential information can be obtained?
  - > Integrity: What permissions do you have post-exploitation?
  - > Availability: What services can you deny users from making further use of?

# \$ cat Pen\_Testing/methodologies/PTES/7

## > Reporting

- > The thing that customers are paying you for.
- > High-level non-technical summary + Detailed technical explanation of findings
- > Contains the following sections:
  - > Executive Summary
  - > Scope of Assessment
  - > Assumptions and Limitations
  - > Summary of Findings
  - > Detailed Findings

# \$ cat Pen\_Testing/methodologies/PTES/7

## > Reporting

### > Executive Summary

- > Who (customer), What (type of test), Where (environment), When (dates that test was performed)
- > How many findings?
- > Any important things to note?

### > Scope of Assessment

- > Systems/IP addresses/domains/URLs in scope
- > Credentials/Accounts used + their permissions
- > Explicit out-of-scope list (E.g. Singpass, Corppass)

### > Assumptions and Limitations

- > E.g. Application in UAT is an accurate representation of the application in Production
- > E.g. Results of the tests are accurate at the time when the tests were conducted and are on a best effort basis in identifying vulnerabilities.

# \$ cat Pen\_Testing/methodologies/PTES/7

## > Reporting

### > Summary of Findings

- > Table of all findings ordered by severity scores (E.g. CVSS), from highest to lowest scores.

### > Detailed Findings

- > Includes details about the finding such that a suitably-competent technical person would be able to reproduce the finding with a high degree of confidence.

- > Usually contains screenshots as evidence.

- > Also includes:

- > Severity Rating

- > Implications

- > Recommendations

- > References