TechMagic

# Penetration Testing Plan

This plan is based on best practices, PTES [1], OWASP web application security testing guide [2], Penetration testing methodologies [3], OWASP mobile application security testing guide [5].

# Pre-engagement Operations

## Scoping Meeting

In many cases, the scoping meeting will occur after the contract has been signed. Situations do occur wherein many of the scope-related topics can be discussed before contract signing, but they are few and far between. For those situations, it is recommended that a non-disclosure agreement be signed before any in-depth scoping discussions occur.

Additionally, the countries, provinces, and states in which the target environments operate must be identified. Laws vary from region to region and the testing may be impacted by these laws. For instance, countries belonging to the European Union are well known to have very stringent laws surrounding the privacy of individuals, which can significantly change how a social engineering engagement would be executed. Below are questions that will help to prepare for the penetration test. Please note that some of them may be not relevant to the chosen type of penetration test (Black Box, Gray Box, White Box).

### Web Application Penetration Test

1. How many web applications are being assessed?
2. How many login systems are being assessed?
3. How many static pages are being assessed? (approximate)
4. How many dynamic pages are being assessed? (approximate)
5. Will the source code be made readily available?
6. Can we use brute force attacks to test the lock-out mechanism?
7. Will there be any kind of documentation? - *If yes, what kind of documentation*?
8. Will static analysis be performed on this application?
9. Does the client want fuzzing performed against this application?
10. Does the client want role-based testing performed against this application?
11. Does the client want credentialed scans of web applications performed?
12. Who will be defined as a contact person for the pentesting team?
13. Will the team have a dedicated environment for testing?

14. Are there any endpoints that could cause any harm to the tested environment in case of misuse (e.g. endpoints that send emails and should not be scanned with automated tools; endpoints that could increase monthly infrastructure costs in case of increased usage, etc.)?

## Mobile Application Penetration Test

1. Are there dedicated applications for iOS and Android?
2. Can we perform dynamic analysis of the application only on the Android OS?
3. How many login systems are being assessed?
4. How many views are being assessed?
5. Will the source code be made readily available?
6. Can we use brute force attacks to test the lock-out mechanism?
7. Are there any limitations regarding the supported versions of OS (both Android and iOS)?
8. Will there be any kind of documentation? - *If yes, what kind of documentation?*
9. Will static analysis be performed on this application?
10. Does the client want fuzzing performed against this application?
11. Does the client want role-based testing performed against this application?
12. Does the client want credentialed scans of applications performed?
13. Do you have detections for a rooted device or emulator in place?

## Network Penetration Test

1. Are there any limitations on used tools?
2. How many systems are being assessed? (approximate)
3. Is there an Active Directory implemented?
4. Will there be any kind of documentation? - If yes, what kind of documentation?
5. Are there any systems that should not be tested or have some restrictions?
6. Should we perform privilege escalation on the exploited hosts?
7. Is it allowed to conduct brute force attacks?

## Questions for Business Unit Managers

1. Is the manager aware that a test is about to be performed?
2. What is the main datum that would create the greatest risk to the organization if exposed, corrupted, or deleted?
3. Are testing and validation procedures to verify that business applications are functioning properly in place?
4. Will the testers have access to the Quality Assurance testing procedures from when the application was first developed?
5. Are Disaster Recovery Procedures in place for the application data?

## Questions for Systems Administrators

1. Are there any systems which could be characterized as fragile? (systems with tendencies to crash, older operating systems, or which are unpatched)
2. Are there systems on the network which the client does not own, that may require additional approval to test?
3. Are Change Management procedures in place?
4. What is the mean time to repair systems outages?
5. Is any system monitoring software in place?
6. What are the most critical servers and applications?
7. Are backups tested on a regular basis?
8. When was the last time the backups were restored?

## Incident Reporting Process

Discussing the organization’s current incident response capabilities is important to do before an engagement for several reasons. Part of a penetration test is not only testing the security an organization has in place, but also their incident response capabilities.

If an entire engagement can be completed without the target’s internal security teams ever noticing, a major gap in security posture has been identified. It is also important to ensure that before testing begins, someone at the target organization is aware of when the tests are being conducted so the incident response team does not start to call every member of upper management in the middle of the night because they thought they were under attack or compromised.

## Permission to Test

One of the most important documents which need to be obtained for a penetration test is the Permission to Test document. This document states the scope and contains a signature which acknowledges awareness of the activities of the testers. Further, it should clearly state that testing can lead to system instability and all due care will be given by the tester not to crash systems in the process. However, because testing can lead to instability, the customer shall not hold the tester liable for any system instability or crashes. It is critical that testing does not begin until the customer signs this document.

In addition, some service providers require advance notice and/or separate permission before testing their systems. For example, Amazon has an online request form that must be completed, and the request must be approved before scanning any hosts on their cloud. If this is required, it should be part of the document.

## Legal Considerations

Some activities common in penetration tests may violate local laws. For this reason, it is advised to check the legality of common pentest tasks in the location where the work is to be performed. For example, any VOIP calls captured in the course of the penetration test may be considered wiretapping in some areas. Also, we need to obtain a signed document that contains all law rules, customer agreement for pentest and agreement from third-party services vendors if involved.

# Information gathering

## OSINT

Open Source Intelligence (OSINT) takes three forms; Passive, Semi-passive, and Active.

### Passive Information Gathering

Passive Information Gathering is generally only useful if there is a very clear requirement that the information-gathering activities never be detected by the target. This type of profiling is technically difficult to perform as we are never sending any traffic to the target organization neither from one of our hosts or “anonymous” hosts or services across the Internet. This means we can only use and gather archived or stored information. As such, this information can be outdated or incorrect as we are limited to results gathered from a third party.

### Semi-passive Information Gathering

The goal of semi-passive information gathering is to profile the target with methods that would appear like normal Internet traffic and behavior. We query only the published name servers for information, we aren’t performing in-depth reverse lookups or brute force DNS requests, we aren’t searching for “unpublished” servers or directories. We aren’t running network-level portscans or crawlers and we are only looking at metadata in published documents and files; not actively seeking hidden content. The key here is not to draw attention to our activities. Post mortem the target may be able to go back and discover the reconnaissance activities but they shouldn’t be able to attribute the activity back to anyone.

### Active Information Gathering

Active information gathering should be detected by the target as suspicious or malicious behavior. During this stage we are actively mapping network infrastructure (think full port scans nmap –p1-65535), actively enumerating and/or vulnerability scanning the open services, we are actively searching for unpublished directories, files, and servers. Most of this activity falls into your typically “reconnaissance” or “scanning” activities for your standard pentest.

## Port Scanning

Port scanning techniques will vary based on the amount of time available for the test, and the need to be stealthy. If there is zero knowledge of the systems, a fast ping scan can be used to identify systems. In addition, a quick scan without ping verification (-PN in nmap) should be run to detect the most common ports available. Once this is complete, a more comprehensive scan can be run. Some testers check for only open TCP ports, make sure to check UDP as well. The http://nmap.org/nmap_doc.html document details port scan types. Nmap ("Network Mapper'') is the de facto standard for network auditing/scanning. Nmap runs on both Linux and Windows.

Nmap has dozens of options available. Since this section is dealing with port scanning, we will focus on the commands required to perform this task. It is important to note that the commands utilized depend mainly on the time and number of hosts being scanned. The more hosts or less time that you have to perform these tasks, the less that we will interrogate the host. This will become evident as we continue to discuss the options. IPv6 should also be tested.

## Banner Grabbing

Banner Grabbing is an enumeration technique used to glean information about computer systems on a network and the services running its open ports. Banner grabbing is used to identify a network, the version of applications and operating system that the target host is running.

Banner grabbing is usually performed on Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP); ports 80, 21, and 25 respectively. Tools commonly used to perform banner grabbing are Telnet, nmap, and Netcat.

## SNMP Sweeps

SNMP sweeps are performed too as they offer tons of information about a specific system. The SNMP protocol is a stateless, datagram oriented protocol. Unfortunately SNMP servers don't respond to requests with invalid community strings and the underlying UDP protocol does not reliably report closed UDP ports. This means that "no response" from a probed IP address can mean either of the following:

- machine unreachable
- SNMP server not running
- invalid community string
- the response datagram has not yet arrived

## Zone Transfers

DNS zone transfer, also known as AXFR, is a type of DNS transaction. It is a mechanism designed to replicate the databases containing the DNS data across a set of DNS servers. Zone transfer comes in two flavors, full (AXFR) and incremental (IXFR). There are numerous tools available to test the ability to perform a DNS zone transfer. Tools commonly used to perform zone transfers are host, dig and nmap.

## SMTP Bounce Back

SMTP bounce back, also called a Non-Delivery Report/Receipt (NDR), a (failed) Delivery Status Notification (DSN) message, a Non-Delivery Notification (NDN) or simply a bounce, is an automated electronic mail message from a mail system informing the sender of another message about a delivery problem. This can be used to assist an attacker in fingerprint the SMTP server as SMTP server information, including software and versions, may be included in a bounce message.

This can be done by simply creating a bogus address within the target's domain. For instance, asDFADSF_garbage_address@target.com could be used to test target.com. Gmail provides full access to the headers, making it an easy choice for testers.

## DNS Discovery

DNS discovery can be performed by looking at the WHOIS records for the domain's authoritative nameserver. Additionally, variations of the main domain name should be checked, and the website should be checked for references to other domains which could be under the target's control.

## Forward/Reverse DNS

Reverse DNS can be used to obtain valid server names in use within an organization. There is a caveat that it must have a PTR (reverse) DNS record for it to resolve a name from a provided IP address. If it does resolve then the results are returned. This is usually performed by testing the server with various IP addresses to see if it returns any results.

## DNS Bruteforce

Since DNS is used to map IP addresses to hostnames, and vice versa we will want to see if it is insecurely configured. We will seek to use DNS to reveal additional information about the client. One of the most serious misconfigurations involving DNS is allowing Internet users to perform a DNS zone transfer. There are several tools that we can use to enumerate DNS to not only check for the ability to perform zone transfers, but to potentially discover additional host names that are not commonly known.

## Virtual Host Detection & Enumeration

Web servers often host multiple "virtual" hosts to consolidate functionality on a single server. If multiple servers point to the same DNS address, they may be hosted on the same server. Tools such as MSN search can be used to map an ip address to a set of virtual hosts.

## Establish External Target List

Once the activities above have been completed, a list of users, emails, domains, applications, hosts and services should be compiled.

## Mapping Versions

Version checking is a quick way to identify application information. To some extent, versions of services can be fingerprinted using nmap, and versions of web applications can often be gathered by looking at the source of an arbitrary page.

## Identifying Patch Levels

To identify the patch level of services internally, consider using software which will interrogate the system for differences between versions. Credentials may be used for this phase of the penetration test, provided the client has acquiesced. Vulnerability scanners are particularly effective at identifying patch levels remotely, without credentials.

## Identify Lockout Threshold

Identifying the lockout threshold of an authentication service will allow you to ensure that your brute force attacks do not intentionally lock out valid users during your testing. Identify all disparate authentication services in the environment, and test a single, innocuous account for lockout. Often 5 - 10 tries of a valid account is enough to determine if the service will lock users out.

## Tools

- OSINT framework [https://osintframework.com/]
- Sublist3r [https://github.com/aboul3la/Sublist3r]
- nmap [https://nmap.org]
- hping [https://www.kali.org/tools/hping3/]
- Spider Foot [https://www.spiderfoot.net/]
- Spider Foot HX [https://www.spiderfoot.net/hx/]
- DNS dumpster [https://dnsdumpster.com/]
- Shodan [https://www.shodan.io/]
- dirb [https://www.kali.org/tools/dirb/]
- OWASP ZAP Spider [https://www.zaproxy.org/docs/desktop/addons/spider/]
- Maltego [https://www.paterva.com/web7/]
- crt.sh [https://crt.sh/]
- Wappalyzer [https://www.wappalyzer.com/]

Approximate time for phase: ~2 days (depending on the type of pentest).

# Vulnerability Testing

Once we've collected all the required information regarding the target, we can use it to look for known vulnerabilities in the identified components. Vulnerability testing is the process of discovering flaws in systems and applications which can be leveraged by an attacker. These flaws can range anywhere from host and service misconfiguration, or insecure application design. Although the process used to look for flaws varies and is highly dependent on the particular component being tested, some key representatives take part in the process.

When conducting vulnerability analysis of any type the tester should properly scope the testing for applicable depth and breadth to meet the goals and/or requirements of the desired outcome. Depth values can include such things as the location of an assessment tool, authentication requirements, etc. For example; in some cases it may be the goal of the test to validate mitigation is in place and working and the vulnerability is not accessible; while in other instances the goal may be to test every applicable variable with authenticated access in an effort to discover all applicable vulnerabilities. Whatever your scope, the testing should be tailored to meet the depth requirements to reach your goals. Depth of testing should always be validated to ensure the results of the assessment meet the expectation (i.e. did all the machines authenticate, etc.). In addition to depth, breadth must also be taken into consideration when conducting vulnerability testing. Breadth values can include things such as target networks, segments, hosts, application, inventories, etc. At its simplest element, your testing may be to find all the vulnerabilities on a host system; while in other instances you may need to find all the vulnerabilities on hosts within a given inventory or boundary. Additionally, breadth of testing should always be validated to ensure you have met your testing scope (i.e. was every machine in the inventory alive at the time of scanning? If not, why).

### Tools

- OWASP ZAP [https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project]
- Burp Suite [https://portswigger.net/burp]
- Arachni [http://www.arachni-scanner.com/]
- nuclei [https://github.com/projectdiscovery/nuclei]
- SQLmap [https://github.com/sqlmapproject/sqlmap]
- OpenVAS [https://greenbone.github.io/docs/latest/]
- Pacu [https://github.com/RhinoSecurityLabs/pacu]
- weirdAAL [https://github.com/carnal0wnage/weirdAAL]
- CSP evaluator [https://github.com/google/csp-evaluator]

- Snyk [https://snyk.io/]
- nikto [https://github.com/sullo/nikto]
- SonarQube [https://www.sonarqube.org/]
- Semgrep [https://semgrep.dev/]

Other tools that can be used during this phase:

- faraday [https://github.com/infobyte/faraday]
- XSSStrike [https://github.com/s0md3v/XSStrike]
- Hydra [https://sectools.org/tool/hydra/]

Approximate time for this phase: ~6 days (depending on the type of pentest).

# Exploitation of the detected vulnerabilities

After the active scanning phase and review of the detected vulnerabilities, penetration testers investigate an application manually with any proxy interceptor such as Burp Suite. Mentioned in previous step tools cannot cover some advanced cases (arbitrary file upload with token generation, advanced XSS exploitation through different parts of apps, writing exploits for fuzzing in order to provoke Internal server errors or Gateway Timeout errors). Also, manual review will help to better understand application functionality, cover critical, commonly used, rarely used by users areas. For example, the login page is used by all users but administrator panel functionality isn't accessible to all. We can check different broken access control cases and try to exploit them for retrieving sensitive information.

Approximate time for this phase: ~2 days (depending on the type of pentest).

# Report Generation Phase

## Executive Summary

This section will communicate to the reader the specific goals of the Penetration Test and the high level findings of the testing exercise. The intended audience will be those who are in charge of the oversight and strategic vision of the security program as well as any members of the organization which may be impacted by the identified/confirmed threats.  The executive summary contains the following sections:

### Introduction

The introduction section of the technical report is intended to be an initial inventory of:

- Personnel involved in the testing from both the Client and Penetration Testing Team
- Contact information
- Assets involved in testing
- Objectives of Test
- Scope of Test
- Strength of Test
- Approach
- Threat/Grading Structure

This section should be a reference for the specific resources involved in the testing and the overall technical scope of the test.

(Example: (CLIENT) tasked <Pentester> with performing an internal/external vulnerability assessment and penetration testing of specific systems located in (logical area or physical location). These systems have been identified as (risk ranking) and contain (data classification level) data which, if accessed inappropriately, could cause material harm to (Client). In an effort to test (CLIENT's) ability to defend against direct and indirect attack, <Pentester> executed a comprehensive network vulnerability scan, Vulnerability conformation( <-insert attack types agreed upon->) exploitation of weakened services, client side attacks, browser side attacks (etc) The purpose of this assessment was to verify the effectiveness of the security controls put in place by (CLIENT) to secure business-critical information. This report represents the findings from the assessment and the associated remediation recommendations to help CLIENT strengthen its security posture.

## Findings Severity Ratings

This section contains descriptions of possible severities that can be assigned to the detected vulnerabilities. Listed severities are also taken into account during the estimation of overall application/network risk level.

| Severity | CVSS V3 Score Range | Definition |
|---|---|---|
| **Critical** | 9.0-10.0 | Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately. |
| **High** | 7.0-8.9 | Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible. |
| **Moderate** | 4.0-6.9 | Vulnerabilities exist but are not exploitable or require extra steps such as social engineering or some actions from the end-users. It is advised to form a plan of action and patch after high-priority issues have been resolved. |
| **Low** | 0.1-3.9 | Vulnerabilities in the low range typically have very little impact on an organization's business. The exploitation of such vulnerabilities usually requires local or physical system access. |
| **Informational** | N/A | No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation. |

## Scope

Contains details of the tested systems/applications with the selected type of penetration test (Black/Grey/White Box).

| Assessment | Details |
|---|---|
| Gray Box Web Application Penetration Test | Domain: owasp-juice.shop<br>Subdomains: demo.owasp-juice.shop<br>admin.owasp-juice.shop |

## Overall Risks of the Targets in Scope

This area will include details of the general risk level assigned to the assessed targets, as well as brief description of detected exploitation flows.

## Vulnerabilities by Impact

Short overview of the overall severity of detected vulnerabilities together with details on relationship between hosts/applications and detected number of vulnerabilities for each of them.

## Successful Attacks by type

Information about the detected vulnerabilities, divided by categories (Injections, Headers security, Misconfigurations, Unpatched software, .etc). This data helps to better understand the most successful attacks and their reasons in scope of the entire system/application.

## Vulnerabilities by cause

Here you will find a list of vulnerabilities that will be divided into categories (Injections, Headers security, Misconfigurations, Unpatched software, .etc) and grouped by affected hosts/applications. Similar to the previous section but here we can review the most vulnerable systems/applications and the count of different types of attacks that were successful.

## Recommendations

This section contains a list of short-/long-term recommendations that will help to properly prioritize the team’s efforts in fixing detected issues. Recommendations are sorted by priority that helps to concentrate attention on the most critical mitigation measures.

## Vulnerability Report

This section will communicate to the reader the technical details of the detected issues, step by step guide to reproduce them and related proofs that confirm successful exploitation of the issue. Exploitation or Vulnerability confirmation is the act of triggering the vulnerabilities identified in the previous sections to gain a specified level of access to the target asset. This section should review, in detail, all of the steps taken to confirm the defined vulnerability as well as the following:

- Exploitation Timeline
- Targets selected for Exploitation
- Exploitation Activities
- Attacks conducted
- Attacks conducted Level of access Granted + escalation path

Detections are grouped by hosts/subdomains/domains and sorted by priority in descending order (from Critical to Informational/Low).

## Remediation Report

Final overview of the test. It is suggested that this section echo portions of the overall test as well as support the growth of the CLIENT security posture. It contains detailed step by step guidance for the team on how to fix the detected issues and measures that would help to avoid them in future.

Approximate estimation for this phase: ~3 days (depends on the penetration test type).

# Total Estimation (approx)

- **Black Box: 40 hours**
- **Gray Box: 80 hours**
- **White Box: 120 hours**

# Terminology

Below you can find a provided list of terms which are used in this document (full list of terms and additional links can be found in [4]).

| | |
|---|---|
| **Access control** | Controlling who has access to a computer or online service and the information it stores. |
| **Asset** | Something of value to a person, business or organization. |
| **Authentication** | The process to verify that someone is who they claim to be when they try to access a computer or online service. |
| **Backing up** | To make a copy of data stored on a computer or server to lessen the potential impact of failure or loss. |
| **Bring your own device (BYOD)** | The authorised use of personally owned mobile devices such as smartphones or tablets in the workplace. |
| **Broadband** | High-speed data transmission system where the communications circuit is shared between multiple users. |
| **Business continuity management** | Preparing for and maintaining continued business operations following disruption or crisis. |
| **Certification** | Declaration that specified requirements have been met. |

| | |
|---|---|
| **Certification body** | An independent organization that provides certification services. |
| **Chargeback** | A payment card transaction where the supplier initially receives payment but the transaction is later rejected by the cardholder or the card issuing company. The supplier's account is then debited with the disputed amount. |
| **Cloud computing** | Delivery of storage or computing services from remote servers online (ie via the internet). |
| **Common text** | A structure and series of requirements defined by the International Organization for Standardization, that are being incorporated in all management system International Standards as they are revised. |
| **Data server** | A computer or program that provides other computers with access to shared files over a network. |
| **Declaration of conformity** | Confirmation issued by the supplier of a product that specified requirements have been met. |
| **DMZ** | Segment of a network where servers accessed by less trusted users are isolated. The name is derived from the term "demilitarised zone". |

| | |
|---|---|
| **Encryption** | The transformation of data to hide its information content. |
| **Ethernet** | Communications architecture for wired local area networks based uponIEEE 802.3 standards. |
| **Firewall** | Hardware or software designed to prevent unauthorised access to a computer or network from another computer or network. |
| **Gap analysis** | The comparison of actual performance against expected or required performance. |
| **Hacker** | Someone who violates computer security for malicious reasons, kudos or personal gain. |
| **Hard disk** | The permanent storage medium within a computer used to store programs and data. |
| **Identification** | The process of recognising a particular user of a computer or online service. |
| **Infrastructure-as-a-service (IaaS)** | Provision of computing infrastructure (such as server or storage capacity) as a remotely provided service accessed online (ie via the internet). |
| **Inspection certificate** | A declaration issued by an interested party that specified requirements have been met. |

| | |
|---|---|
| **Instant messaging** | Chat conversations between two or more people via typing on computers or portable devices. |
| **Internet service provider (ISP)** | Company that provides access to the internet and related services. |
| **Intrusion detection system (IDS)** | Program or device used to detect that an attacker is or has attempted unauthorised access to computer resources. |
| **Intrusion prevention system (IPS)** | Intrusion detection system that also blocks unauthorised access when detected. |
| **‘Just in time’ manufacturing** | Manufacturing to meet an immediate requirement, not in surplus or in advance of need. |
| **Keyboard logger** | A virus or physical device that logs keystrokes to secretly capture private information such as passwords or credit card details. |
| **Leased circuit** | Communications link between two locations used exclusively by one organization. In modern communications, dedicated bandwidth on a shared link reserved for that user. |
| **Local area network (LAN)** | Communications network linking multiple computers within a defined location such as an office building. |

| | |
|---|---|
| **Macro virus** | Malware (ie malicious software) that uses the macro capabilities of common applications such as spreadsheets and word processors to infect data. |
| **Malware** | Software intended to infiltrate and damage or disable computers. Shortened form of malicious software. |
| **Management system** | A set of processes used by an organization to meet policies and objectives for that organization. |
| **Network firewall** | Device that controls traffic to and from a network. |
| **Outsourcing** | Obtaining services by using someone else’s resources. |
| **Passing off** | Making false representation that goods or services are those of another business. |
| **Password** | A secret series of characters used to authenticate a person’s identity. |
| **Personal firewall** | Software running on a PC that controls network traffic to and from that computer. |
| **Personal information** | Personal data relating to an identifiable living individual. |

| | |
|---|---|
| **Phishing** | Method used by criminals to try to obtain financial or other confidential information (including usernames and passwords) from internet users, usually by sending an email that looks as though it has been sent by a legitimate organization (often a bank). The email usually contains a link to a fake website that looks authentic. |
| **Platform-as-a-service (PaaS)** | The provision of remote infrastructure allowing the development and deployment of new software applications over the internet. |
| **Portable device** | A small, easily transportable computing device such as a smartphone, laptop or tablet computer. |
| **Proxy server** | Server that acts as an intermediary between users and others servers, validating user requests. |
| **Restore** | The recovery of data following computer failure or loss. |
| **Risk** | Something that could cause an organization not to meet one of its objectives. |
| **Risk assessment** | The process of identifying, analysing and evaluating risk. |
| **Router** | Device that directs messages within or between networks. |

| | |
|---|---|
| **Screen scraper** | A virus or physical device that logs information sent to a visual display to capture private or personal information. |
| **Security control** | Something that modifies or reduces one or more security risks. |
| **Security information and event management (SIEM)** | Process in which network information is aggregated, sorted and correlated to detect suspicious activities. |
| **Security perimeter** | A well-defined boundary within which security controls are enforced. |
| **Server** | Computer that provides data or services to other computers over a network. |
| **Smartphone** | A mobile phone built on a mobile computing platform that offers more advanced computing ability and connectivity than a standard mobile phone. |
| **Software-as-a-service (SaaS)** | The delivery of software applications remotely by a provider over the internet; perhaps through a web interface. |
| **Spyware** | Malware that passes information about a computer user’s activities to an external party. |

|  |  |
|---|---|
| **Supply chain** | A set of organisations with linked resources and processes involved in the production of a product. |
| **Tablet** | An ultra-portable, touch screen computer that shares much of the functionality and operating system of smartphones, but generally has greater computing power. |
| **Threat** | Something that could cause harm to a system or organization. |
| **Threat actor** | A person who performs a cyber attack or causes an accident. |
| **Two-factor authentication** | Obtaining evidence of identity by two independent means, such as knowing a password and successfully completing a smartcard transaction. |
| **Username** | The short name, usually meaningful in some way, associated with a particular computer user. |
| **User account** | The record of a user kept by a computer to control their access to files and programs. |
| **Virtual private network (VPN)** | Link(s) between computers or local area networks across different locations using a wide area network that cannot access or be accessed by other users of the wide area network. |

| | |
|---|---|
| **Virus** | Malware that is loaded onto a computer and then run without the user’s knowledge or knowledge of its full effects. |
| **Vulnerability** | A flaw or weakness that can be used to attack a system or organization. |
| **Wide area network (WAN)** | Communications network linking computers or local area networks across different locations. |
| **Wi-Fi** | Wireless local area network based uponIEEE 802.11standards. |
| **Worm** | Malware that replicates itself so it can spread to infiltrate other computers. |

# Library

1. Penetration testing execution standard: online resource. - Access link:

   http://www.pentest-standard.org/index.php/Main_Page

2. OWASP Web application security testing guide: online resource. - Access link:

   https://owasp.org/www-project-web-security-testing-guide/

3. Penetration testing methodologies: online resource. - Access link:

   https://www.owasp.org/index.php/Penetration_testing_methodologies#PCI_Penetration_testing_guide

4. Security words list: online resource. - Access link:

   https://www.bsigroup.com/en-GB/Cyber-Security/Cyber-security-for-SMEs/Glossary-of-cyber-security-terms/

5. OWASP Mobile application security testing guide: online resource. - Access link:

   https://github.com/OWASP/owasp-mastg