

## 1.1

### Importance of Security

Today, we live in the digital age, every process involves or is under the process of being involved with digitization. Data can be found everywhere, networks are being established everywhere. With this increase in our dependence on technologies, there also increases the threat to them. And thus as there is an increase in our dependence on cyberspace, there also increases the need of keeping it secure. Data privacy, network privacy, network integrity, and functioning that keeps our lives streamlined need cybersecurity to run seamlessly.

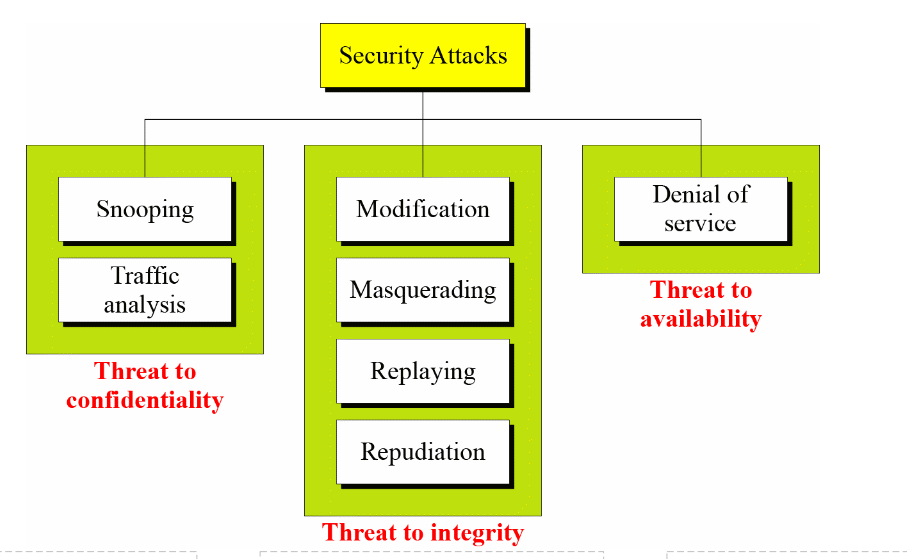
### Elements of Security

#### Accountability

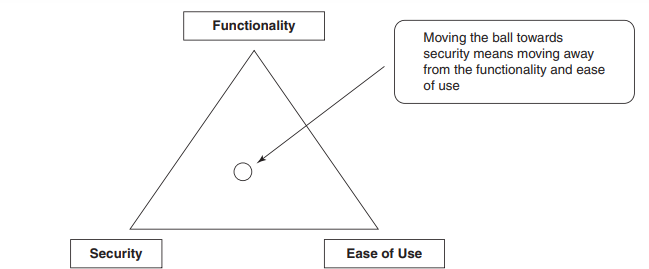
In a network environment, the system, users, and applications interact with each other. In such a case, there needs to be identification and authentication. Audit trails or log files address this thus leading to accountability.

#### Reusability

The way that a network is structured, not all resources are available to everyone. Having such access control enhances security. Here, reusability/availability plays an important role. A user or program may not reuse or manipulate objects that another user or program is currently accessing in order to prevent violation of security. Information and processes need to be accurate as accuracy and integrity of data play a very important role in creating a secure environment.



#### The Security, Functionality, and Ease of Use Triangle



### Phases of an attack

#### Reconnaissance

During the reconnaissance phase, the penetration tester (or attacker) attempts to learn as much as possible about a network. Testers use both passive reconnaissance and active network reconnaissance and discovery when gathering information on targets.

##### Passive reconnaissance

Passive reconnaissance collects information about a targeted system, network, or organization using open-source intelligence (OSINT) about DNS records, domain names, email addresses, IP addresses. Tools: Dimitry lookup, Who is loopup, Shodan

##### Active reconnaissance

Active reconnaissance involves direct interactions with the target system by using tools to detect open ports, accessible hosts, router locations, network mapping, details of operating systems, and applications.

Eg: Nmap, Nessus, Nikita

#### Scanning

The attacker uses the details gathered during reconnaissance to identify specific vulnerabilities. Scanning can be considered a logical extension of active reconnaissance.

#### Gaining Access

If a vulnerability is detected, the attacker exploits it to gain access to the system. Access can be gained locally, offline, over a LAN, or over the Internet.

A hacker’s chances of gaining access into a target system are influenced by factors such as the architecture and configuration of the target system, the skill level of the perpetrator, and the initial level of access obtained.

Eg. Stack-based buffer overflows, denial-of-service, and session hijacking. Attackers use a technique called spoofing to exploit the system by pretending to be a legitimate user or different systems. They can use this technique to send a data packet containing a bug to the target system in order to exploit a vulnerability. Packet flooding may be used to remotely stop availability of essential services. Smurf attacks attempt to cause users on a network to flood each other with data, making it appear as if everyone is attacking each other, and leaving the hacker anonymous.

#### Maintaining Access

Once access is gained, the attacker usually maintains access to fulfill the goal of the attack.

Once an attacker gains access to the target system, he or she is able to use both the system and its resources at will and can either use the system as a launchpad to scan and exploit other systems, or keep a low profile and continue exploiting the system.

Attackers, who choose to remain undetected, remove evidence of their entry and install a backdoor or a Trojan to gain repeat access. They can also install rootkits at the kernel level to gain full administrator access to the target computer. Rootkits gain access at the operating system level, while a Trojan horse gains access at the application level. Both rootkits and Trojans require users to install them locally. In Windows systems, most Trojans install themselves as a service and run as local system, which has administrative access.

Hackers can use Trojans to transfer user names, passwords, and any other information stored on the system. Organizations can use intrusion detection systems or deploy traps known as honeypots and honeynets to detect intruders.

#### Covering Tracks

The attacker tries to destroy all evidence of the attack.

Trojans such as ps or netcat are often used to erase the attacker’s activities from the system’s log files. Once the Trojans are in place, the attacker has likely gained total control of the system. By executing a script in a Trojan or rootkit, a variety of critical files are replaced with new versions, hiding the attacker in seconds.

Other techniques include steganography and tunneling. Steganography is the process of hiding data in other data, for instance, image and sound files. Tunneling takes advantage of the transmission protocol by carrying one protocol over another.

### Types of Hacker Attacks

#### Operating system attacks

Attackers look for vulnerabilities in OS such that they can exploit through vulnerabilities and gain access to the target system or network. The vulnerabilities in the OS can be open ports and services as most of the operating systems install these services and ports by default.

#### Application-level attacks

Undiscovered security vulnerabilities in applications lead to it being the entry point into a system. Eg. Error checking in applications can be very poor (or even nonexistent), which leads to buffer overflow attacks.

#### Shrink-wrap code attacks

Software developers will often use free libraries and code licensed from other sources in their programs. This means that large portions of many pieces of software will be exactly the same, and if vulnerabilities in that code are discovered, many pieces of software are at risk.

#### Misconfiguration attacks

Even systems that are otherwise very secure can be hacked if they are not configured correctly. System administrators need to be careful when configuring systems, and always know what is running. It is important to create a simple but usable configuration, removing all unnecessary services and software.

### Hacktivism and Ethical Hackers

Hacktivism is when hackers break into government or corporate computer systems as an act of protest. Hacktivists use hacking to increase awareness of their social or political agendas, as well as themselves, in both the online and offline arenas. Common hacktivist targets include government agencies, multinational corporations, or any other entity that they perceive as a threat. It remains a fact, however, that gaining unauthorized access is a crime, irrespective of their intentions.

#### Hacker Classes

##### Black hats

Use their computer skills for illegal or malicious purposes. This category of hacker is often involved with criminal activities and is sought by law enforcement agencies.

##### White hats

Use their hacking ability for defensive purposes. White hats include security analysts who are knowledgeable about hacking countermeasures.

##### Gray hats

Believe in full disclosure. They believe that information is better out in the open than kept in secret, and the average person will make good use of that information rather than abuse it.

##### Suicide hackers

Hacktivists that are willing to become martyrs for their causes. They attempt to sabotage large-scale infrastructures and are fully willing to accept any consequences of their actions.

#### Ethical Hackers

Ethical hackers are information security professionals who specialize in evaluating and defending against, threats from attackers.

##### Ethical Hacker Categories

(This is different from exercise types i.e red, blue, purple and white)

###### Former black hats

Reformed attackers. They are well informed about security due to their past actions in attempting to defeat it, and retain access to hacker networks in order to keep up with new developments; however, they may pass along sensitive information to those hacker networks, knowingly or accidentally, thereby putting their clients at risk.

###### White hats

These are independent security consultants working either individually or as a group. They have not been on the attacking side, so they don’t have the same experience as the former black hats, but that does not mean that they can’t be just as knowledgeable. Most ethical hackers are white hats.

###### Consulting firms

With the increasing demand for third-party security evaluations, consulting firms are becoming more common. These firms can boast impressive talent and credentials, but due diligence must be done in checking up on these firms before hiring them.

## 1.2

### Penetration testing stages

#### Planning Phase

In the planning phase, rules are identified, management approval is finalized and documented, and testing goals are set. The planning phase sets the groundwork for a successful penetration test. No actual testing occurs in this phase.

#### Pre attack Phase

This phase is focused on gathering as much information as possible about the target to be attacked. This can be noninvasive or invasive.

#### Attack Phase

The information gathered in the preattack phase forms the basis of the attack strategy. Before deciding on the attack strategy, the tester may choose to carry out an invasive information-gathering process such as scanning.

#### Postattack Phase

This is a crucial part of the testing process, as the tester needs to restore the network to its original state. This will involve cleanup of testing processes and removal of vulnerabilities created (not those that existed originally), exploits crafted, etc.

### Penetration testing methods

#### External testing

External penetration tests target the assets of a company that are visible on the internet, e.g., the web application itself, the company website, and email and domain name servers (DNS). The goal is to gain access and extract valuable data.

#### Internal testing

In an internal test, a tester with access to an application behind its firewall simulates an attack by a malicious insider. This isn’t necessarily simulating a rogue employee. A common starting scenario can be an employee whose credentials were stolen due to a phishing attack.

#### Blind testing

In a blind test, a tester is only given the name of the enterprise that’s being targeted. This gives security personnel a real-time look into how an actual application assault would take place.

#### Double-blind testing

In a double blind test, security personnel have no prior knowledge of the simulated attack. As in the real world, they won’t have any time to shore up their defenses before an attempted breach.

#### Targeted testing

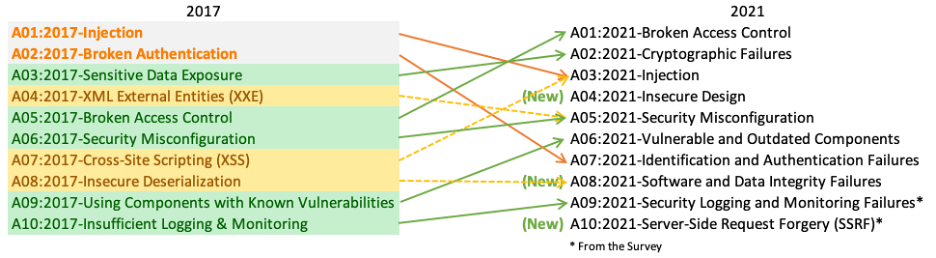
In this scenario, both the tester and security personnel work together and keep each other appraised of their movements. This is a valuable training exercise that provides a security team with real-time feedback from a hacker’s point of view.

## 1.3

### Proper and Ethical disclosure

Vulnerability disclosure is the practice of publishing information related to a security vulnerability found in software. The purpose for such a disclosure is to inform the customer of the potential risks, so that they can take actions to minimize the effects of the vulnerability. The question of whether or not to disclose a newly-found vulnerability is one of the most sensitive decisions a software provider can make.

### OWASP Top Ten



#### A01:2021-Broken Access Control

Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits.

#### A02:2021-Cryptographic Failures

Failures related to cryptography (or lack thereof). Which often leads to exposure of sensitive data.

#### A03:2021-Injection

An application is vulnerable to attack when:

User-supplied data is not validated, filtered, or sanitized by the application.

Hostile data is directly used or concatenated. The SQL or command contains the structure and malicious data in dynamic queries, commands, or stored procedures.

#### A04:2021-Insecure Design

Vulnerabilities which exist due to lack of security implementation in an application at the time of development. It denotes that the best practices for the designing an application has not been taken into consideration.

#### A05:2021-Security Misconfiguration

Security misconfigurations are security controls that are inaccurately configured or left insecure, putting your systems and data at risk. Basically, any poorly documented configuration changes, default settings, or a technical issue across any component in your endpoints could lead to a misconfiguration.

#### A06:2021-Vulnerable and Outdated Components

A software component is part of a system or application that extends the functionality of the application, such as a module, software package, or API. Component-based vulnerabilities occur when a software component is unsupported, out of date, or vulnerable to a known exploit.

#### A07:2021-Identification and Authentication Failures

Cryptography failures, session fixation, default login credentials, and brute-forcing access.

#### A08:2021-Software and Data Integrity Failures

Use of critical data or apps without verification of their identity falls under this category.

#### A09:2021-Security Logging and Monitoring Failures

There is no direct vulnerability that can arise due to these issues but in general, logging and monitoring are quite critical and their absence or failures can directly impact visibility, incident alerting, and forensics. Thus, it’s quite important to have a functional logging and monitoring system to collect logs and also give alerts if any malfunctions or errors happen, else these can go unnoticed for a long time and cause a lot more damage.

#### A10:2021-Server-Side Request Forgery

Server-side request forgery (also known as SSRF) is a web security vulnerability that allows an attacker to induce the server-side application to make HTTP requests to an arbitrary domain of the attacker's choosing.