Risk Analysis

The objective of a security program is to mitigate risks. Mitigating risks does not mean eliminating them; it means reducing them to an acceptable level. To make sure that the security controls are effectively controlling the risks in the environment, it is important to anticipate what kinds of incidents may occur. It is also needed to identify what one is trying to protect, and from whom. That's where risk analysis, threat definition, and vulnerability analysis come in.

Threat

A cyber or cybersecurity threat is a malicious act that seeks to damage data, steal data, or disrupt digital life in general. Threats can take many forms, and in order to be successful, a security strategy must be comprehensive enough to manage the most significant threats.

Threat Vector

A threat vector is a term used to describe where a threat originates and the path it takes to reach a target. An example of a threat vector is an e-mail message sent from outside the organization to an inside employee, containing an irresistible subject line along with an executable attachment that happens to be a Trojan program, which will compromise the recipient's computer if opened.

Source	Threats	Targets
Employee	Theft	Intellectual Property
Software	Corruption	Email
Software Bug	Error	Application

Different Types of Attacks

Any computer that is accessible from the Internet will be attacked. It will constantly be probed by attackers and malicious programs intending to exploit vulnerabilities. There can be different types of attacks, such as:

- Malicious Mobile Code
- Advanced Persistence Threat (APT)
- Manual Attacks

Malicious Mobile Code

There are three generally recognized variants of malicious mobile code: ***viruses, worms, and Trojans***. In addition, many *malware programs* have components that act like two or more of these types, which are called hybrid threats or mixed threats. The lifecycle of malicious mobile code looks like this:

- Find
- Exploit
- Infect
- Repeat

Computer Viruses

A virus is a self-replicating program that uses other host files or code to replicate. Most viruses infect files so that every time the host file is executed, the virus is executed too. A virus infection is simply another way of saying the virus made a copy of itself (replicated) and placed its code in the host in such a way that it will always be executed when the host is executed.

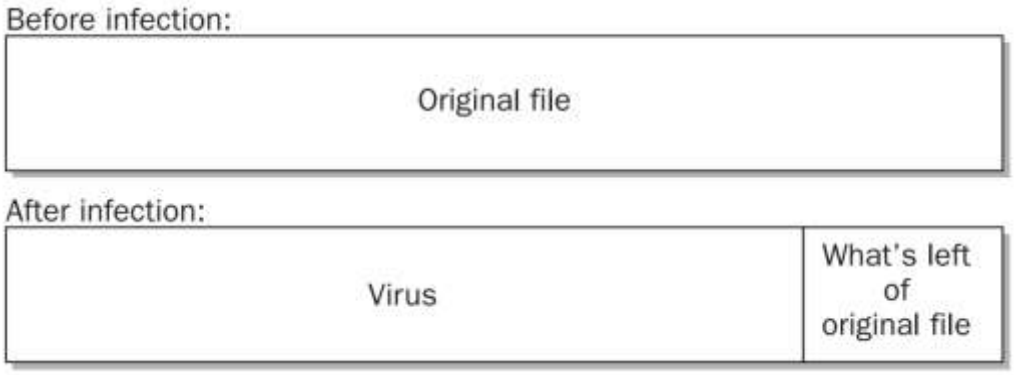
The damage routine of a virus (or really of any malware program) is called the payload. The vast majority of malicious program files do not carry a destructive payload beyond the requisite replication. This means they aren't intentionally designed by their creators to cause damage. However, their very nature requires that they modify other files and processes without appropriate authorization, and most end up causing program crashes of one type or another.

At the very least, a "harmless" virus takes up CPU cycles and storage space. Of course, payloads can be intentionally destructive, deleting files, corrupting data, copying confidential information, formatting hard drives, and removing security settings. Some viruses are devious. Many send out random files from the user's hard drive to everyone in the user's e-mail address list.

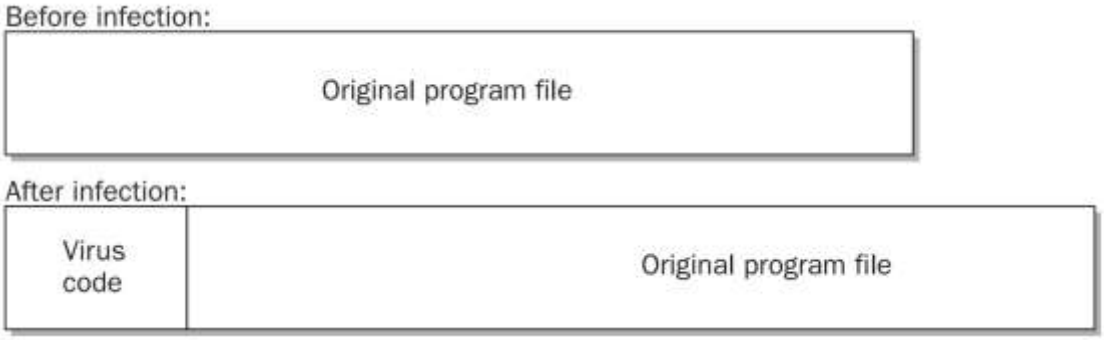
If the virus executes, does its damage, and terminates until the next time it is executed, it is known as a **nonresident virus**. A nonresident virus may, for example, look for and infect five EXE files on the hard disk and then terminate until the next time an infected file is executed. These types of viruses are easier for novice malicious coders to write. If the virus stays in memory after it is executed, it is called a **memory-resident virus**. Memory-resident viruses insert themselves as part of the operating system or application and can manipulate any file that is executed, copied, moved, or listed.

If the virus overwrites the host code with its own code, effectively destroying much of the original contents, it is called an **overwriting virus**.

If the virus inserts itself into the host code, moving the original code around so the host programming still remains and is executed after the virus code, the virus is called a **parasitic virus**. Viruses that copy themselves to the beginning of the file are called **prepending viruses**, and viruses placing themselves at the end of a file are called **appending viruses**. Viruses appearing in the middle of a host file are labeled **mid-infecting viruses**. The modified host code doesn't always have to be a file—it can be a disk boot sector or partition table, in which case the virus is called a **boot sector** or **partition table virus**.



Example of an overwriting virus



Example of a prepending parasitic virus

Computer Worms

A computer worm uses its own coding to replicate, although it may rely on the existence of other related code to do so. The key to a worm is that it does not directly modify other host code to replicate. A worm may travel the Internet trying one or more exploits to

compromise a computer, and if successful, it then writes itself to the computer and begins replicating again. An example of an Internet worm is Bugbear. Bugbear was released in June 2003, arriving as a file attachment in a bogus e-mail.

E-mail worms are a curious intersection of social engineering and automation. They appear in people's inboxes as messages and file attachments from friends, strangers, and companies. They pose as games, official patches from Microsoft, or unofficial applications found in the digital marketplace. There cannot be a computer user in the world who has not been warned multiple times against opening unexpected e-mail attachments, but often the attachments are simply irresistible.

Trojans

Trojan horse programs, or Trojans, work by posing as legitimate programs that are activated by an unsuspecting user. After execution, the Trojan may attempt to continue to pose as the other legitimate program (such as a screensaver) while doing its malicious actions in the background. Many people are infected by Trojans for months and years without realizing it.

If the Trojan simply starts its malicious actions and doesn't pretend to be a legitimate program, it's called a **direct-action Trojan**. Direct-action Trojans don't spread well because the victims notice the compromise and are unlikely, or unable, to spread the program to other unsuspecting users.

An example of a direct-action Trojan is JS.ExitW. It can be downloaded and activated when unsuspecting users browse malicious web sites. In one case, this Trojan posed as a collection of Justin Timberlake pictures and turned up in a search using Google. The link, instead of leading to the pictures, downloaded and installed the JS.ExitW Trojan. When activated, JS.ExitW installs itself in the Windows startup folder as an HTML application (.hta) that shuts down Windows. Because it is in the startup folder, this has the consequence of putting infected PCs in a never-ending loop of starts and shutdowns.

A powerful type of Trojan program called a **remote access Trojan (RAT)** is very popular in today's attacker circles. Once installed, a RAT becomes a back door into the compromised system and allows the remote attackers to do virtually anything they want to the compromised PC. **Password-stealing Trojans** look for saved passwords on the computer and email them to the hackers. Some can even steal passwords cached in the browser history. **Destructive Trojans** destroy and delete files from the computer.

Antivirus killer Trojans detect and kill the antivirus and firewall programs to give the attacker easier access to computer.

Zombie Trojans infect a host and wait for their originating attacker's commands telling them to attack other hosts. The attacker installs a series of zombie Trojans, sometimes numbering in the thousands. With one predefined command, the attacker can cause all the zombies to begin to attack another remote system with a distributed denial of service (DDoS) attack. DDoS attacks flood the intended victim computer with so much traffic, legitimate or malformed, that it becomes overutilized or locks up, denying legitimate connections.

Advanced Persistent Threat (APT)

The use of sophisticated malware for targeted cybercrime is known as advanced persistent threats (APTs). Usually targeted at businesses (especially high-tech businesses with juicy intellectual property and trade secrets desired by competitors) and governments that have political adversaries, APTs are created and directed by hostile governments and organized criminals for financial or political gain.

APTs rely on targeted attacks to achieve success. While malware and phishing attacks are not new, the APT is a new way to commit these types of attacks. The APT attacks are generally targeted towards specific organizations, often including high-level executives, to gain access to proprietary information or trade secrets.

Common APT Schemes

- **Spear Phishing**: Spear phishing is a method that targets specific individuals or groups within an organization. It is a potent a malicious tactic which uses emails, social media, instant messaging, and other platforms to get users to divulge personal information or perform actions that cause network compromise, data loss, or financial loss. Spear phishing focuses on specific targets and involve prior research. A typical spear phishing attack includes an email and attachment. The email includes information specific to the target, including the target's name and rank within the company. This social engineering tactic boosts the chances that the victim will carry out all the actions necessary for infection, including opening the email and the included attachment.
- **Watering Hole Attack**: A watering hole attack is a targeted attack designed to compromise users within a specific industry or group of users by infecting websites they typically visit and luring them to a malicious site. The end goal is to

infect the user's computer with malware and gain access to the organization's network. Watering hole attacks, also known as strategic website compromise attacks, are limited in scope as they rely on an element of luck.

- **Privilege Escalation**: Privilege escalation attacks occur when a threat actor gains access to an employee's account, bypasses the proper authorization channel, and successfully grants themselves access to data they are not supposed to have.
- **Credential Harvesting**: Credential harvesting, also known as password harvesting or username harvesting, is a form of cyberattack that involves the theft of personal or financial data such as usernames and passwords, typically carried out through phishing, malicious websites, email scams, or malware but not always. Any social engineering techniques, digital scamming, and malware may be used to steal login credentials.
- **Data Exfiltration**: Data exfiltration, also known as data extrusion or data exportation, is data theft. It is the intentional, unauthorized, covert transfer of data from a computer or other device. Data exfiltration may be conducted manually, or automated using malware.

Detection of APT

- **Abnormal Activities**: An infected system will have abnormal user account activities like multiple logins, frequent password changes, and random posts or emails. This is because the threat will try to reach out to the crucial database and will try everything possible.
- **Trojans in Abundance**: It is common to find infected components in a system when APT is trying to make its way. If one finds one's systems to be using Trojan horses (or remote access Trojan) excessively, be it can be assured that APT is there.
- **Database Defects**: The prime aim of a threat is to access the database only. If APT is present in the system, there will be sudden changes in the data access activities. For instance, more failed attempts to access databases, trying to access the large quality of data that were not accessed before, or making changes in sensitive data.
- **Suspicious Data**: The data files that a system stores should always be properly monitored. If one finds anything unusual in one's system that one cannot remember downloading or creating, it can be considered a sign of an APT attack.

Manual Attacks

While automated attacks may satisfy virus writers, typical attackers want to test their own mental wits and toolkits against a foreign computer, changing their attack plan as

the host exposes its weaknesses. They love the challenge manual hacking gives. An example of such attack is ARP poisoning.

Risk Analysis

A risk analysis needs to be a part of every security effort. It should analyze and categorize the assets that need to be protected and the risks that need to be avoided, and it should facilitate the identification and prioritization of protective elements. It can also provide a means to measure the effectiveness of the overall security architecture, by tracking those risks and their associated mitigation over time to observe trends.

- **Qualitative risk analysis** is the process of rating or scoring risk based on a person's perception of the severity and likelihood of its consequences. The goal of qualitative risk analysis is to come up with a short list of risks which need to be prioritized above others. Qualitative risk analysis is best described as a project manager's first line of defense against risks. It helps weed out potential detractors to the project's success, including risks that are unlikely to cause any severe harm to the project. By targeting the most dangerous risks first, risk analysis in project management becomes more efficient and project managers are able to allocate their time and resources more effectively.
- **Quantitative risk analysis** is the process of calculating risk based on data gathered. The goal of quantitative risk analysis is to further specify how much will the impact of the risk cost the business. This is achieved by using what's already known to predict or estimate an outcome. For data to be suitable for quantitative risk analysis, it has to have been studied for a long period of time or to have been observed in multiple situations. For example, in the past five projects, equipment type A has broken down after 7 hours of use. With this information, it can be assumed that if a project requires workers to use equipment type A for 8 hours, then it has a 100% chance of breaking down.
- **The key difference** between qualitative and quantitative risk analysis is the basis for evaluating risks. As mentioned earlier, qualitative risk analysis is based on a person's perception or judgment while quantitative risk analysis is based on verified and specific data.