Software Vulnerability
viruses, worms

Software Vulnerability

- A vulnerability can be an error in the way that user management occurs in the system, an error in the code or a flaw in how it responds to certain requests.
- Software vulnerabilities involve bugs in software.
- Bugs are coding errors that cause the system to make an unwanted action.
- All software has bugs of one form or another. Some bugs cause the system to crash, some cause connectivity to fail, some do not let a person to log in, and some cause printing not to work properly.
- Some bugs create information leakage or elevate user privileges or grant otherwise unauthorized access. These are **security vulnerabilities**.
- Operating systems are composed of software, as are web browsers, word processing programs, spreadsheets, video players, websites, and every other application.
- Even computer hardware includes a form of software called firmware.
- Networking equipment and cell phones also have software, and therefore inevitably security vulnerabilities.

Types of Software Vulnerability

- Missing data encryption
- OS command injection
- SQL injection
- Buffer overflow
- Missing authentication for critical function
- Missing authorization
- Unrestricted upload of dangerous file types
- Reliance on untrusted inputs in a security decision
- Cross-site scripting and forgery
- Download of codes without integrity checks
- Use of broken algorithms
- URL redirection to untrusted sites
- Path traversal
- Bugs
- Weak passwords
- Software that is already infected with virus

Buffer Overflow

- A buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers.
- A buffer overflow, or buffer overrun, occurs when more data is put into a fixed-length buffer than the buffer can handle.
- The extra information, which has to go somewhere, can overflow into adjacent memory space, corrupting or overwriting the data held in that space.
- This overflow usually results in a system crash, but it also creates the opportunity for an attacker to run arbitrary code or manipulate the coding errors to prompt malicious actions.
- Many programming languages are prone to buffer overflow attacks.
- The code written in Perl and JavaScript is generally not susceptible to buffer overflows.
- However, a buffer overflow in a program written in C, C++, Fortran or Assembly could allow the attacker to fully compromise the targeted system.

Buffer Overflow

Types of Buffer Overflow Attacks

- **Stack-based buffer overflows** are more common, and leverage stack memory that only exists during the execution time of a function.
- **Heap-based attacks** are harder to carry out and involve flooding the memory space allocated for a program beyond memory used for current runtime operations

Buffer overflow example

- **Bfrovrflw.c file**
- #include <stdio.h>
- #include <string.h>
- int main(void)
- { char buff[15];
- int pass = 0;
- printf("\n Enter the password\n");
- gets(buff);
- if(strcmp(buff, "thegeekstuff"))
- {
- printf ("\n Wrong Password \n"); }
- else {
- printf ("\n Correct Password \n");
- pass = 1; }
- if(pass) {

- /* Now Give root or admin rights to user*/
- printf ("\n Root privileges given to the user \n"); }
- return 0; }

Buffer overflow example

- $ ./bfrovrflw
- Enter the password : hhhhhhhhhhhhhhhhhhhh
- Wrong Password
- Root privileges given to the user

Here inspite of wrong password privileges are given to user

- The attacker supplied an input of length greater than what buffer can hold and at a particular length of input the buffer overflow so took place that it overwrote the memory of integer ‘pass’.
- So despite of a wrong password, the value of ‘pass’ became non zero and hence root privileges were granted to an attacker.

Format String

- Format string vulnerabilities are a class of bug that take advantage of an easily avoidable programmer error.
- If the programmer passes an attacker-controlled buffer as an argument to a printf (or any of the related functions, including sprintf, fprintf, etc), the attacker can perform writes to arbitrary memory addresses.
- The following program contains such an error:

Format String

```
// A simple C program with format
// string vulnerability
#include<stdio.h>

int main(int argc, char** argv)
{
    char buffer[100];
    strncpy(buffer, argv[1], 100);

    // We are passing command line
    // argument to printf
    printf(buffer);

    return 0;
}
```

Format String

- $ ./a.out "AAAA%p %p %p %p %p %p %p %p %p %p"
- Output is
- AAAA0xffffdde8 0x64 0xf7ec1289 0xffffdbef 0xffffdbee (nil) 0xffffdcd4 0xffffdc74 (nil) 0x41414141
- It prints next addresses on the stack

SQL Injection

- SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.
- It generally allows an attacker to view data that they are not normally able to retrieve.
- This might include data belonging to other users, or any other data that the application itself is able to access.
- In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior.
- In some situations, an attacker can escalate an SQL injection attack to compromise the underlying server or other back-end infrastructure, or perform a denial-of-service attack.

Cross site scripting

- Cross-site Scripting (XSS) is a client-side code injection attack
- The attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application.
- The actual attack occurs when the victim visits the web page or web application that executes the malicious code.
- The web page or web application becomes a vehicle to deliver the malicious script to the user's browser. Vulnerable vehicles that are commonly used for Cross-site Scripting attacks are forums, message boards, and web pages that allow comments.
- A web page or web application is vulnerable to XSS if it uses unsanitized user input in the output that it generates.
- This user input must then be parsed by the victim's browser. XSS attacks are possible in VBScript, ActiveX, Flash, and even CSS.
- However, they are most common in JavaScript, primarily because JavaScript is fundamental to most browsing experiences.

What is malware

- Malware, or malicious software, is any program or file that is harmful to a computer user.
- Malware includes computer viruses, worms, Trojan horses and spyware.
- These malicious programs can perform a variety of functions,
  - Stealing
  - Encrypting
  - Deleting sensitive data
  - Altering or hijacking core computing functions
  - Monitoring users computer activity without their permission.

Causes of malware

- Drive-by download—Unintended download of computer software from the Internet
- Unsolicited email —Unwanted attachments or embedded links in electronic mail
- Physical media—Integrated or removable media such as USB drives
- Self propagation—Ability of malware to move itself from computer to computer or network to network, thus spreading on its own

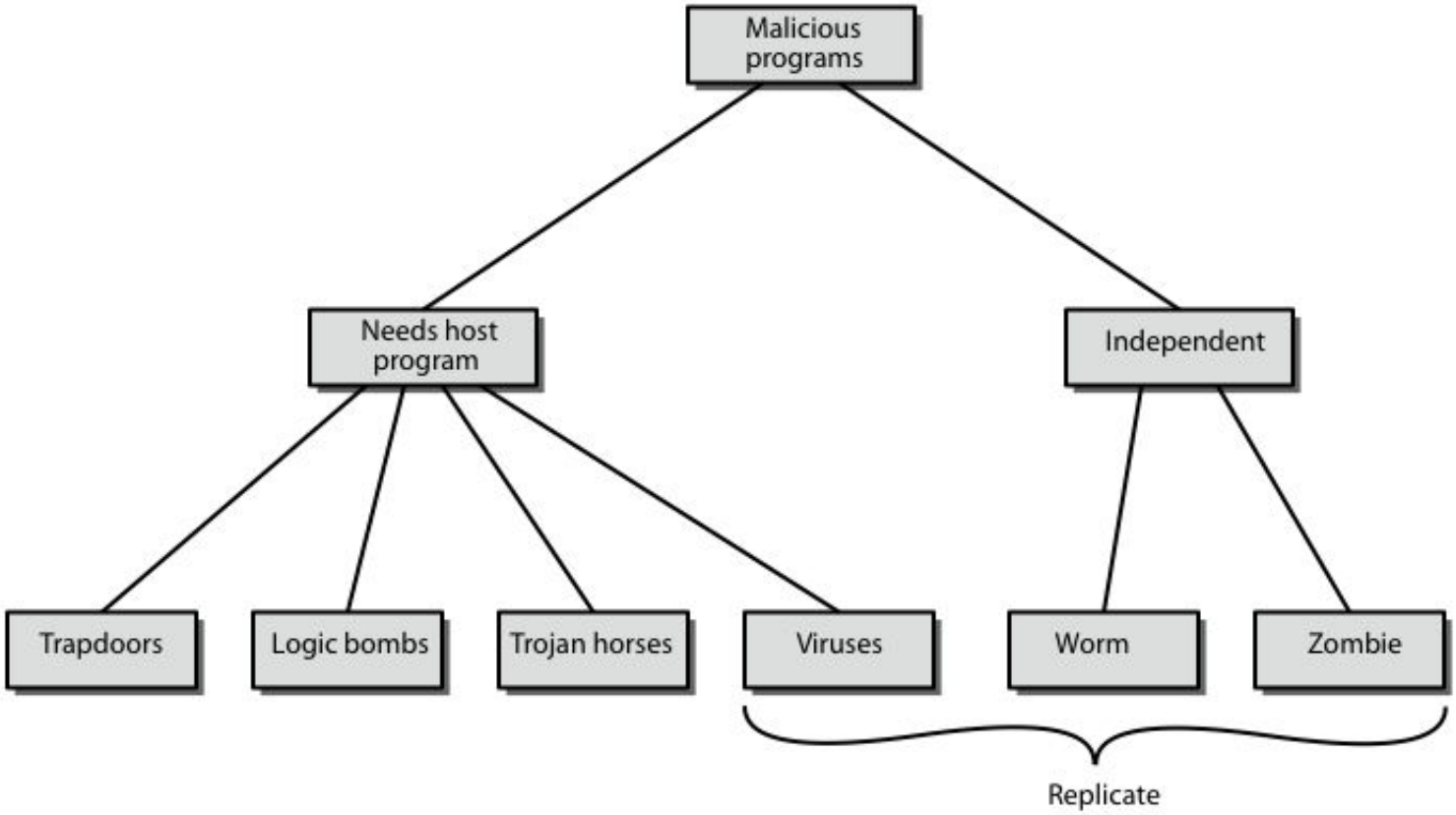
Malicious Software

- **Malicious software can be divided into two categories:** those that **need a host program, and those that are independent.**
- The former are essentially fragments of programs that cannot exist independently of some actual application program, utility, or system program.

  Examples: Viruses, logic bombs, and backdoors
- The latter **are self-contained programs** that can be scheduled and run by the operating system.

  Examples: Worms and zombie programs

Malicious Software



Malicious Software

Name	Description
Virus	Attaches itself to a program and propagates copies of itself to other programs
Worm	Program that propagates copies of itself to other computers
Logic bomb	Triggers action when condition occurs
Trojan horse	Program that contains unexpected additional functionality
Backdoor (trapdoor)	Program modification that allows unauthorized access to functionality
Exploits	Code specific to a single vulnerability or set of vulnerabilities
Downloaders	Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.
Auto-rooter	Malicious hacker tools used to break into new machines remotely
Kit (virus generator)	Set of tools for generating new viruses automatically
Spammer programs	Used to send large volumes of unwanted e-mail
Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial of service (DoS) attack
Keyloggers	Captures keystrokes on a compromised system
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access
Zombie	Program activated on an infected machine that is activated to launch attacks on other machines

Symptoms of malware

- Your computer slows down.
- A tidal wave of annoying ads that shouldn't be there washes over your screen. Unexpected pop-up ads are a typical sign of a malware infection. (adware)
- Your system repeatedly crashes, freezes, or displays a BSOD (Blue Screen of Death), which can occur on Windows systems after encountering a fatal error.
- You notice a mysterious loss of disk space.
- There's a weird increase in your system's Internet activity.
- Usage of your system resources is abnormally high and your computer's fan starts whirling away at full speed—signs of malware activity taking up system resources in the background.
- Your browser's homepage changes without your permission. Similarly, links you click send you to an unwanted web destination.
- New toolbars, extensions, or plugins unexpectedly populate your browser.
- Your antivirus product stops working and you cannot update it, leaving you unprotected against the sneaky malware that disabled it.
- Then there's the painfully obvious, intentionally non-stealthy malware attack. This famously happens with[ransomware](), which announces itself, tells you it has your data, and demands a ransom to return your files.

Virus

- A computer virus is a type of malware that propagates by inserting a copy of itself into and becoming part of another program.
- It spreads from one computer to another, leaving infections as it travels.
- Viruses can range in severity from causing mildly annoying effects to damaging data or software and causing denial-of-service (DoS) conditions.
- Almost all viruses are attached to an **executable file** which means the virus may exist on a system but will not be active or able to spread until a user runs or opens the malicious host file or program.
- When the host code is executed, the viral code is executed as well.
- Normally, the host program keeps functioning after it is infected by the virus.
- Some viruses overwrite other programs with copies of themselves, which destroys the host program altogether.
- Viruses spread when the software or document they are attached to is transferred from one computer to another using the network, a disk, file sharing, or infected email attachments.

Virus

- A computer virus is a piece of software that can “infect” other programs by modifying them
- The modification includes
  - injecting the original program with a routine to make copies of the virus program, which can then go on to infect other programs.
- Similar to Biological viruses are tiny scraps of genetic code—DNA or RNA—that can take over the machinery of a living cell and trick it into makin thousands of flawless replicas of the original virus.
- A computer virus carries in its instructional code the recipe for making perfect copies of itself.
- The typical virus becomes embedded in a program on a computer. Then, whenever the infected computer comes into contact with an uninfected piece of software, a fresh copy of the virus passes into the new program.
- Thus, the infection can be spread from computer to computer by unsuspecting users who either swap disks or send programs to one another over a network. In a network environment, the ability to access applications and system services on other computers provides a perfect culture for the spread of a virus.

Viruses

- a piece of self-replicating code attached to some other code
- attaches itself to another program and executes secretly when the host program is executed.
- propagates itself & carries a payload
  - carries code to make copies of itself
  - as well as code to perform some covert task

Virus

A computer virus has three parts

- **Infection mechanism:** The means by which a virus spreads, enabling it to replicate. The mechanism is also referred to as the **infection vector**.
- **Trigger:** The event or condition that determines when the payload is activated or delivered.
- **Payload:** What the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity.

Virus Operation

- virus phases:
  - dormant – waiting on trigger event
  - propagation – replicating to programs/disks
  - triggering – by event to execute payload
  - execution – of payload
- details usually machine/OS specific
  - exploiting features/weaknesses

A simple virus

```
     program V :=

{goto main;
     1234567;

     subroutine infect-executable :=
          {loop:
          file := get-random-executable-file;
          if (first-line-of-file = 1234567)
             then goto loop
             else prepend V to file; }

     subroutine do-damage :=
          {whatever damage is to be done}

     subroutine trigger-pulled :=
          {return true if some condition holds}

main:    main-program :=
          {infect-executable;
          if trigger-pulled then do-damage;
          goto next;}
next:

}
```

Types of Viruses

Classification on basis of how they attack

- parasitic virus

  -attaches itself to executable files and replicates

- memory-resident virus

  -lodges in the main memory and infects every program that executes.

- boot sector virus

  -infects a boot record and spreads when the system is booted from the disk

Types of Viruses…

Classification on the basis of concealment strategy

- **Encrypted virus**
  - A portion of the virus creates a random encryption key and encrypts the remainder of the virus. The key is stored with the virus. When an infected program is invoked, the virus uses the stored random key to decrypt the virus. When the virus replicates, a different random key is selected. Because the bulk of the virus is encrypted with a different key for each instance, there is no constant bit pattern to observe.
- **Stealth virus**
  - A form of virus explicitly designed to hide itself from detection by antivirus software. Thus, the entire virus, not just a payload is hidden.

Types of Viruses…

Classification on the basis of concealment strategy

- **Polymorphic virus**
  - A virus that mutates with every infection, making detection by the "signature" of the virus impossible.
- **Metamorphic virus**
  - As with a polymorphic virus, a metamorphic virus mutates with every infection. The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection. Metamorphic viruses may change their behavior as well as their appearance.

Email Virus

- spread using email with attachment containing a macro virus
- triggered when user opens attachment
- worse even when mail viewed by using scripting features in mail agent
- propagates very quickly
- usually targeted at Microsoft Outlook mail agent & Word/Excel documents

Worm

- Computer worms are similar to viruses in that they replicate functional copies of themselves
- In contrast to viruses, which require the spreading of an infected host file, worms are standalone software and do not require a host program or human help to propagate.
- A worm enters a computer through a vulnerability in the system and takes advantage of file-transport or information-transport features on the system, allowing it to travel unaided.
- More advanced worms leverage encryption, wipers, and ransomware technologies to harm their targets.

Worms

- replicating but not infecting program (does not attach itself to a program)
- typically spreads over a network
  - Morris Internet Worm in 1988
- using users distributed privileges or by exploiting system vulnerabilities worms perform unwanted functions
- widely used by hackers to create **zombie PC's**, subsequently used for further attacks, esp DoS
- major issue is lack of security of permanently connected systems, esp PC's

Worm Operation

- worm has phases like those of viruses:
  - dormant
  - propagation
    - search for other systems to infect
    - establish connection to target remote system
    - replicate self onto remote system
  - triggering
  - execution

Morris Worm

- best known classic worm
- released by Robert Morris in 1988
- targeted Unix systems
- using several propagation techniques
  - simple password cracking of local pw file
  - exploit bug in finger daemon
  - exploit debug trapdoor in sendmail daemon
- if any attack succeeds then replicated self

Malicious Software Cont..

- Malicious software is software that is intentionally included or inserted in a system for a harmful purpose.
- **A virus** is a piece of software that can "infect" other programs by modifying them; **the modification includes a copy of the virus program, which can then go on to infect other programs.**
- **A worm is a program that can replicate itself and send copies from computer to computer across network connections.** Upon arrival, the worm may be activated to replicate and propagate again. In addition to propagation, the worm usually performs some unwanted function.
- A **denial of service (DoS) attack** is an attempt to prevent legitimate users of a service from using that service.
- A **distributed denial of service attack** is launched from multiple coordinated sources.

Backdoor or Trapdoor

- secret entry point into a program
- allows those who know access bypassing usual security procedures
- have been commonly used by developers
- a threat when left in production programs allowing exploited by attackers
- very hard to block in O/S

Logic Bomb

- one of oldest types of malicious software
- code embedded in legitimate program
- activated when specified conditions met
  - E.g., presence/absence of some file
  - particular date/time
  - particular user
- when triggered typically damage system
  - modify/delete files/disks, halt machine, etc.

Trojan Horse

- program with hidden side-effects
- which is usually superficially attractive
  - E.g., game, s/w upgrade, etc.
- when run performs some additional tasks
  - allows attacker to indirectly gain access they do not have directly
- often used to propagate a virus/worm or install a backdoor or simply to destroy data
- Mail the password file.

Zombie

- program which secretly takes over another networked computer
- then uses it to indirectly launch attacks

(difficult to trace zombie's creator)

- often used to launch distributed denial of service (DDoS) attacks
- exploits known flaws in network systems

Virus Countermeasures

- best countermeasure is prevention

(do not allow a virus to get into the system in the first place.)

- but in general not possible
- hence need to do one or more of:
  - **detection** - of viruses in infected system
  - **identification** - of specific infecting virus
  - **removal** - restoring system to clean state