# Information Security CS3002 (Sections BDS-7A/B) Lecture 13

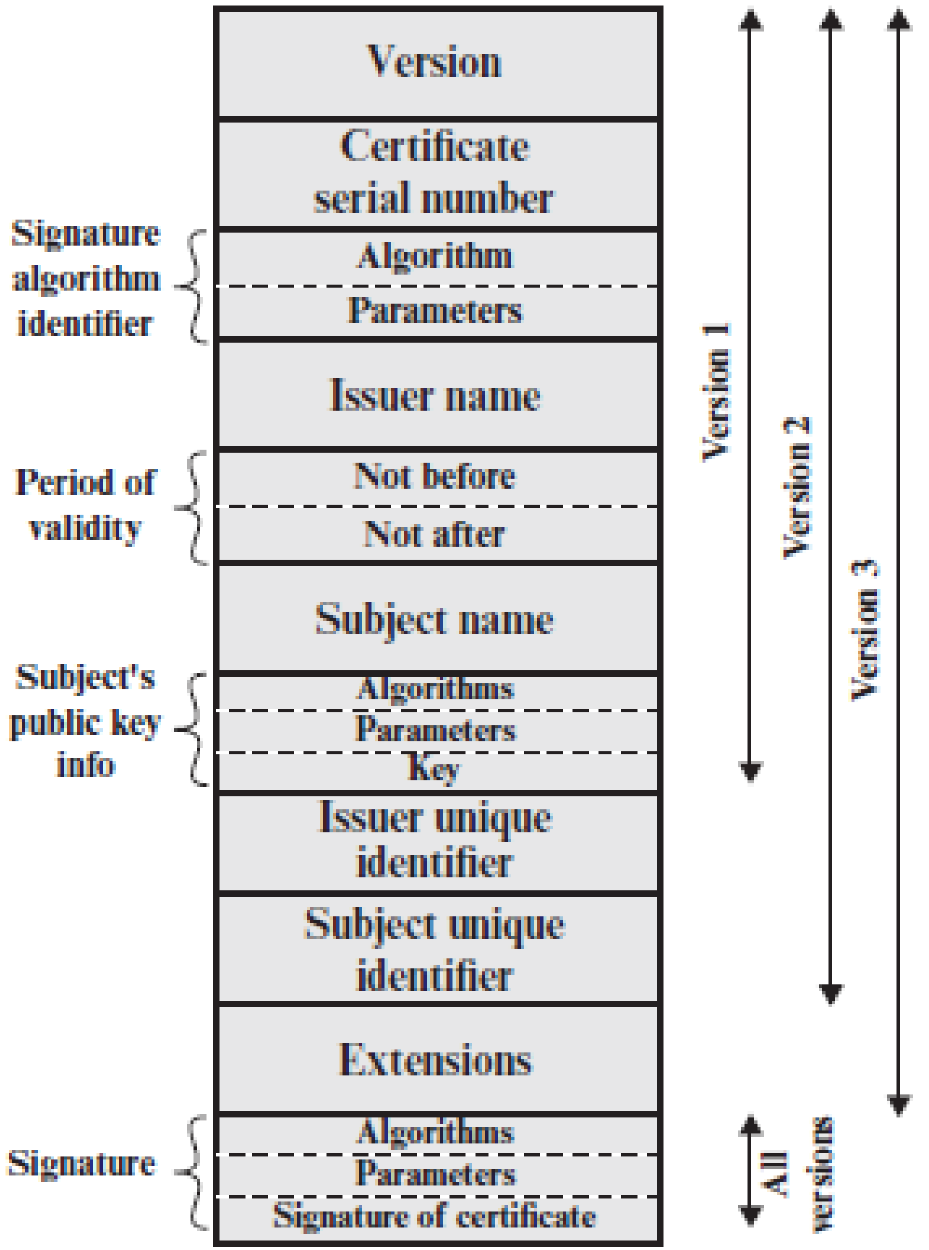
Instructor: Dr. Syed Mohammad Irteza

Assistant Professor, Department of Computer Science

02 October, 2024

# Initial Topics

- Public Key Infrastructure (PKI)
- Elements of PKI



Version

Certificate serial number

Signature algorithm identifier

Algorithm

Parameters

Issuer name

Period of validity

Not before

Not after

Subject name

Subject's public key info

Algorithms

Parameters

Key

Issuer unique identifier

Subject unique identifier

Extensions

Signature

Algorithms

Parameters

Signature of certificate

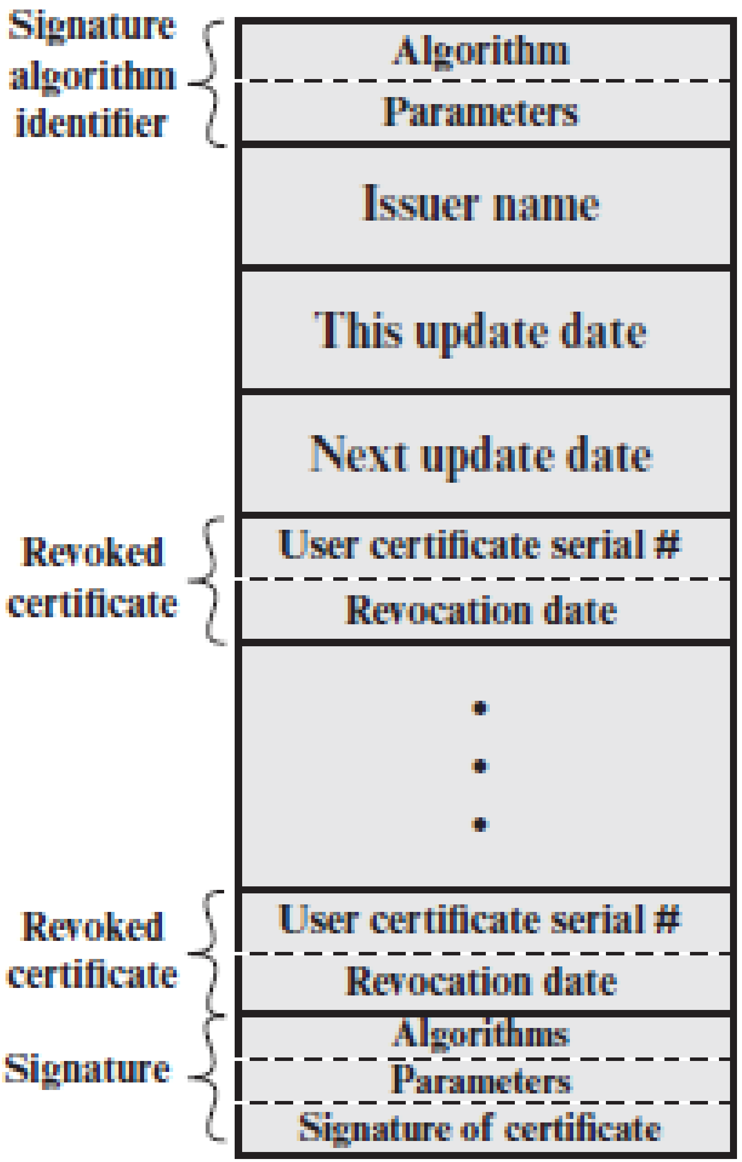
Version 1

Version 2

Version 3

All versions

(a) X.509 certificate



Signature algorithm identifier: Algorithm, Parameters

Issuer name

This update date

Next update date

Revoked certificate: User certificate serial #, Revocation date

•
•
•

Revoked certificate: User certificate serial #, Revocation date

Signature: Algorithms, Parameters, Signature of certificate

(b) Certificate revocation list

# Public Key Infrastructure

14.5 (Cryptography & Network Security, Stallings)

- RFC 4949 (Internet Security Glossary) defines public-key infrastructure (PKI) as the set of *hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates* based on asymmetric cryptography.
- The principal objective for developing a PKI is to enable secure, convenient, and efficient *acquisition of public keys*

# IETF PKIX Working Group

- The Internet Engineering Task Force (IETF) *Public Key Infrastructure X.509 (PKIX) working group* has been the driving force behind setting up a formal (and generic) model based on X.509 that is suitable for deploying a *certificate-based architecture on the Internet*.

# Elements of the PKIX Model

- **End entity**: A generic term used to denote end users, devices (e.g., servers, routers), or any other entity that can be identified in the subject field of a public-key certificate.
  - End entities typically consume and/or support PKI-related services.
- **Certification authority** (CA): The issuer of certificates and (usually) certificate revocation lists (CRLs).
  - It may also support a variety of administrative functions, although these are often delegated to one or more Registration Authorities.

# Elements of the PKIX Model (cont’d)

- **Registration authority** (RA): An optional component that can assume a number of administrative functions from the CA.
  - The RA is often associated with the end entity registration process but can assist in a number of other areas as well.
- **CRL issuer**: An optional component that a CA can delegate to publish CRLs.
- **Repository**: A generic term used to denote any method for storing certificates and CRLs so that they can be retrieved by end entities

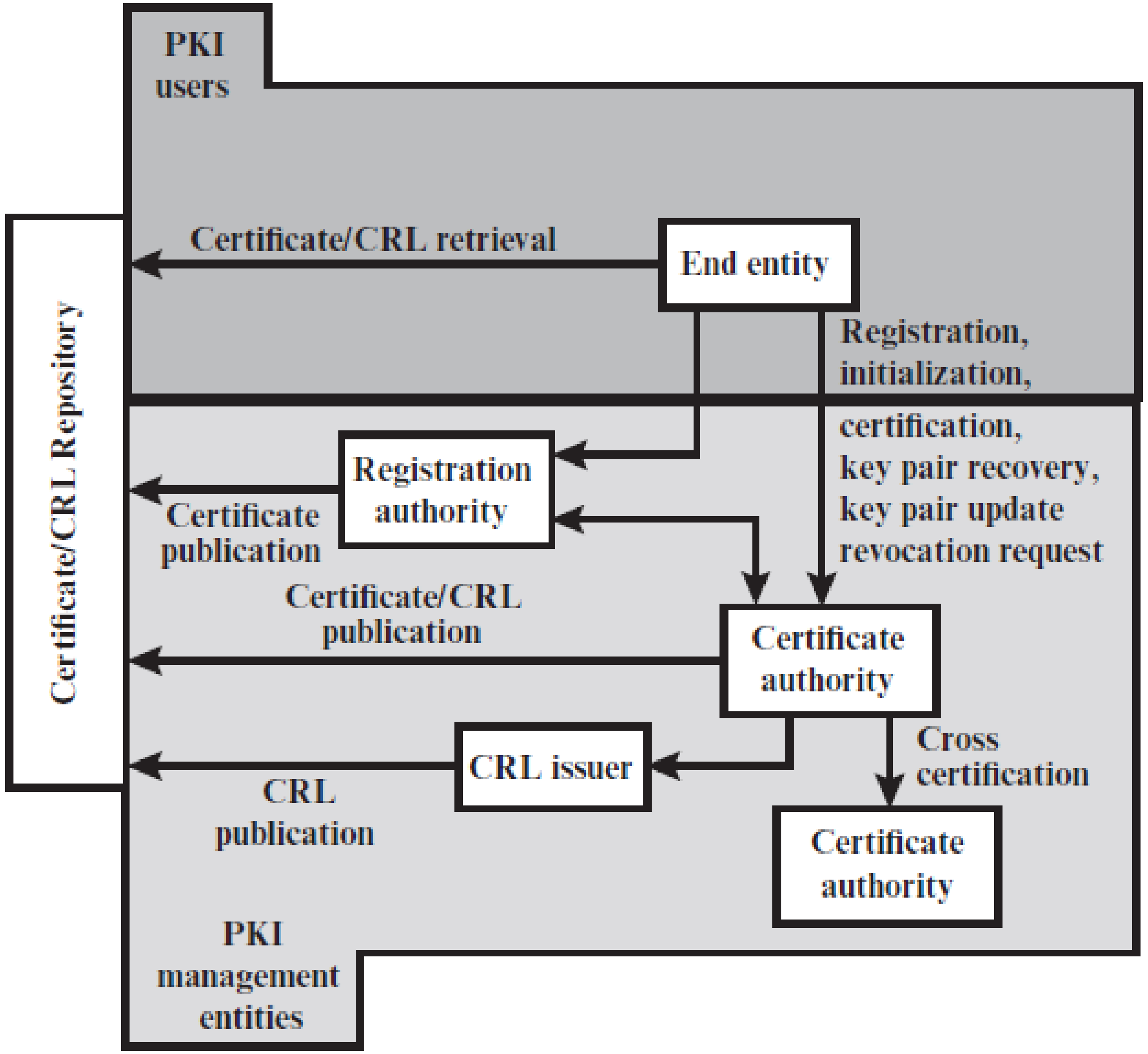

PKI users
Certificate/CRL retrieval
End entity
Registration, initialization, certification, key pair recovery, key pair update revocation request
Certificate/CRL Repository
Registration authority
Certificate publication
Certificate/CRL publication
Certificate authority
CRL issuer
CRL publication
Cross certification
Certificate authority
PKI management entities

Figure 14.17 PKIX Architectural Model

# PKIX Management Functions

- Registration
  - This is the process whereby a *user first makes itself known to a CA* (directly or through an RA), prior to that CA issuing a certificate or certificates for that user.
  - Registration *begins the process of enrolling in a PKI*.
    - Registration usually involves some offline or online procedure for mutual authentication.
    - Typically, the *end entity is issued one or more shared secret keys* used for subsequent authentication.
- Initialization
  - Before a client system can operate securely, it is necessary to *install key materials that have the appropriate relationship with keys stored elsewhere in the infrastructure*.
    - For example, the client needs to be *securely initialized with the public key and other assured information of the trusted CA(s)*, to be used in validating certificate paths.
- Certification
  - This is the process in which a *CA issues a certificate for a user's public key*, returns that certificate to the user's client system, and/or *posts that certificate in a repository*.

# PKIX Management Functions

- Key-pair recovery
    - Key pairs can be used to support digital signature creation and verification, encryption and decryption, or both.
    - When a key pair is used for encryption/decryption, it is important to *provide a mechanism to recover the necessary decryption keys when normal access to the keying material is no longer possible*, otherwise it will not be possible to recover the encrypted data.
        - Loss of access to the decryption key can result *from forgotten passwords/PINs, corrupted disk drives, damage to hardware tokens*, and so on.
        - Key pair recovery allows end entities to *restore their encryption/decryption key pair* from an authorized key backup facility (typically, the CA that issued the end entity’s certificate).
- Key pair update
    - All key pairs need to be *updated regularly* (i.e., replaced with a new key pair) and new certificates issued. Update is required when the *certificate lifetime expires* and as a result of *certificate revocation*.

# PKIX Management Functions

- Revocation request
  - An authorized person advises a CA of an *abnormal situation requiring certificate revocation*.
    - Reasons for revocation include *private key compromise, change in affiliation, and name change*.
- Cross certification
  - Two CAs exchange information used in *establishing a cross-certificate*.
    - A cross-certificate is a *certificate issued by one CA to another CA that contains a CA signature key used for issuing certificates*.

# PKIX Management Protocols

- The PKIX working group has defined two alternative management protocols between PKIX entities that support the management functions listed in the preceding subsection.
- RFC 2510 defines the **certificate management protocols** (CMP).
- Within CMP, each of the management functions is explicitly identified by specific protocol exchanges.
- CMP is designed to be a flexible protocol able to accommodate a variety of technical, operational, and business models.

# PKIX Management Protocols

- RFC 2797 defines **certificate management messages** over CMS (CMC), where CMS refers to RFC 2630, cryptographic message syntax. CMC is built on earlier work and is intended to leverage existing implementations.
- Although all of the PKIX functions are supported, the functions do not all map into specific protocol exchanges.

# Software Security

- Malware
- Types of malware
  - Virus
  - Worms
  - Trojan horse
  - Adware
  - Spyware
  - Backdoor
  - Ransomware
  - Rootkits
  - Bootkits
- Malware analysis and countermeasures

# Topics

- What malware are?
- How do they infect hosts?
- How do they propagate?
- How to detect them?
- To prevent them?
- Malware analysis
- Conclusion

# Malware Definition

- “A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system (OS) or of otherwise annoying or disrupting the victim.” (NIST –2005)
- Also called digital pests

# What can it do?

- Steal personal information
- Delete files
- Make you click fraud
- Steal software serial numbers
- Use your computer as relay (zombie)
- Corrupt files
- Other things as well

# Types of Malware

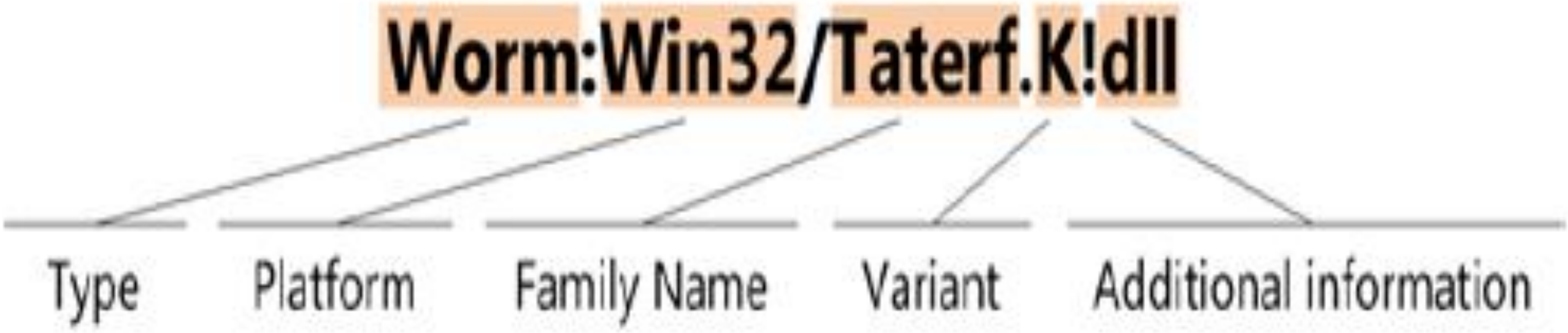
- Virus: attaches itself to a program
- Worm: propagates copies of itself to other computers
- Logic bomb: “explodes” when a condition occurs
- Trojan horse: fakes/contains additional functionality
- Backdoor (trapdoor): allows unauthorized access to functionality
- Spyware: used to spy on victim’s activities on a system and also for stealing sensitive information of the client.

# Types of Malware

- Ransom-ware: steals some functionality and returns after a ransom is paid
- Scare-ware: users are tricked by scaring and motivated to perform some action. E.g. buying a software license
- Key-loggers: capture keystrokes
- Browser hijacker: modifies a web browser's settings without a user's permission, to inject unwanted advertising into the user's browser
- Zombie: software on infected computers that launch attack on others

# Malware Naming

- CARO (computer antivirus researchers organization)
- CARO naming convention (1991)



# History

- 1982 First reported virus : Elk Cloner (Apple 2)
- 1983 Virus gets defined
- 1986 First PC virus MS DOS (Brain written by two Pakistani brothers)
- 1988 First worm : Morris worm
- 1998 Back orifice: remote management tool
- 1999 Melissa virus: macro virus
- 1999 Zombie concept
- 2000 love bug: vbsworm: damage: $15B
- 2001 Nimdaworm
- 2003 SQL Slammer worm: damage $1.2B (Vulnerability: buffer overflow)
- 2001 Code Red: DoSworm, damage: $2.6B
- 2004 MYDOOM Ddosworm, damage: $38B

# What is a virus?

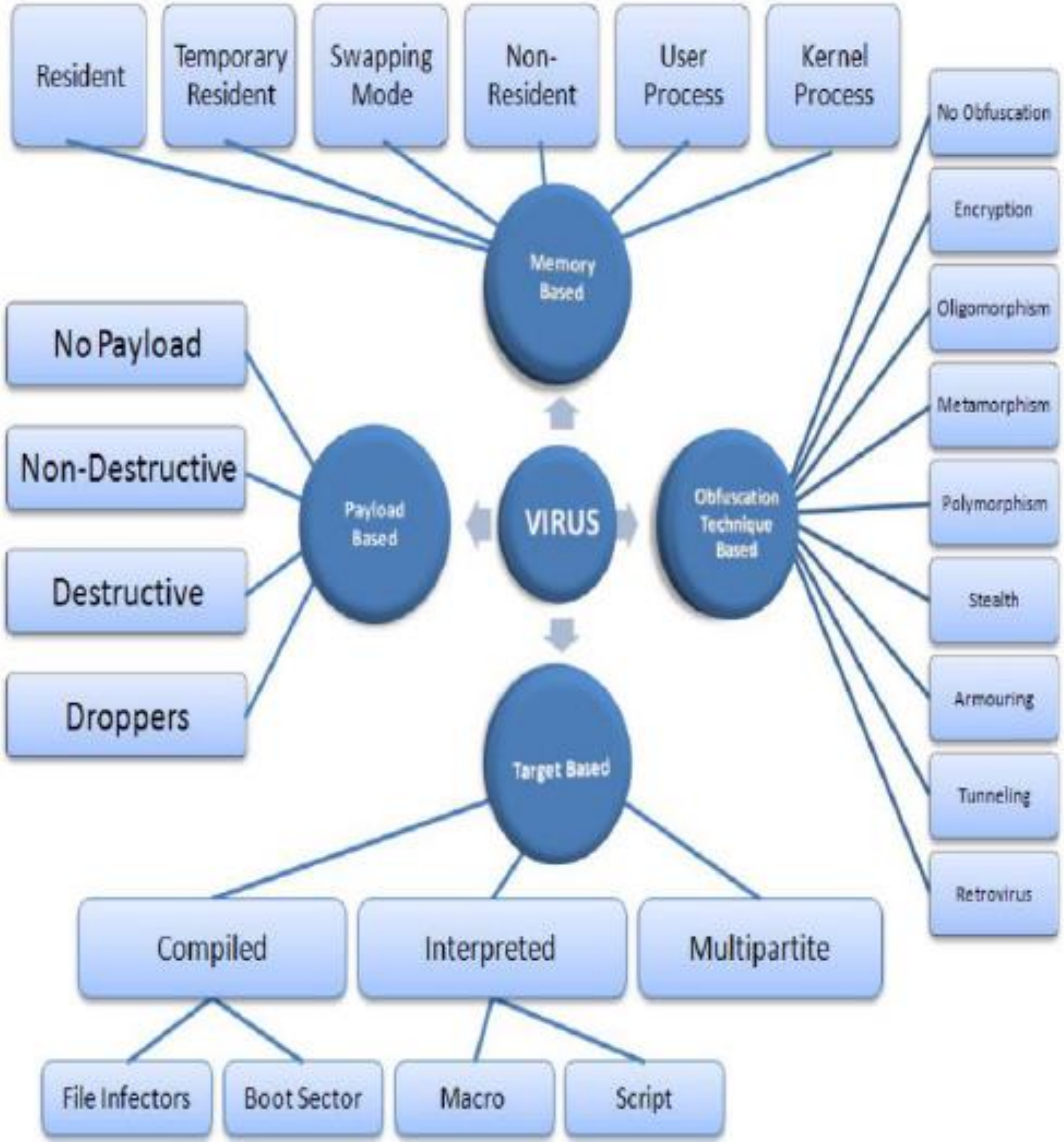
- A program that can infect other programs by modifying them to include a, possibly evolved, version of itself (Fred Cohen 1983)
- It executes secretly when host program is run
- Inserts copies of itself into host programs/data files
- Requires user interaction
- Often specific to operating system and hardware
  - taking advantage of their details and weaknesses

# Classification of Virus

Classification of viruses can be done as follows:

- Memory Based
  - How they live (stay) in memory
- Target Based
  - How they spread to others
- Obfuscation Technique Based
  - What they do to hide
- Payload Based
  - What they do after infection

# Classification of Virus



# Memory based Classification

- This categorization is based on their behavior while they operate in memory.
  - Resident
  - Temporary Resident
  - Swapping Mode
  - Non-Resident
  - User Process
  - Kernel Process

# Memory based Classification (cont’d)

- **Resident Virus**:
  - The virus code is loaded into memory and is copied to all the relevant host files that are running in the memory.
  - For example: A TSR [Terminate and Stay Resident] program that stays in the allocated memory even after the termination of the main program
- **Temporary Resident Virus**:
  - Stays in memory temporarily and removes itself out of memory when a certain event occurs
  - Extremely difficult to detect

# Memory based Classification (cont’d)

- **Swapping Memory Virus**:
  - Such kind of viruses load only a part of their code into memory on occurrence of a certain event, infect the files present in memory and unload the code from memory.
  - These viruses may be spotted by the increase in disk activity due to loading and unloading of viral code and infection of other host files.

# Memory based Classification (cont’d)

- **Non Resident Virus**:
  - Such viruses do not reside in physical memory.
  - They have an offline mechanism to search for and infect files present in the hard disk.
  - They contain two key sub-routines:
    - Finder or search sub-routine that searches the hard disk for the relevant files to infect
    - Copy sub-routine that copies the virus code into the files found
  - If writable network shares are present, these can spread to other systems using them. These are also called ‘Direct action viruses’.

# Memory based Classification (cont’d)

- **User Process**:
  - These viruses run as a user process and infect files that are accessible.
  - Although the virus can exist as its own process, most of the time, they exist as a sub-process loading before or after the main process.
  - In some cases, the virus exist as a DLL and uses DLL injection method (through registry keys) to load the DLL into the process.
  - Autorun is an example of this type of virus

# Memory based Classification (cont’d)

- **Kernel Process**:
  - These types of viruses generally hook themselves into the kernel through a system driver like program
  - They have the highest privileges after infection as they are present in the kernel space
  - These generally infect/modify the IDT (Interrupt Descriptor Table) to get themselves executed every time a particular interrupt is generated
  - As these viruses require changes to the main file system, they need administrator/superuser privileges to run
  - CIH, Infis are examples of this type of virus