CS 547: Foundation of Computer Security

S. Tripathy
IIT Patna

Previous *Class*

- Malicious code: Malware
  - Viruses
    - *Resident*
    - *Code*
      - *Spreading and payload*

Present Class

- Virus
  - To evade detection
- Malicious code: Malware
  - Worms
- *Other malicious codes*
  - *Backdoor*
  - *Rootkit*
  - *Trojan horse and Logic Bomb*
- *Detection mechanisms*
  - *Signature based*
  - *Behaviour based*

Malware!

- Malware is
  - Software, intended to intercept or take partial control of a computer's operation without the user's consent/ knowledge.
  - It subverts the computer's operation for the benefit of a third party.

- [NIST05] defines malware as:
  - "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 or otherwise annoying or disrupting the victim."

- Malware covers all kinds of intruder software
  - including viruses, worms, backdoors, rootkits, Trojan horses, stealware etc. These terms have more specific meanings.

Viruses

- A virus is a particular kind of malware that infects other files
  - Traditionally, a virus could only infect executable programs
  - Typically, when the file is executed (or sometimes just opened), the virus activates, and tries to infect other files with copies of itself
- Infection
  - For executable programs:
    - Typically, the virus will modify other programs and copy itself to the beginning of the targets' program code
  - For documents with macros:
    - The virus will edit other documents to add itself as a macro which starts automatically when the file is opened

Virus Phases

dormant phase

- virus is idle
- will eventually be activated by some event
- not all viruses have this stage

triggering phase

- virus is activated to perform the function for which it was intended
- can be caused by a variety of system events

propagation phase

- virus places a copy of itself into other programs or into certain system areas on the disk
- may not be identical to the propagating version
- each infected program will now contain a clone of the virus which will itself enter a propagation phase

execution phase

- function is performed
- may be harmless or damaging

Virus structure

```
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:
}
```

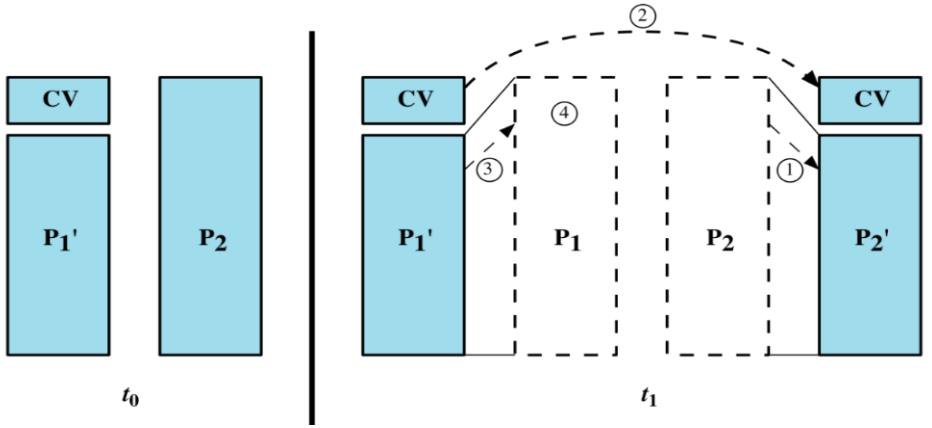
Spread and Payload

- In addition to trying to spread
  - Some viruses try to evade detection by disabling any active virus scanning software
- Most viruses have some sort of payload
  - At some point, the payload of an infected machine will activate, and something (usually bad) will happen
    - Erase your hard drive
    - Subtly corrupt some of your spreadsheets
    - Install a keystroke logger to capture your online banking password
    - Start attacking a particular target website

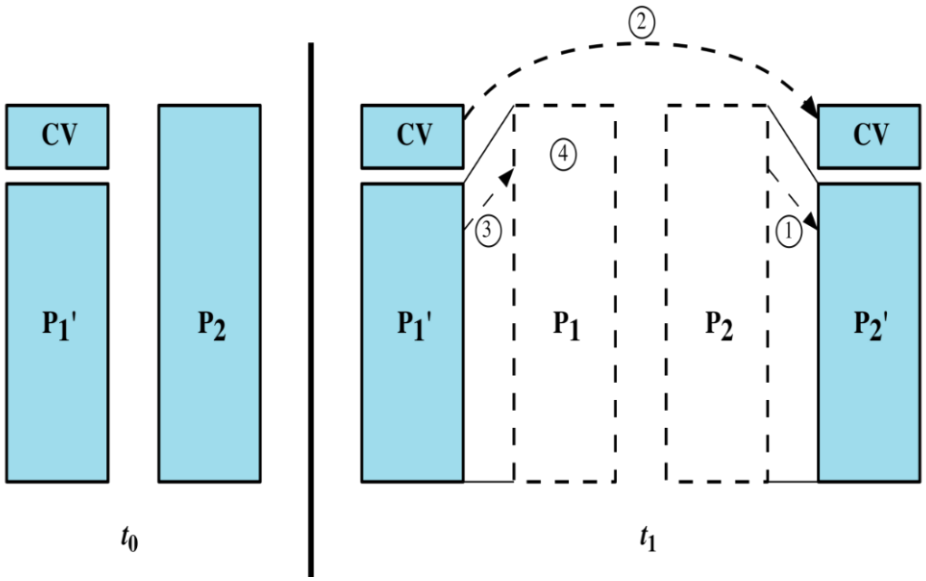
To evade detection!!!

- The infected program will first run the virus code when invoked
  - If the infection phase is fast, then it will be unrecognizable

- Infected version of a program is longer than the normal
  - A virus can compress the infected program to make its versions identical length



How to evade detection: Compression Virus




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

Main: main_program :=
        {is ask_permission then infect-executable
    uncompress rest of file;
    run uncompressed file;}
 }
```

Spotting viruses

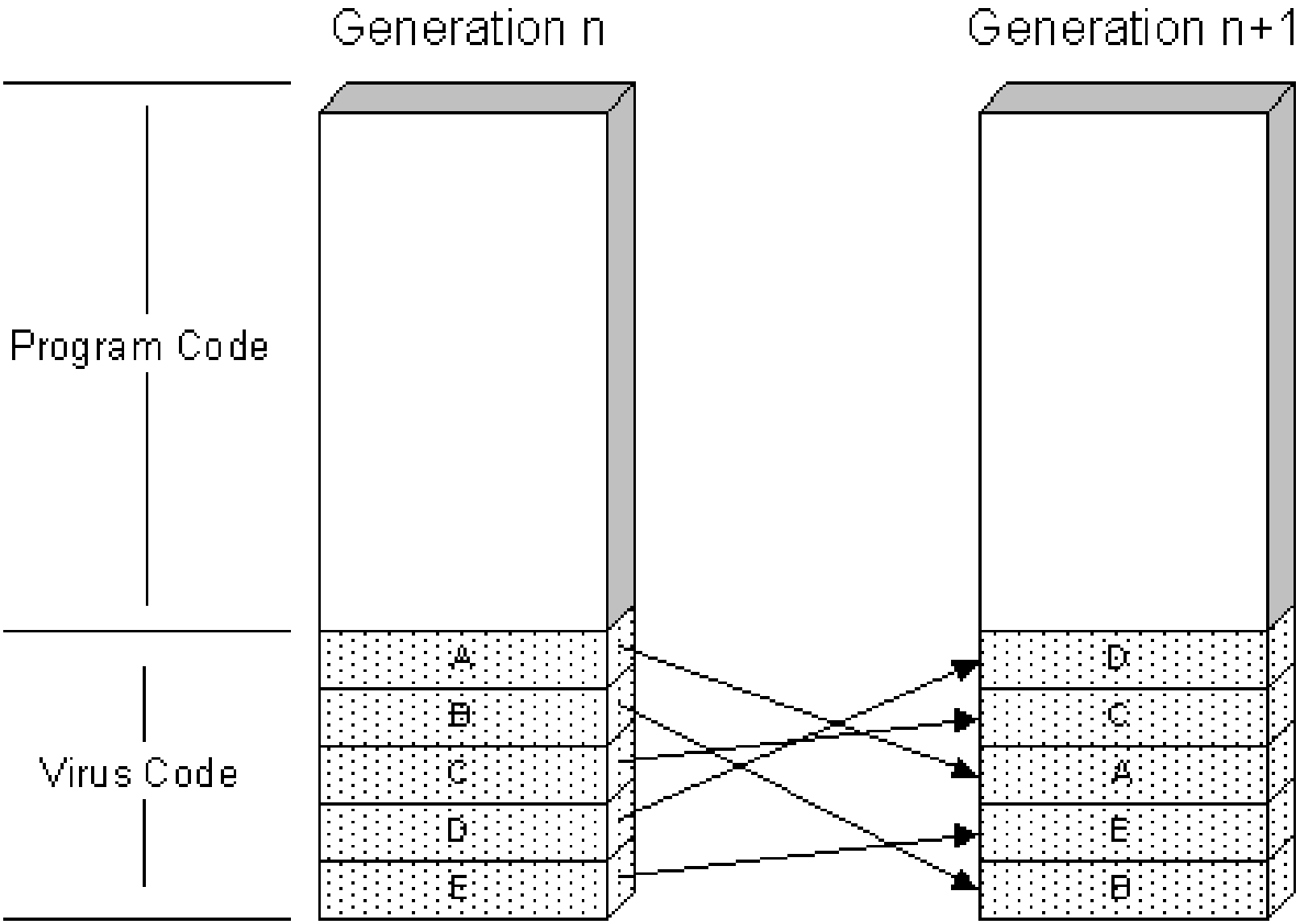
- When should we look for viruses?
  - As files are added to our computer
    - Via portable media
    - Via a network
  - From time to time, scan the entire state of the computer
    - To catch anything we might have missed on its way in
    - But of course, any damage the virus might have done may not be reversible
- How do we look for viruses?
  - Signature-based protection
  - Behaviour-based protection

Signature-based protection

- Keep a list of all known viruses
- For each virus in the list, store some characteristic feature (the signature)
  - Most signature-based systems use features of the virus code itself
    - The infection code
    - The payload code
  - Can also try to identify other patterns characteristic of a particular virus
    - Where on the system it tries to hide itself
    - How it propagates from one place to another

Virus Goal	How Achieved
Attach to executable	Modify file directory / Write to executable pgm file
Attach to data/ control file	Modify directory / Rewrite data Append to data / Append data to self
Remain in memory	Intercept interrupt by modifying interrupt handler address table / Load self in non-transient memory area
Infect disks	Intercept interrupt /Intercept OS call (e.g., to format disk) Modify system file / Modify ordinary executable pgm
Conceal self	Intercept system calls that would reveal self and falsify results / Classify self as "hidden" file
Spread self	Infect boot sector / Infect systems pgm Infect ordinary pgm / Infect data ordinary pgm reads to control its executable
Prevent deactivation	Activate before deactivating pgmand block deactivation Store copy to reinfect after deactivation

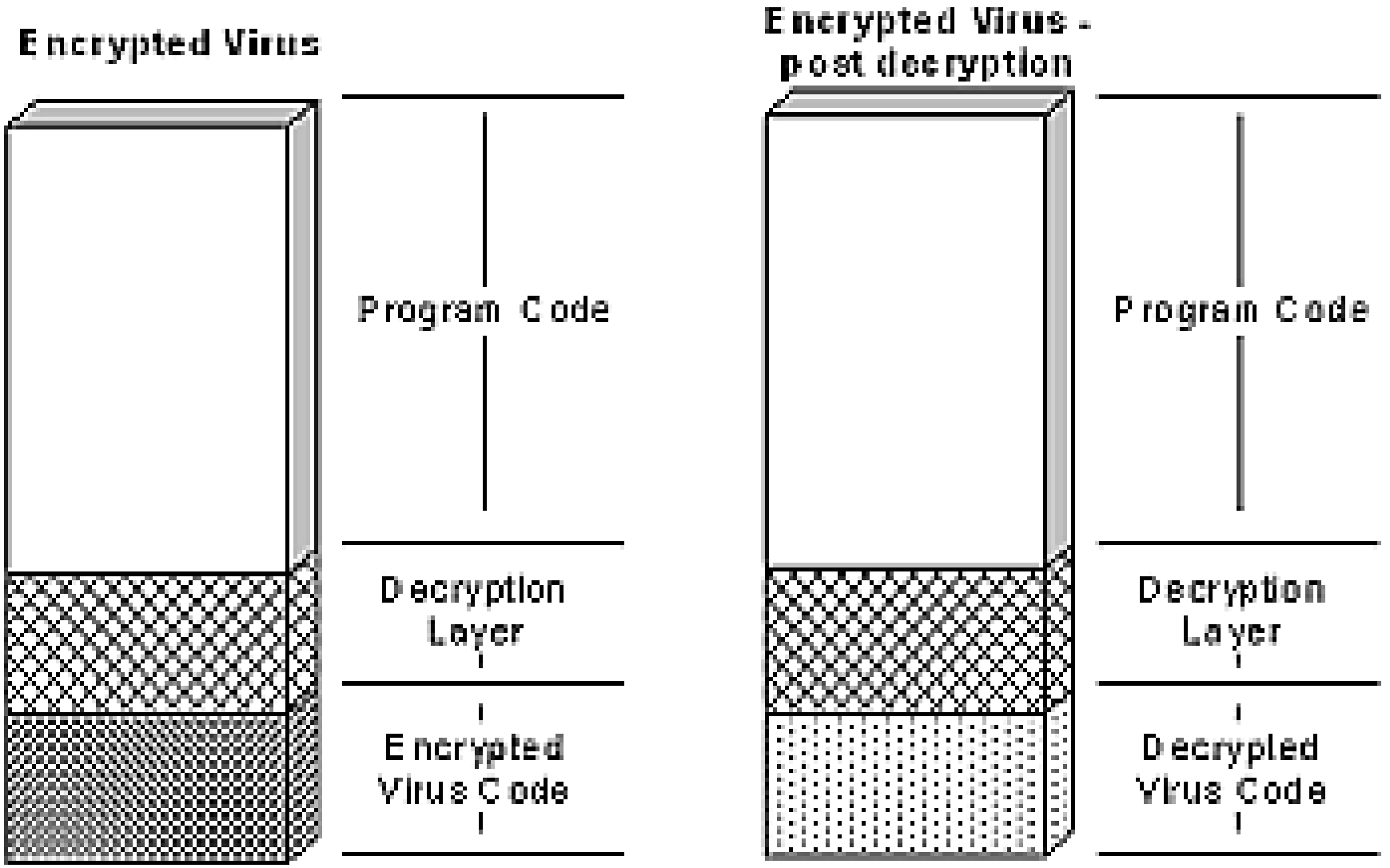
Metamorphic



Polymorphism

- To try to evade signature-based virus scanners, some viruses are polymorphic
  - Instead of making perfect copies of itself every time it infects a new file or host, it makes a modified copy
  - This is often done by encrypting the virus code
    - The virus starts with a decryption routine which decrypts the rest of the virus, which is then executed
    - When the virus spreads, it encrypts the new copy with a newly-chosen random key
- How would you scan for polymorphic viruses?

Encrypted


Encrypted Virus
Program Code
Decryption Layer
Encrypted Virus Code
Encrypted Virus - post decryption
Program Code
Decryption Layer
Decrypted Virus Code

Encrypting virus structure

- Encrypting virus structure (informal pseudo-code)

```
        array decr_key;
        procedure decrypt(virus_code, decr_key)
            ...
        end /* decrypt */
    begin /* virus V in target pgm T */
            decrypt (V, decr_key);
    infect: if infect_condition met then
            find new target pgms NT to infect;
            mutate V into V' for copying;
            encrypt V' with random key into V";
            save new key in file for V";
            attach V" to NT;
        hide modification of NT (with stealth
            code of V);
    damage: if damage_condition met then
            execute damage_code of V
        else start T
end /* virus V in target pgm T */
```

sto-red en-cryp-ted

Virus Signatures Detecting Virus Signatures

- **Encrypting virus**: Encrypts its object code (each time with a different/random key), decrypts code to run

- Q: Is there any signature for encryption virus that a scanner can see?
  - Hint: consider 3 parts of encryption virus:
    - „proper" virus code (infect/damage code)
    - decr_key
    - procedure decrypt

Virus Signatures Detecting Virus Signatures

- Q: Is there any signature for encryption virus that a scanner can see?
- A:
  - „proper" virus code – encrypted with random key – polymorphic
  - decr_key – random key used to encrypt/decrypt – polymorphic
  - procedure decrypt (or a pointer to a library decrypt procedure) – unencrypted, static

    => procedure *decrypt* of V is its signature

    visible to a scanner
- But: Virus writer can use polymorphic techniques on decryption code to make it „less visible" (to hide it)

- Virus writers and scanner writers challenge each other
  - An endless game?

Behaviour-based protection

- Signature-based protection systems have a major limitation
  - You can only scan for viruses that are in the list!
  - But there are several brand-new viruses identified every day
    - More than 75 thousand
  - What can we do?

- Behaviour-based systems look for suspicious patterns of behaviour, rather than for specific code fragments
  - Of course, this is only useful post-infection

False negatives and positives

- Any kind of test or scanner can have two types of errors:
  - False negatives: fail to identify a threat that is present
  - False positives: claim a threat is present when it is not
- Which is worse?
- How do you think signature-based and behaviour-based systems compare?

Logic bombs

- Logic bomb is a malicious code hiding in the software already on your computer, waiting for a certain trigger to "go off" (execute its payload)
- Logic bombs are usually written by "insiders", and are meant to be triggered sometime in the future
  - After the insider leaves the company
- The payload of a logic bomb is usually pretty dire
  - Erase your data
  - Corrupt your data
  - Encrypt your data, and ask you to send money to some offshore bank account in order to get the decryption key!

Logic bombs

- What is the trigger?
- Usually something the insider can affect once he is no longer an insider
  - Trigger when this particular account gets three deposits of equal value in one day
  - Trigger when a special sequence of numbers is entered on the keypad of an ATM
  - Just trigger at a certain time in the future (called a "time bomb")

Trojan horses

- Trojan horses are programs which claim to do something innocuous (and usually do), but which also hide malicious behaviour

You're surfing the Web and you see a button on the Web site saying, "Click here to see the dancing pigs." And you click on the Web site and then this window comes up saying, "Warning: this is an untrusted Java applet. It might damage your system. Do you want to continue? Yes/No." Well, the average computer user is going to pick dancing pigs over security any day. And we can't expect them not to.

-- Bruce Schneier

Spotting Trojan horses and logic bombs

- Spotting Trojan horses and logic bombs is extremely tricky. Why?

- The user is intentionally running the code!
  - Trojan horses: the user clicked "yes, I want to see the dancing pigs"
  - Logic bombs: the code is just (a hidden) part of the software already installed on the computer

- Don't run code from untrusted sources?
- Better: prevent the payload from doing bad things

Thanks