

Cybersecurity

Virus and other Malicious Code

Kasun De Zoysa

*Department of Communication and Media Technologies
University of Colombo School of Computing
University of Colombo
Sri Lanka*

Can we trust software?

You can't trust code you did not totally create yourself.

(Especially code from companies that employ people like you and me)."



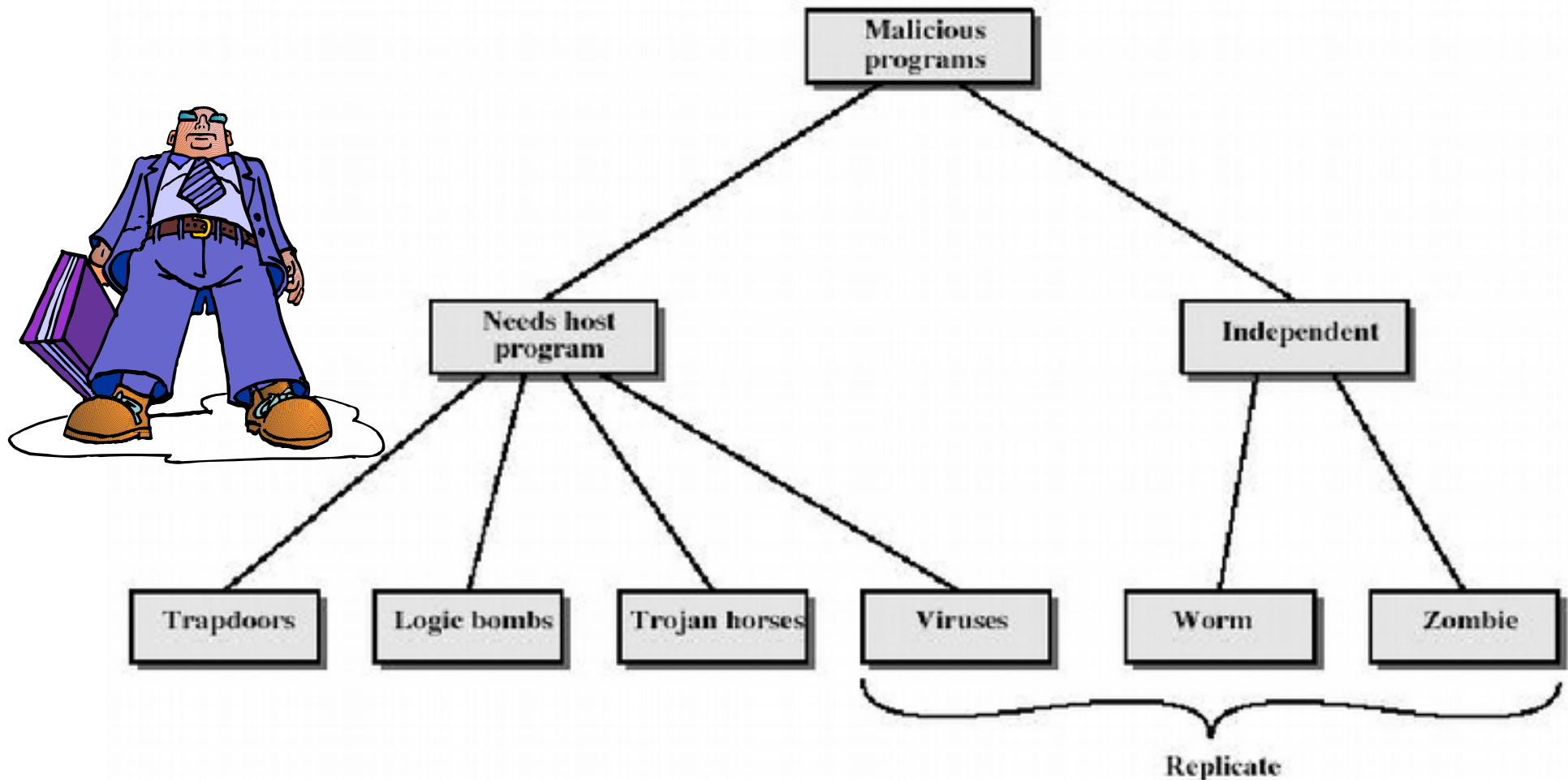
Malicious Software

- Malicious code often masquerades as good software or attaches itself to good software
- Some malicious programs need host programs
 - Trojan horses, logic bombs, viruses
- Others can exist and propagate independently
 - Worms, automated viruses



Image courtesy of: Tech Tips.com

Malicious Software

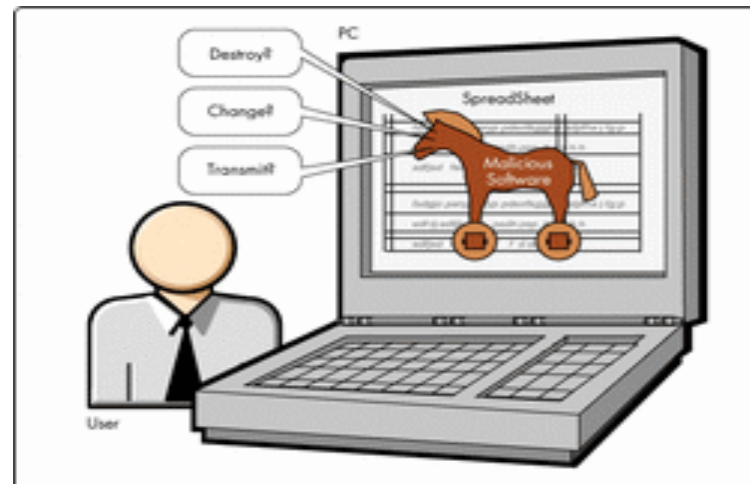


Logic Bombs

- A program that performs an action that violates the site security policy when some external event occurs
- Example: program that deletes company's payroll records when one particular record is deleted
 - The “particular record” is usually that of the person writing the logic bomb
 - Idea is if (when) he or she is fired, and the payroll record deleted, the company loses *all* those records

Trojan Horses

- A **trojan horse** is malicious code hidden in an apparently useful host program
- When the host program is executed, trojan does something harmful or unwanted
- Trojans do not replicate
 - This is the main difference from worms and viruses



Viruses

- Virus - a program that replicates itself and infects computers
 - ✓ Needs a host file
 - ✓ May use an email program to infect other computers
 - ✓ The attack is called the payload

Types of Viruses

- Boot sector infectors
- Executable infectors
- Multipartite viruses
- TSR viruses
- Stealth viruses
- Encrypted viruses
- Polymorphic viruses
- Macro viruses

Polymorphic Viruses

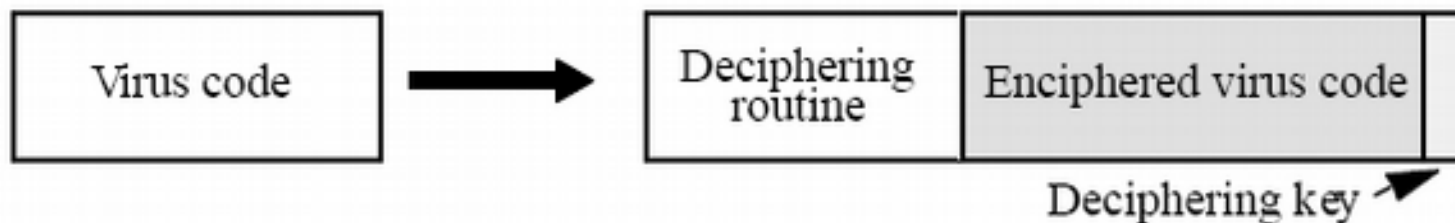
- A virus that changes its form each time it inserts itself into another program
- Idea is to prevent signature detection by changing the “signature” or instructions used for deciphering routine
- At instruction level: substitute instructions
- At algorithm level: different algorithms to achieve the same purpose
- Toolkits to make these exist (Mutation Engine, Trident Polymorphic Engine)

Example

- These are different instructions (with different bit patterns) but have the same effect:
 - add 0 to register
 - subtract 0 from register
 - xor 0 with register
 - no-op
- Polymorphic virus would pick randomly from among these instructions

Encrypted Viruses

- A virus that is enciphered except for a small deciphering routine
 - Detecting virus by signature now much harder as most of virus is enciphered



Polymorphic Virus Techniques

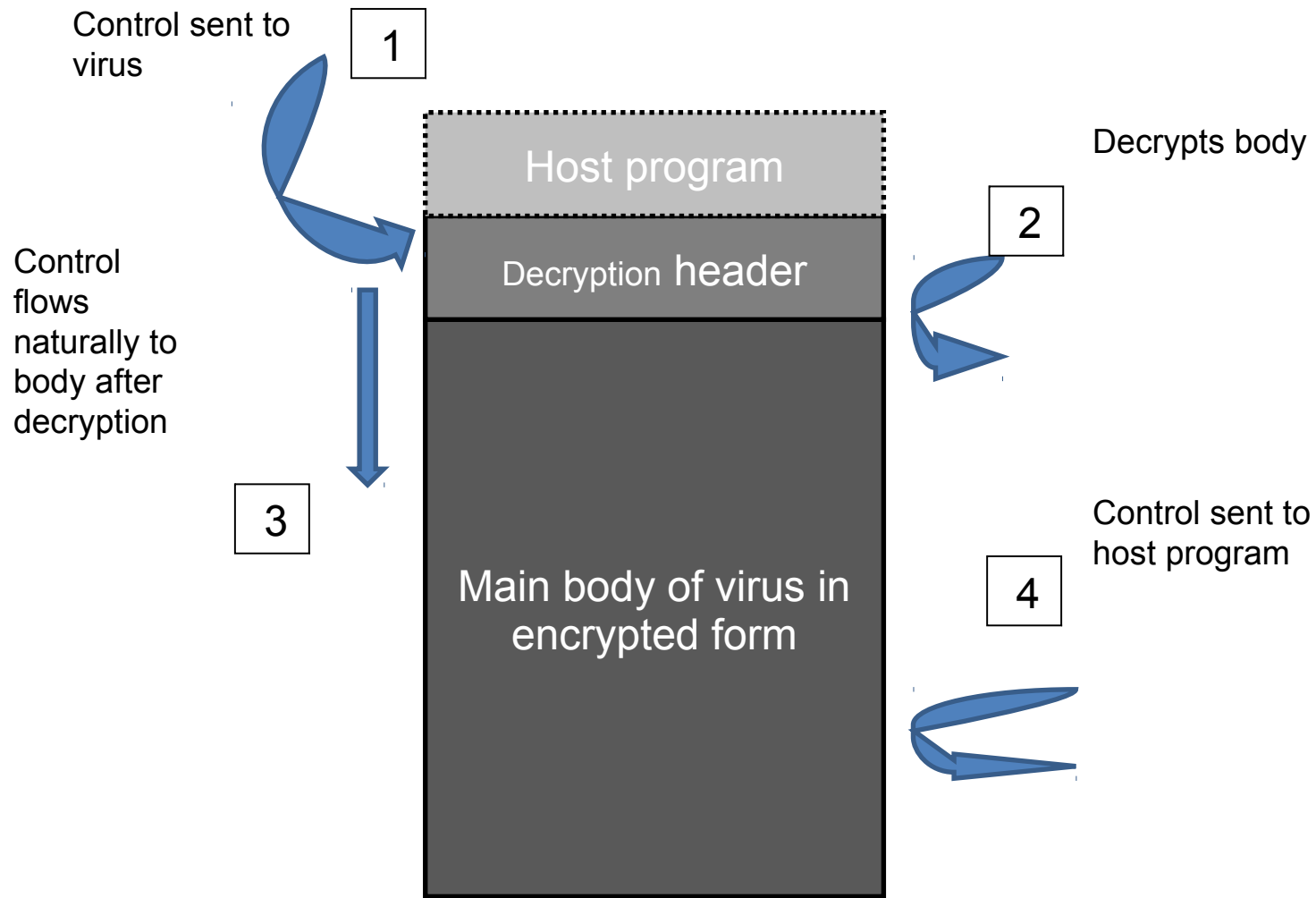
□ If the ciphering is known, the deciphering routine can be guessed

If the key is present in the virus, the virus is fully known

⇒ Use asymmetric cryptography



Polymorphic Viruses



Typical polymorphic virus

Macro Viruses

- A virus composed of a sequence of instructions that are interpreted rather than executed directly
- Can infect either executables (Duff's shell virus) or data files (Highland's Lotus 1-2-3 spreadsheet virus)
- Independent of machine architecture
 - But their effects may be machine dependent

Example

- Melissa
 - Infected Microsoft Word 97 and Word 98 documents
 - Windows and Macintosh systems
 - Invoked when program opens infected file
 - Installs itself as “open” macro and copies itself into Normal template
 - This way, infects any files that are opened in future
 - Invokes mail program, sends itself to everyone in user’s address book

Malicious PDFs

Without question, if someone emails you a PDF file, opening it in the Adobe Reader is a Defensive Computing mistake.

- SumatraPDF



Image courtesy of: Tech Tips.com

Didier Stevens' – PDF Tools

- You can use **pdfid.py** to identify a Java Script occurrences
- With a clear indicator of JavaScript inclusion, you can then use **pdf-parser.py** to learn further details.
- **make-pdf-javascript.py** allows one to create a simple PDF document with embedded JavaScript that will execute upon opening of the PDF document.

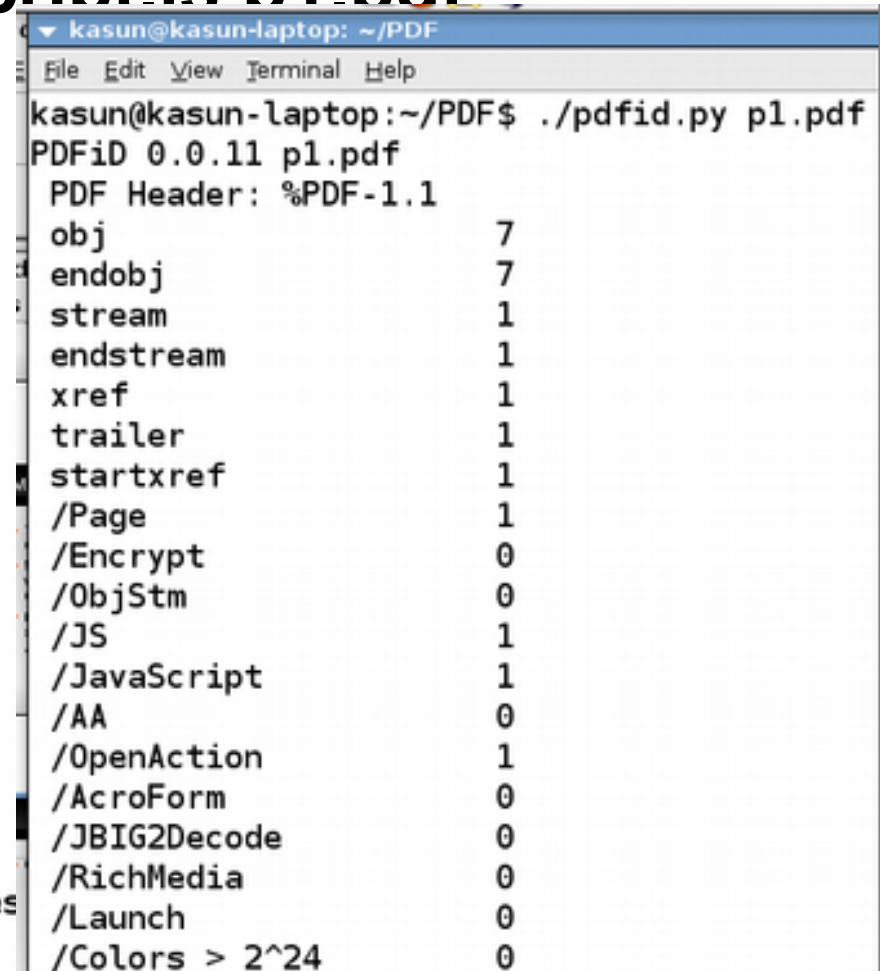
Didier Stevens' – PDF Tools

make-pdf-javascript.py -f myscript.js p1.pdf

pdfid.py p1.pdf

pdf-parser.py -s javascript p1.pdf

```
<<
  /Type /Action
  /S /JavaScript
  /JS (var result = app.alert({
    cMsg: "Are you going to read it again?",
    cTitle: "Yes I am!",
    nIcon: 2,
    nType: 2
  }));
  if(result==4)
  app.alert({cMsg:'I am formatting your HD', cTitle: 'Tes
n: 3});
)
>>
```



```
kasun@kasun-laptop: ~/PDF
File Edit View Terminal Help
kasun@kasun-laptop:~/PDF$ ./pdfid.py p1.pdf
PDFiD 0.0.11 p1.pdf
PDF Header: %PDF-1.1
obj 7
endobj 7
stream 1
endstream 1
xref 1
trailer 1
startxref 1
/Page 1
/Encrypt 0
/ObjStm 0
/JS 1
/JavaScript 1
/AA 0
/OpenAction 1
/AcroForm 0
/JBIG2Decode 0
/RichMedia 0
/Launch 0
/Colors > 2^24 0
```



VirusTotal is a free service that **analyzes suspicious files and URLs** and facilitates the quick detection of viruses, worms, trojans, and all kinds of malware.

 File

 URL

 Search

No file selected

Choose File

Maximum file size: 64MB

By clicking 'Scan it!', you consent to our [Terms of Service](#) and allow VirusTotal to share this file with the security community. See our [Privacy Policy](#) for details.

Scan it!

How Viruses Attach

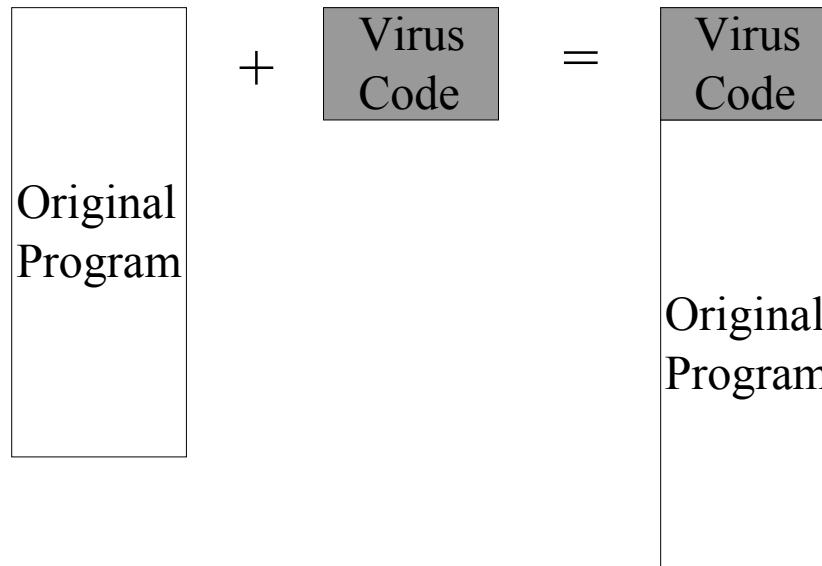
- Appended to other programs
- Surround other programs
- Integration with other programs
- Replacement

Viruses that Append

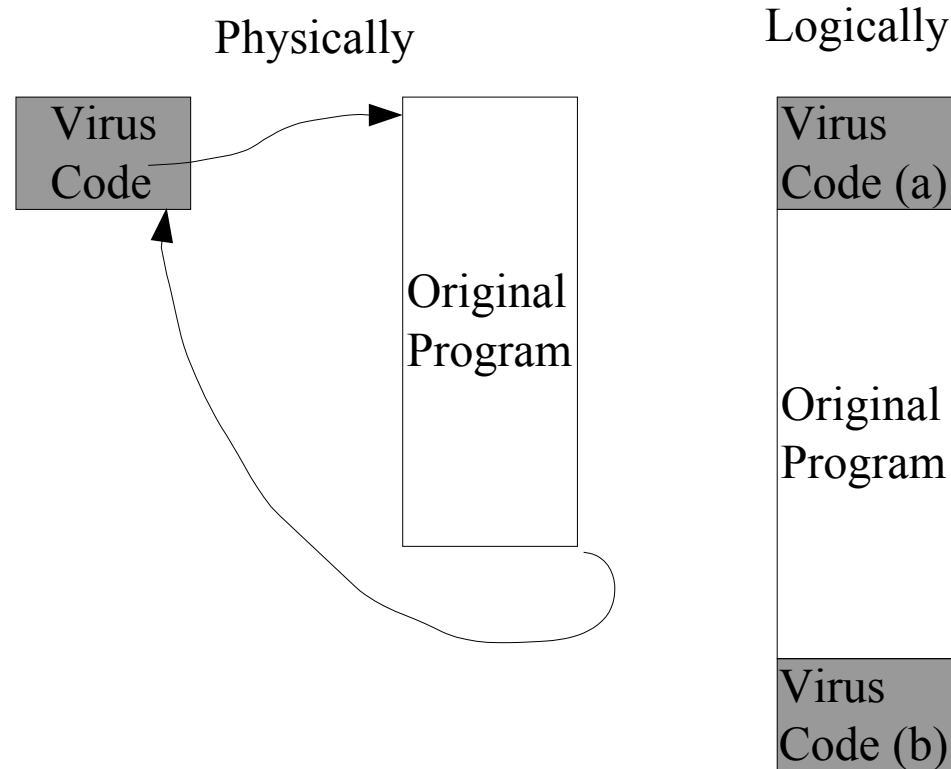
- Easy to program
- Inserted before the non-malicious code
- Virus code executed first
- After virus code program flow continues with non-malicious code normally

Viruses that Append

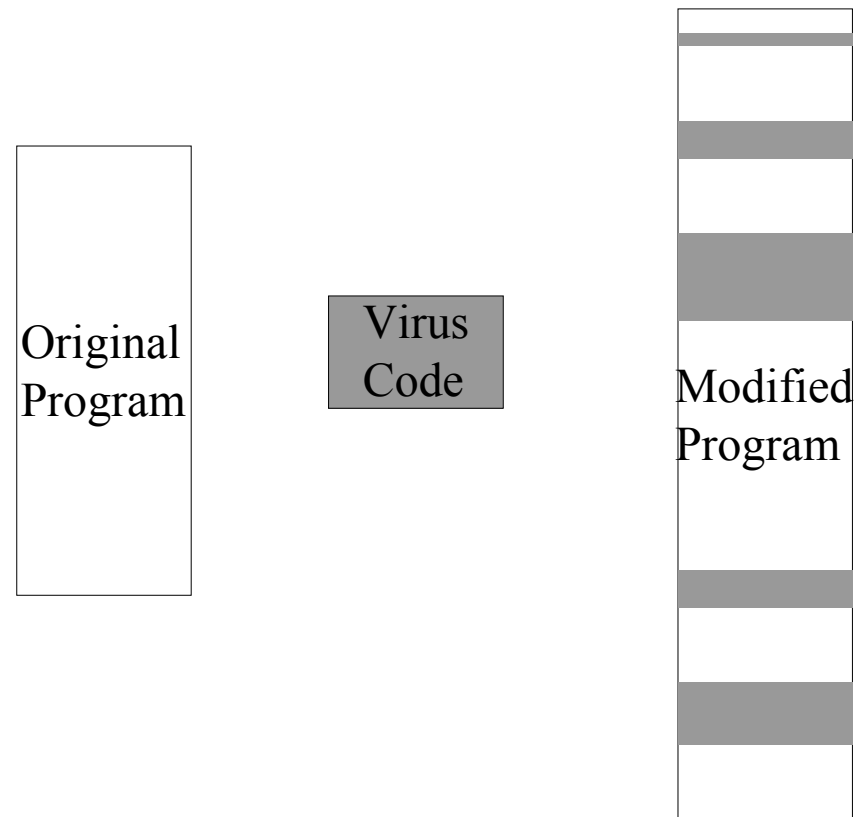
- Most viruses operate in this manner



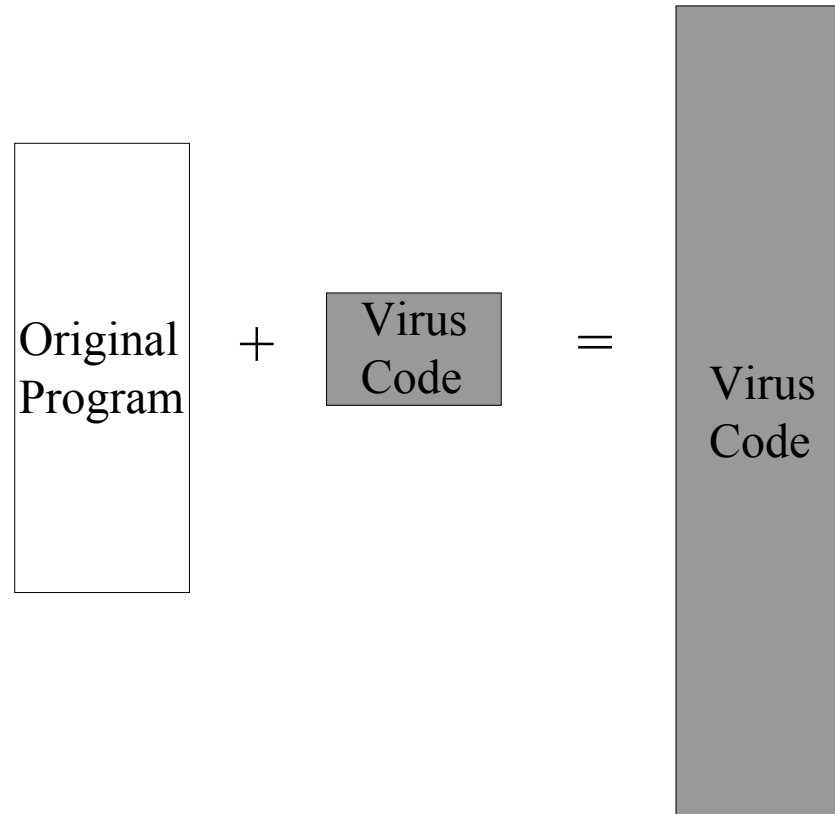
Viruses that surround



Integrated Viruses



Viruses that replace



Virus Phases

1. Dormant Phase



3. Triggering Phase



2. Propagation Phase



4. Execution Phase



Virus Signatures

- A virus cannot be completely invisible. Code for it must be stored somewhere (even if its just in memory)
- Viruses also execute in certain ways
- Spread in certain ways
- (They are essentially still executing on a sequential, deterministic turning machine after all)

Virus Signatures

- All of these aspects of what the virus looks like and acts like adds up to a telltale pattern called a **signature**.
- This signature can be found by other programs that are designed to have knowledge of the signatures and to look for them. (Virus Scanners)

Virus Signatures

- Virus attached to applications or integrated into original code is invariant.
- The beginning of the virus code can serve as a reliable pattern for detecting the presence of the virus.
- Usually located at, or near, the beginning of a programs code so that it will have control before anything else.

Virus Signatures

- File sizes may change as a result of the inserted virus.
- Tricky viruses sometimes avoid this detection though.
- Checksums (MD5) can be used with good reliability to check the actual contents of the file and get a fingerprint of what the contents are

Execution Patterns

- Deviations from expected program behavior can be an alert for the presence of a virus.
 - In fact this is a common excuse these days for just about every deviation; possibly too much so.

Transmission Patterns

- The goal of a virus is usually to spread itself in some manner.
- Slammer, Internet worm, Code-RED all use some type of unique transmission to perform this action.
- Analysis of the communication going on during infection can lead to identification of a virus.

- Worms

- ✓ Self-replicating
- ✓ Do not need a host to travel
- ✓ Travel over networks to infect other machines
- ✓ Conficker worm
 - First released in 2008
 - Reemerged in 2010 with new behaviors



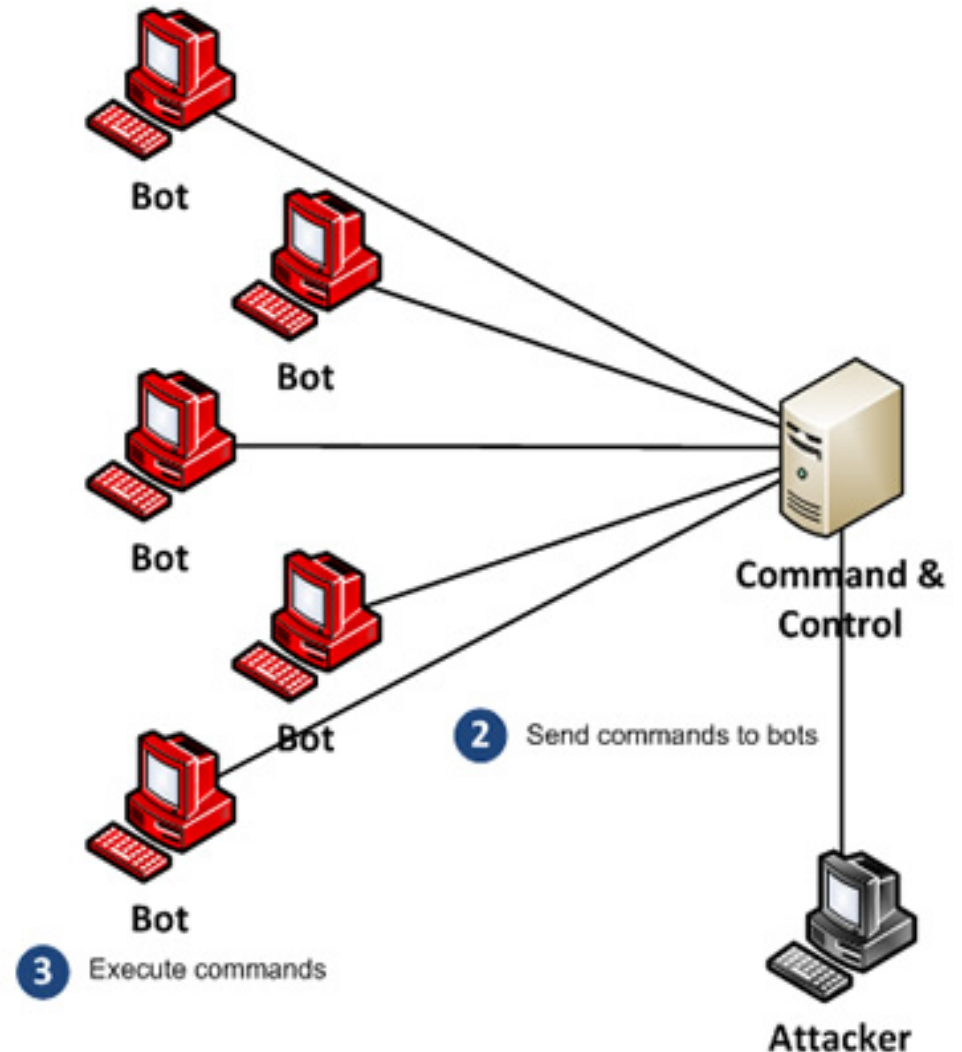
Worms

- replicating but not infecting program
- typically spreads over a network
 - cf Morris Internet Worm in 1988, led to creation of CERTs
- using users distributed privileges or by exploiting system vulnerabilities
- 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

Zombie

- program which secretly takes over another networked computer
- then uses it to indirectly launch attacks
- often used to launch distributed denial of service (DDoS) attacks
- exploits known flaws in network systems

1 Compromise as many devices as possible



Ransom

- The writer of a virus creates a RSA key
 - The public key appears in the body of the virus
 - The private key is kept by the author

The virus spreads, and the payload uses the public key

e.g. it ciphers the data of the target

The author asks for a ransom before sending the private key



Ransom

- The writer of a virus creates a RSA key
The public key is put in the body of the virus
The private key is kept by the author
- The virus spreads
The payload creates a secret key
The secret key is used to cipher data on the disk
The secret key is ciphered with the public key
- The author asks for a ransom before deciphering himself the secret key

Anti-Virus Technologies

□ Simple anti-virus scanners

- Look for **signatures** (fragments of known viruses)
- Heuristics for recognizing code associated with viruses
 - For example, polymorphic viruses often use decryption loops
- Integrity checking to find modified files
 - Record file sizes, checksums, MACs (keyed hashes of contents)

□ Generic decryption and emulation scanners

- Goal: detect polymorphic viruses with known body
- Emulate CPU execution for a few hundred instructions, virus will eventually decrypt, can recognize known body

Defeating Anti-Virus Emulators

- To detect polymorphic viruses, emulators execute suspect code for a little bit and look for opcode sequences of known virus bodies
- Some viruses use **random code block insertion** or insert millions of NOPs at the entry point prior to the main virus body
 - Emulator executes code for a while, does not see virus body and decides the code is benign... when main virus body is finally executed, virus propagates

How Hard Is It to Write a Virus?

- 1000 hits for “virus creation tool”
 - Including dozens of poly- and metamorphic engines
- Virus Construction Toolkit
 - "The perfect choice for beginners"
- Biological Warfare Virus Creation Kit

- Worm Generator
 - Used to create the worm
- Many others

Possible Counter Measures

- Update all softwares like operating system, drivers all softwares that use the internet and update anti virus and anti spyware
- Install inbound and outbound firewall
- **Encrypt important data**
- **Backup the data regularly**
- Install third party registry editor, traffic monitoring software
- Disable autorun feature
- Hope antivirus vendors find a cure for it in near future
- **Use open source software and operating systems**

Conclusion

- ❑ Cryptography with virology is a deadly combination.
- ❑ The battle between Virus writers and anti virus vendors is raging hard
- ❑ The use of cryptography in virology is one such tool used by virus writers to win the battle
- ❑ Anti-virus vendors do not have any answer for such threats as of now and they may come up with remedies.
- ❑ The cycle continues.....
- ❑ So jump out this cycle: Use open source operating systems and software!



Dr. Kasun De Zoysa
e-mail: kasun@ucsc.cmb.ac.lk