CSc 466/566

Computer Security

10 : Malware I — Introduction

Version: 2019/09/23 10:57:49

Department of Computer Science
University of Arizona

collberg@gmail.com

Christian Collberg

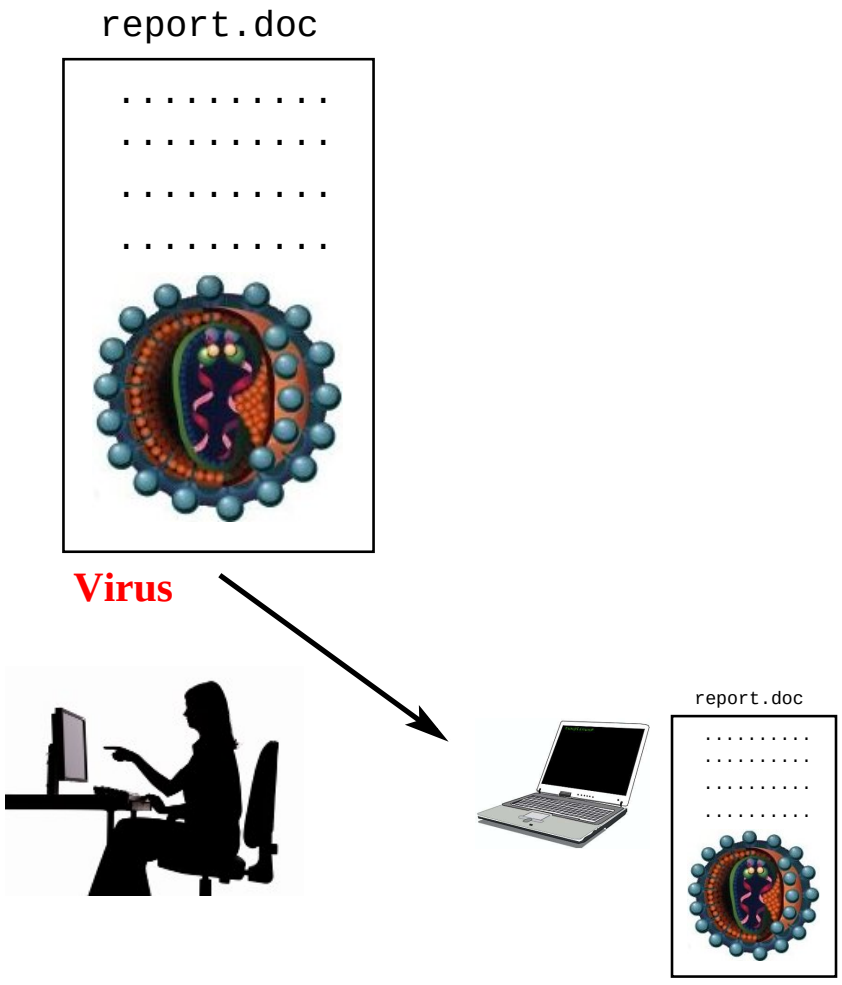
Outline

Logic Bomb

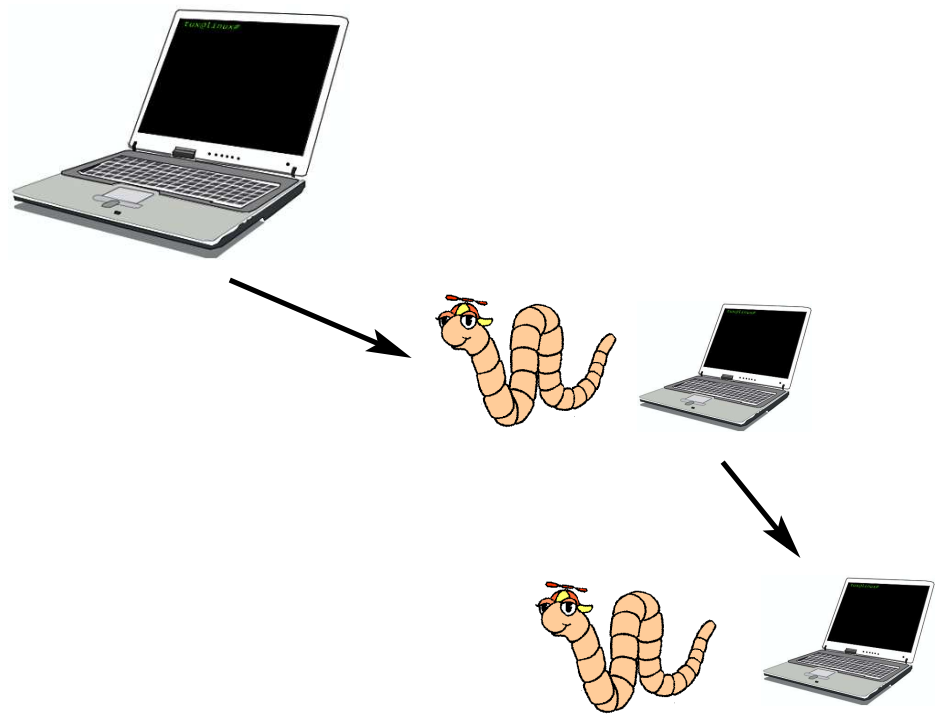

```
if (event)
  do_bad_stuff()
```

Trojan Horse

```
main(){
   play_game();
   send_spam();
}
```


report.doc
Virus
report.doc

Worm




Backdoor

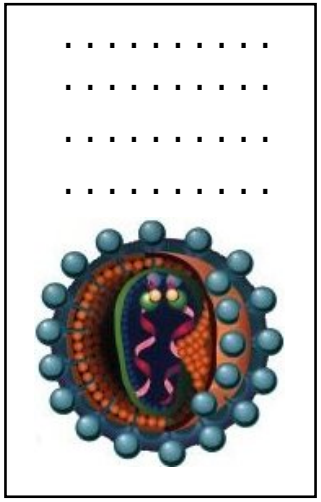
Logic Bomb

```
if (event)
  do_bad_stuff()
```

Trojan Horse

```
main(){
   play_game();
   send_spam();
}
```

report.doc

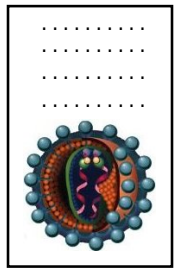


Backdoor

Worm

Virus

report.doc



Definition (Malware (malicious software))

Software used to disrupt computer operations, gather sensitive information, or gain access to private computer systems.
Malware is defined by its malicious intent, acting against the requirements of the computer user.
Malware may steal information, spy on computer users, cause harm, or extort payment.

Source: https://en.wikipedia.org/wiki/Malware

Definition (Backdoor)

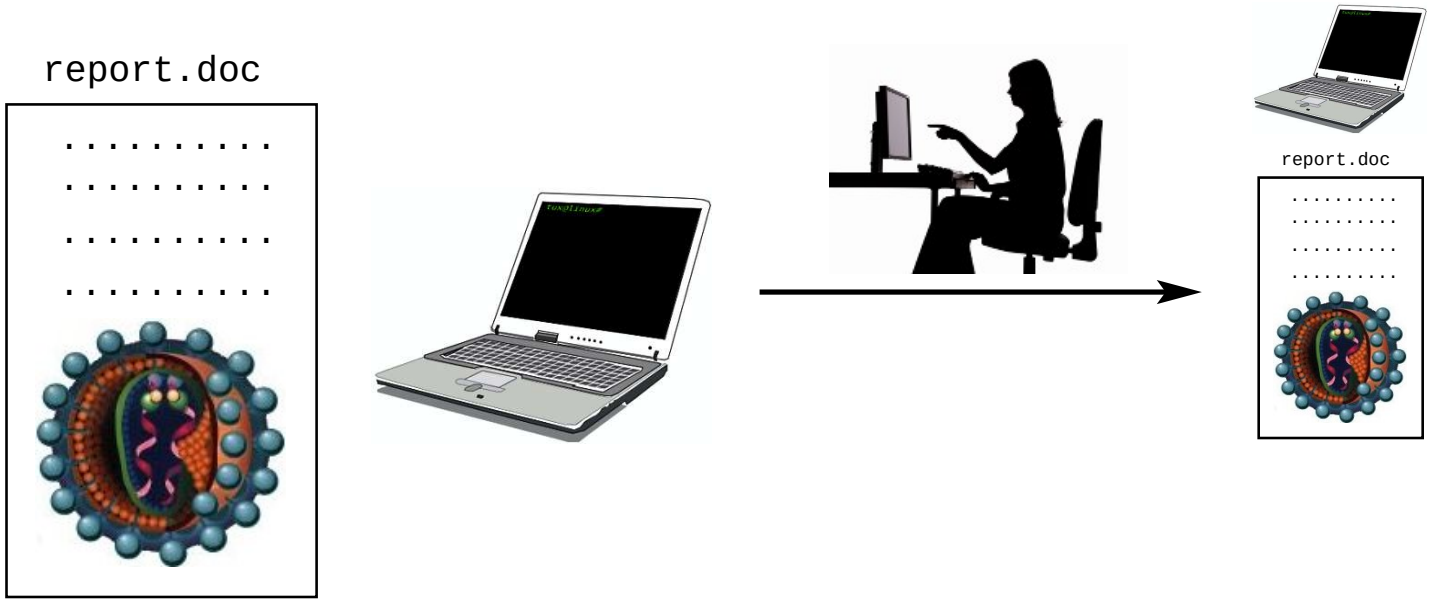
A hidden command inserted by the developer or other malware to allow future access.

Logic Bomb

```
if (event)
  do_bad_stuff()
```

Definition (Logic bomb)

A command inserted by a developer that will cause an action to happen when a certain condition is true.



Definition (Virus)

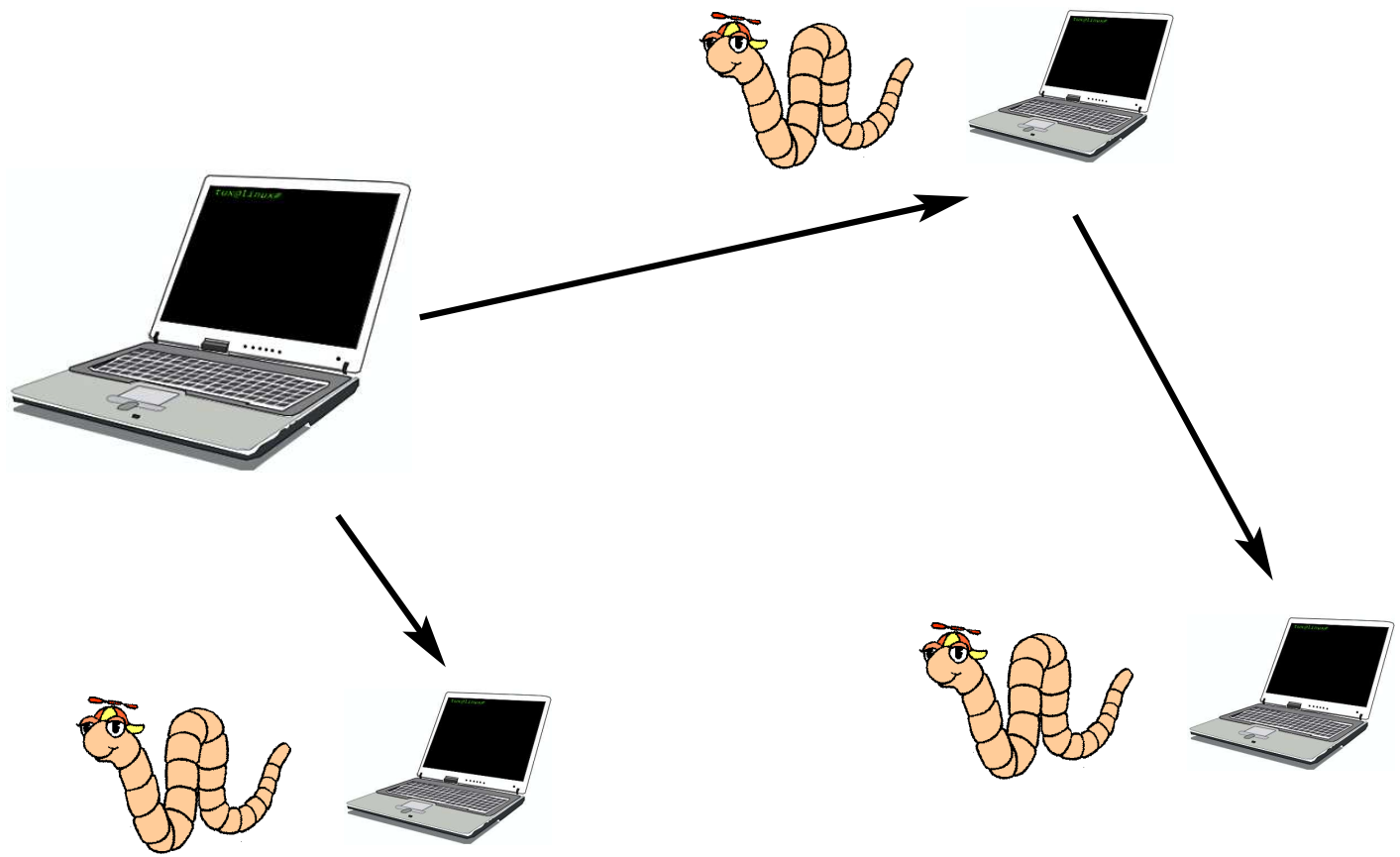
A malware that

1. is self-replicating
2. attaches itself to other files
3. requires user assistance to replicate.

```
main(){
   play_game();
   send_spam();
}
```

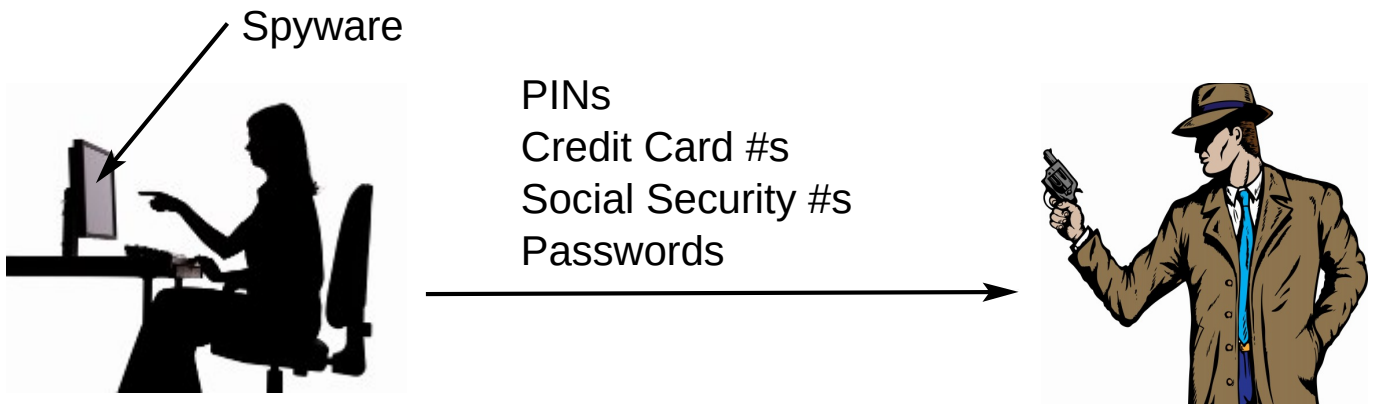
Definition (Trojan horse)

A malware that appears to perform a useful task, and, in addition, performs a malicous task.



Definition (Worm)

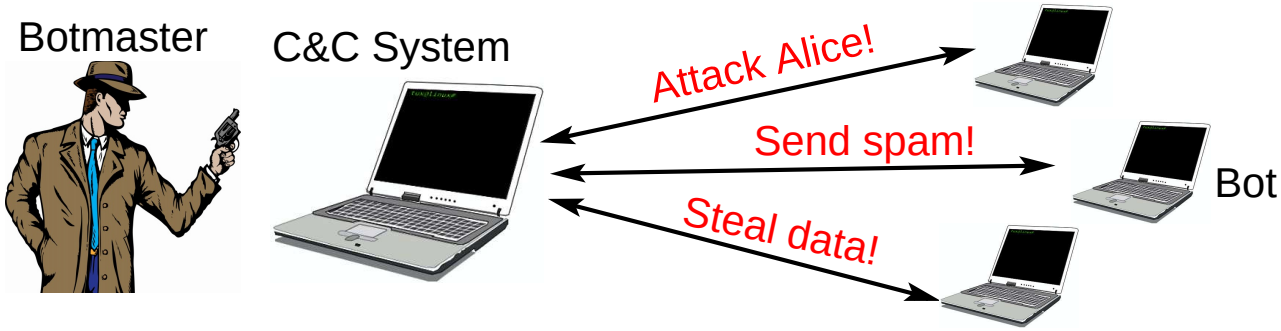
A malware that self-propagates fully working versions of itself to other machines.



Definition (Spyware)

[] software that's designed to gather data from a computer or other device and forward it to a third party without the consent or knowledge of the user. This often includes collecting confidential data such as passwords, PINs and credit card numbers, monitoring keyword strokes, tracking browsing habits and harvesting email addresses.

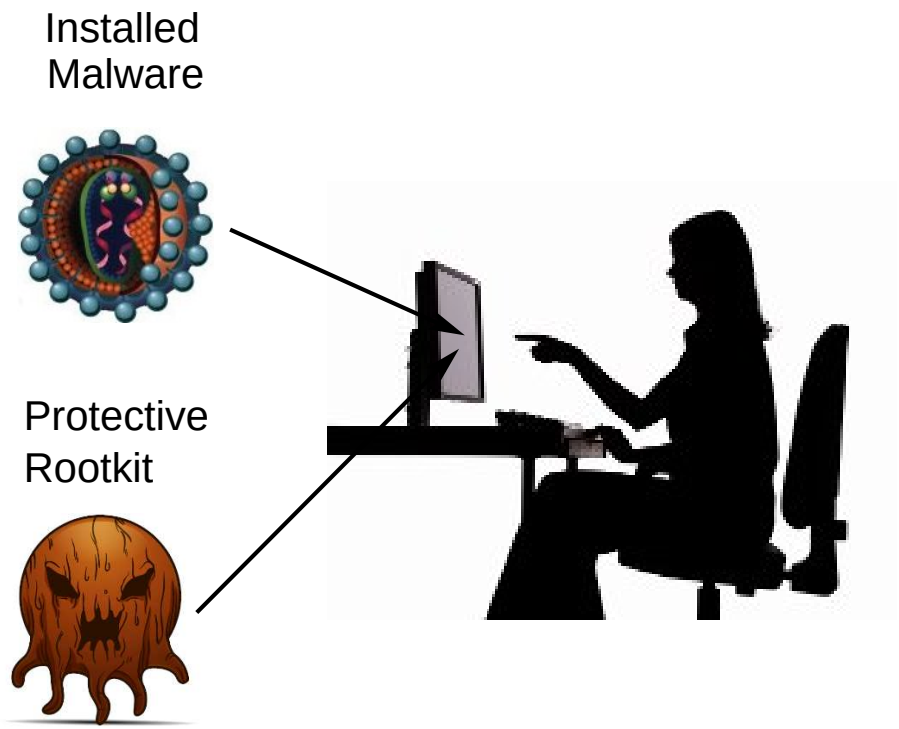
Source: https://usa.kaspersky.com/resource-center/threats/spyware



Definition (Botnet)

A botnet is a number of Internet-connected devices, each of which is running one or more bots. Botnets can be used to perform distributed denial-of-service attack (DDoS attack), steal data, send spam, and allows the attacker to access the device and its connection. The owner can control the botnet using command and control (C&C) software.

Source: https://en.wikipedia.org/wiki/Botnet



Definition (Rootkit)

A rootkit helps a malware to stay hidden on a machine to which it has gained access, and to survive if the machine gets rebooted.

Exercise

- What is a worm?

Exercise

- What is a worm?
- What is a virus?

Exercise

- What is a worm?
- What is a virus?

Exercise

- What is a backdoor?

Exercise

- What is a backdoor?
- What is a logic bomb?

Exercise

- What is a backdoor?
- What is a logic bomb?

Exercise

- What is the difference between a worm and virus?

Exercise

- What is the difference between a worm and virus?
- What is the difference between a virus and trojan?

Exercise

- What is the difference between a worm and virus?
- What is the difference between a virus and trojan?

Exercise

- Can a worm and rootkit cooperate?

Exercise

- Can a worm and rootkit cooperate?
- What is the relationship between a virus and a botnet?

Exercise

- Can a worm and rootkit cooperate?
- What is the relationship between a virus and a botnet?

Exercise

Search the internet to define these types of malware:

- Adware:

Exercise

Search the internet to define these types of malware:

- Adware: programs that display advertisements on your computer, redirect your search requests to advertising websites and collect marketing-type data about you
- Pornware:

Exercise

Search the internet to define these types of malware:

- Adware: programs that display advertisements on your computer, redirect your search requests to advertising websites and collect marketing-type data about you
- Pornware: programs that display pornographic material to advertise fee-based pornographic websites

Exercise

Search the internet to define these types of malware:

- Ransomware:

Exercise

Search the internet to define these types of malware:

- Ransomware: Malware that threatens to publish the victim's data or delete it unless a ransom is paid.

Outline

Insider Threats

Definition (Insider threat)

A malicious threat to an organization that comes from people within the organization, such as employees, former employees, contractors or business associates, who have inside information concerning the organization's security practices.

Source: https://en.wikipedia.org/wiki/Insider_threat

Backdoors

- A backdoor is a hidden command inserted by the developer.
- Backdoors can be inserted
  1. for debugging
  2. to bypass security checks in an emergency
- War Games:
  - Trailer: http://www.youtube.com/watch?v=tAcEzhQ7oqA

Malicious Backdoors

- A programmer may insert a malicious backdoor to gain access later.
- A deliberate vulnerability (buffer overflow, etc.) can be inserted, allowing the programmer easy access.

Logic Bombs

Logic Bomb

```
if (event)
  do_bad_stuff()
```

- The insider could also insert an action that will happen when a certain condition is true:

```
if (Bob is not on the payroll anymore)
   crash_system();
```

Malware Triggers

Malware often contains hidden behavior which is only activated when properly triggered. Well known examples include: the MyDoom worm which DDoS's on particular dates, keyloggers which only log keystrokes for particular sites, and DDoS zombies which are only activated when given the proper command. We call such behavior trigger-based behavior.

Source: http://bitblaze.cs.berkeley.edu/papers/botnet_book-2007.pdf

Logic Bomb + Backdoor

- A logic bomb can be combined with a backdoor that allows the programmer to disable it:

```
boolean disabled=false;

if (today=="April 1" && not disabled)
    delete_all_backups();

void backdoor() {
    disabled = true;
}
```

Fannie Mae Logic Bomb

Unix engineer Rajendrasinh Babubha Makwana, 35, was indicted Tuesday in federal court in Maryland on a single count of computer sabotage for allegedly writing and planting the malicious code on Oct. 24, the day he was fired from his job. The malware had been set to detonate at 9:00 a.m. on Jan. 31, but was instead discovered by another engineer five days after it was planted, according to court records.

Source: http://www.wired.com/threatlevel/2009/01/fannie/

Fannie Mae Logic Bomb. . .

On the afternoon of Oct. 24, he was told he was being fired because of a scripting error he'd made earlier in the month, but he was allowed to work through the end of the day.

Fannie Mae Logic Bomb. . .

Five days later, another Unix engineer at the data center discovered the malicious code hidden inside a legitimate script that ran automatically every morning at 9:00 a.m. Had it not been found, the FBI says the code would have executed a series of other scripts designed to block the company's monitoring system, disable access to the server on which it was running, then systematically wipe out all 4,000 Fannie Mae servers, overwriting all their data with zeroes.

Fannie Mae Logic Bomb. . .

"This would also destroy the backup software of the servers making the restoration of data more difficult because new operating systems would have to be installed on all servers before any restoration could begin," wrote Nye.
As a final measure, the logic bomb would have powered off the servers.

Fannie Mae Logic Bomb. . .

The trigger code was hidden at the end of the legitimate program,
separated by a page of blank lines. Logs showed that Makwana had logged onto the server on which the logic bomb was created in his final hours on the job.

Fannie Mae Logic Bomb...

Source:

http://www.thetechherald.com/articles/Fannie-Mae-logic-bomb-creator-found-guilty/11557

Facts in the case prove that Fannie Mae had strong logging processes. The initial affidavit says Makwana was singled out as the person who wrote the malicious script because logs revealed his username was the last to access the system where the logic bomb was located. In addition, he was the last to access the malicious file itself, and IP address assignment was used to show he did all of this from his company laptop.

Defending Against Insider Attacks

- No single points of failure — administrators shouldn't be allowed to access important systems alone.

Defending Against Insider Attacks

- No single points of failure — administrators shouldn't be allowed to access important systems alone.
- Code walk-throughs — can the programmer explain the logic bomb?

Defending Against Insider Attacks

- No single points of failure — administrators shouldn't be allowed to access important systems alone.
- Code walk-throughs — can the programmer explain the logic bomb?
- Use software engineering tools.

Defending Against Insider Attacks

- No single points of failure — administrators shouldn't be allowed to access important systems alone.
- Code walk-throughs — can the programmer explain the logic bomb?
- Use software engineering tools.
- Use least privilege principle — no one should have more privileges than they need to do their job.

Defending Against Insider Attacks. . .

- Physically secure systems.

Defending Against Insider Attacks. . .

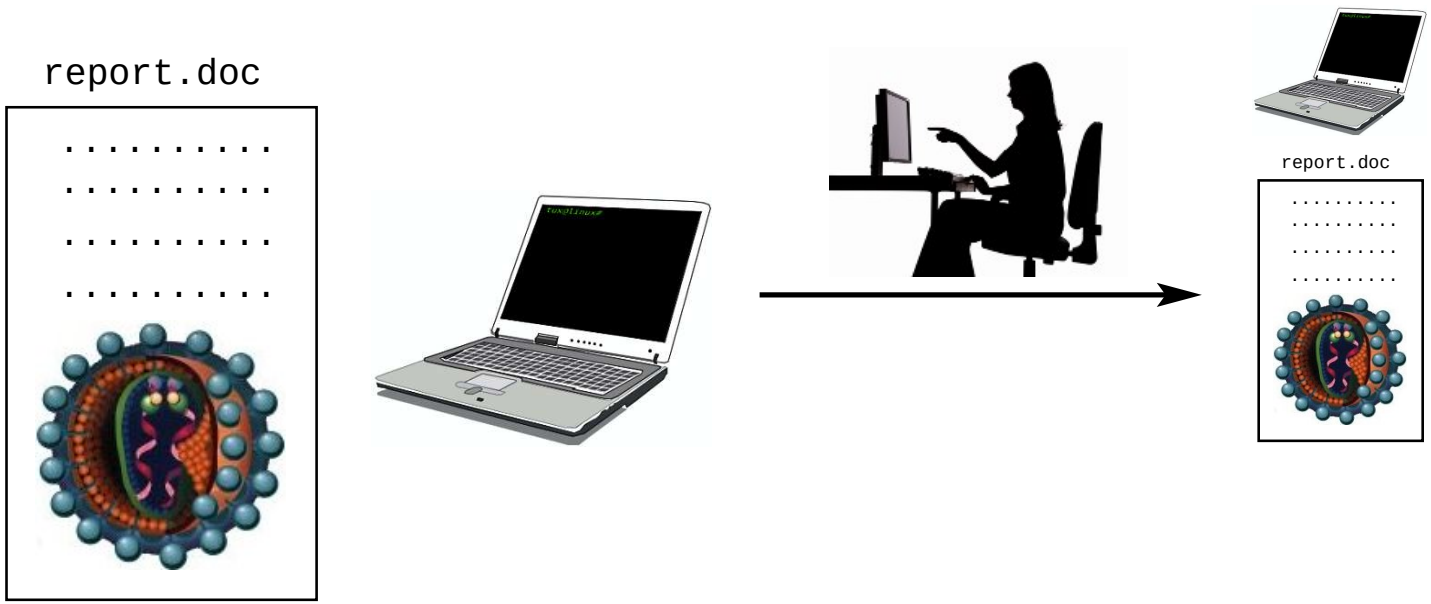
- Physically secure systems.
- Monitor employee behavior — watch out for disgruntled administrators.

Defending Against Insider Attacks. . .

- Physically secure systems.
- Monitor employee behavior — watch out for disgruntled administrators.
- Limit new software installations.

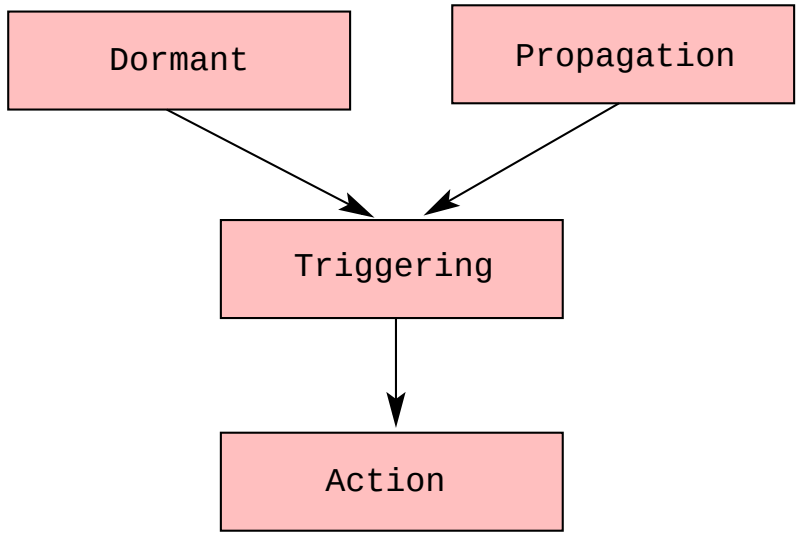
Outline

Computer Viruses



- Viruses
  1. are self-replicating;
  2. attach themselves to other files;
  3. require user assistance to replicate.

Computer Viruses: Phases



- Dormant — lay low, avoid detection.
- Propagation — infect new files and systems.
- Triggering — decide to move to action phase
- Action — execute malicious actions, the payload.

Virus Types

- Program/File virus:
  - Attaches to: program object code.
  - Run when: program executes.
  - Propagates by: program sharing.

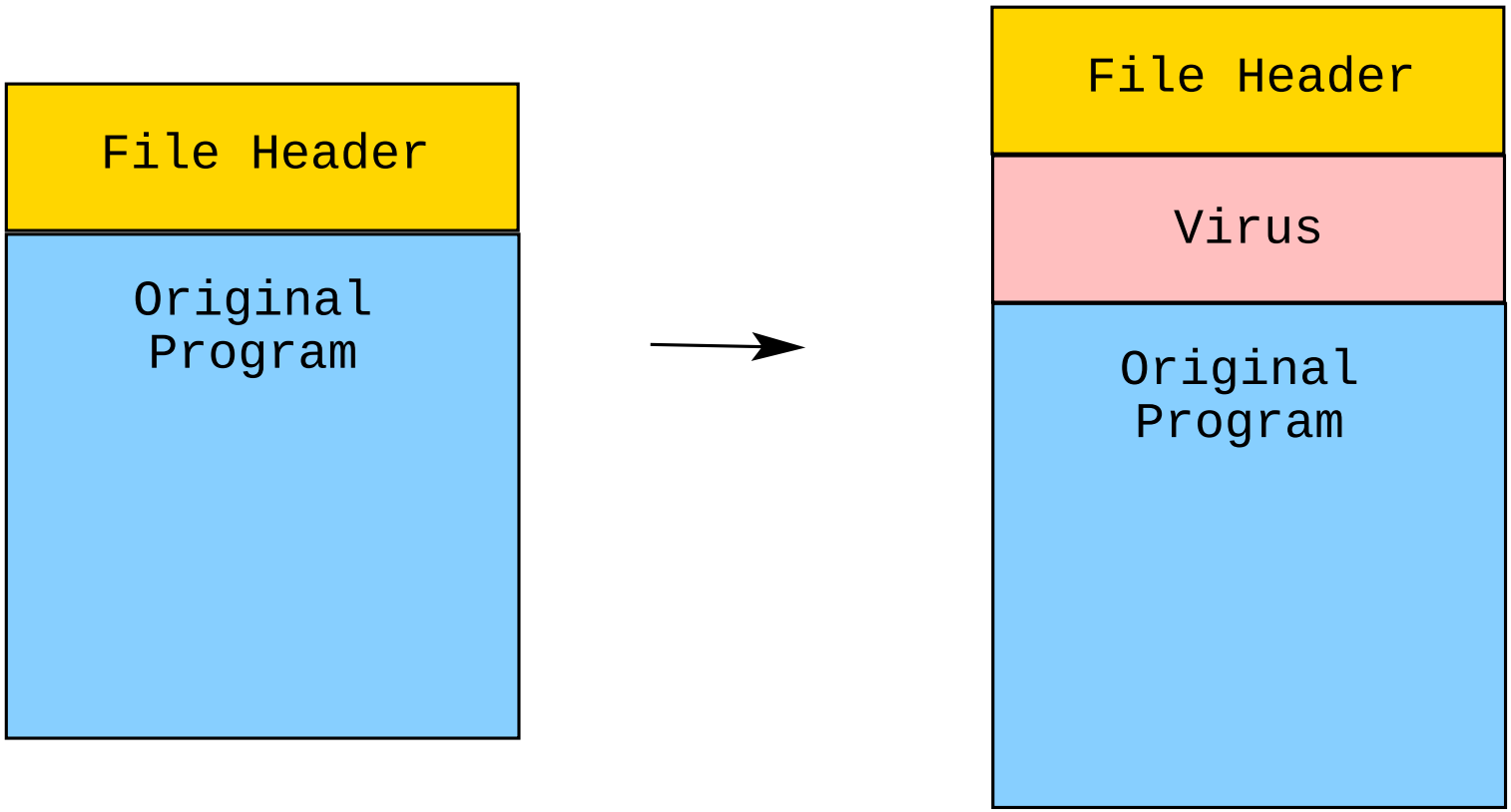
Virus Types

- Program/File virus:
  - Attaches to: program object code.
  - Run when: program executes.
  - Propagates by: program sharing.
- Doocument/Macro virus:
  - Attaches to: document (.doc,.pdf,...).
  - Run when: document is opened.
  - Propagates by: emailing documents.

Virus Types

- Program/File virus:
  - Attaches to: program object code.
  - Run when: program executes.
  - Propagates by: program sharing.
- Doocument/Macro virus:
  - Attaches to: document (.doc,.pdf,...).
  - Run when: document is opened.
  - Propagates by: emailing documents.
- Boot sector virus:
  - Attaches to: hard drive boot sector.
  - Run when: computer boots.
  - Propagates by: sharing floppy disks.

Computer Viruses: Propagation



Example: Jerusalem

- Target: DOS executables.
- Trigger: Friday the 13th.
- Payload: Deletes files.
- Propagation: Memory resident, infects executed programs.
- http://www.youtube.com/watch?v=u3k-8kJ54sg

Example: Melissa

- Target: MS Word Macro virus.
- Trigger: User opens document.
- Payload/Propagation: Emails infected documents to 50 contacts.
- http://www.youtube.com/watch?v=hu-rhzOgExg

Example: Elk Cloner

- Target: Apple II boot sector.
- Payload: Prints poem every 50th time the program is rebooted.

Elk Cloner: The program with a personality
It will get on all your disks
It will infiltrate your chips
Yes, it's Cloner!
It will stick to you like glue
It will modify RAM too
Send in the Cloner!

Example: Elk Cloner...

At age 15 Richard J Skrenta, as a high school student at Mt. Lebanon High School, wrote the Elk Cloner virus that infected Apple II machines. It is widely believed to have been one of the first large-scale self-spreading personal computer viruses ever created. Skrenta was already distrusted by his friends because, in sharing computer games and software, he would often alter the floppy disks to shut down or display taunting on-screen messages. Because his friends no longer trusted his disks, Skrenta thought of methods to alter floppy disks without physically touching them.

Example: Elk Cloner...

During a winter break [] Skrenta discovered how to launch the messages automatically on his Apple II computer. He developed what is now known as a boot sector virus, and began circulating it in early 1982 among high school friends and a local computer club.

Source: http://en.wikipedia.org/wiki/Elk_Cloner

Rich Skrenta: https://en.wikipedia.org/wiki/Rich_Skrenta

Example: Sality

- Target: Windows executable files.
- Propagation: Infects other local executable files.
- Obscures its entry point.
- Downloads and executes other malware.
- Creates peer-to-peer botnet.
- Disables security software.
- Injects itself into running processes to make sure it remains on the computer.

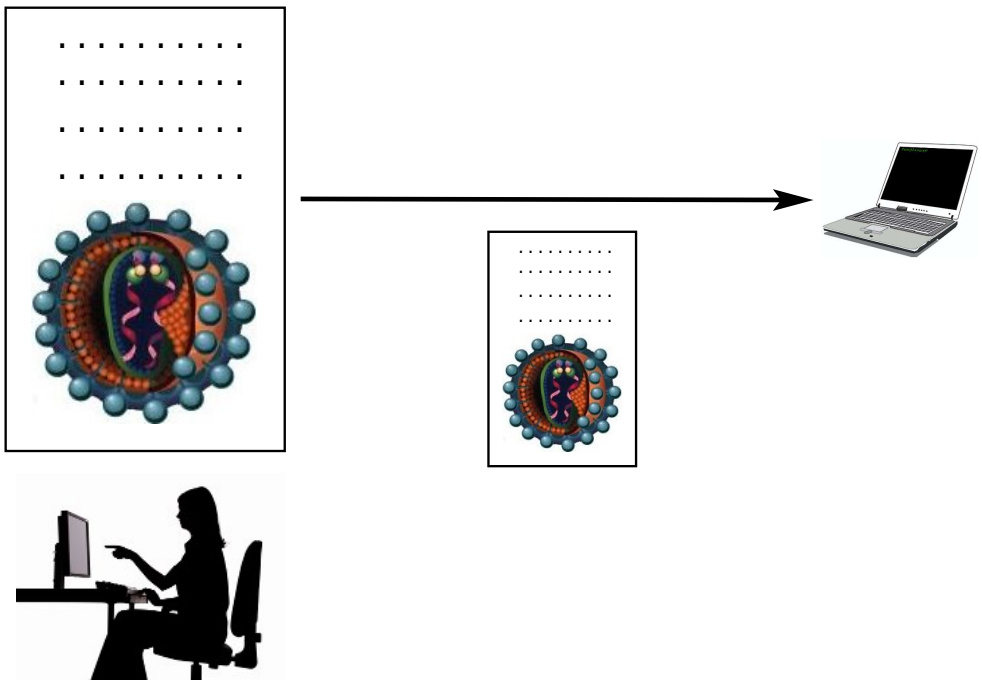
Antivirus software

Definition (Antivirus software)

Antivirus software, or anti-virus software (abbreviated to AV software), also known as anti-malware, is a computer program used to prevent, detect, and remove malware.
Modern antivirus software can also protect from other types of malware, such as ransomware, backdoors, trojans, adware and spyware.

Source: https://en.wikipedia.org/wiki/Antivirus_software

Defenses Against Virus Infection



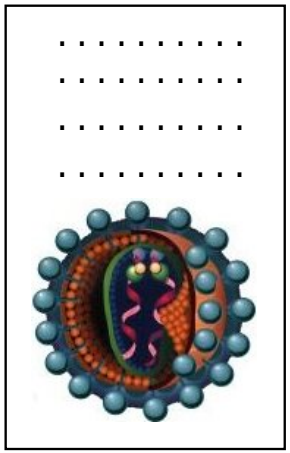
Signature Database

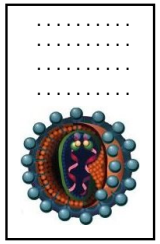
Melissa	/^[54]+/
...	...
...	...

Anti Virus Engine

Heuristics
File scanners
Memory scanner
Signature checker
Emulators

Defenses Against Virus Infection





Updates

Signature Database

Melissa	/^[54]+/
...	...
...	...

Anti Virus Engine

Heuristics
File scanners
Memory scanner
Signature checker
Emulators

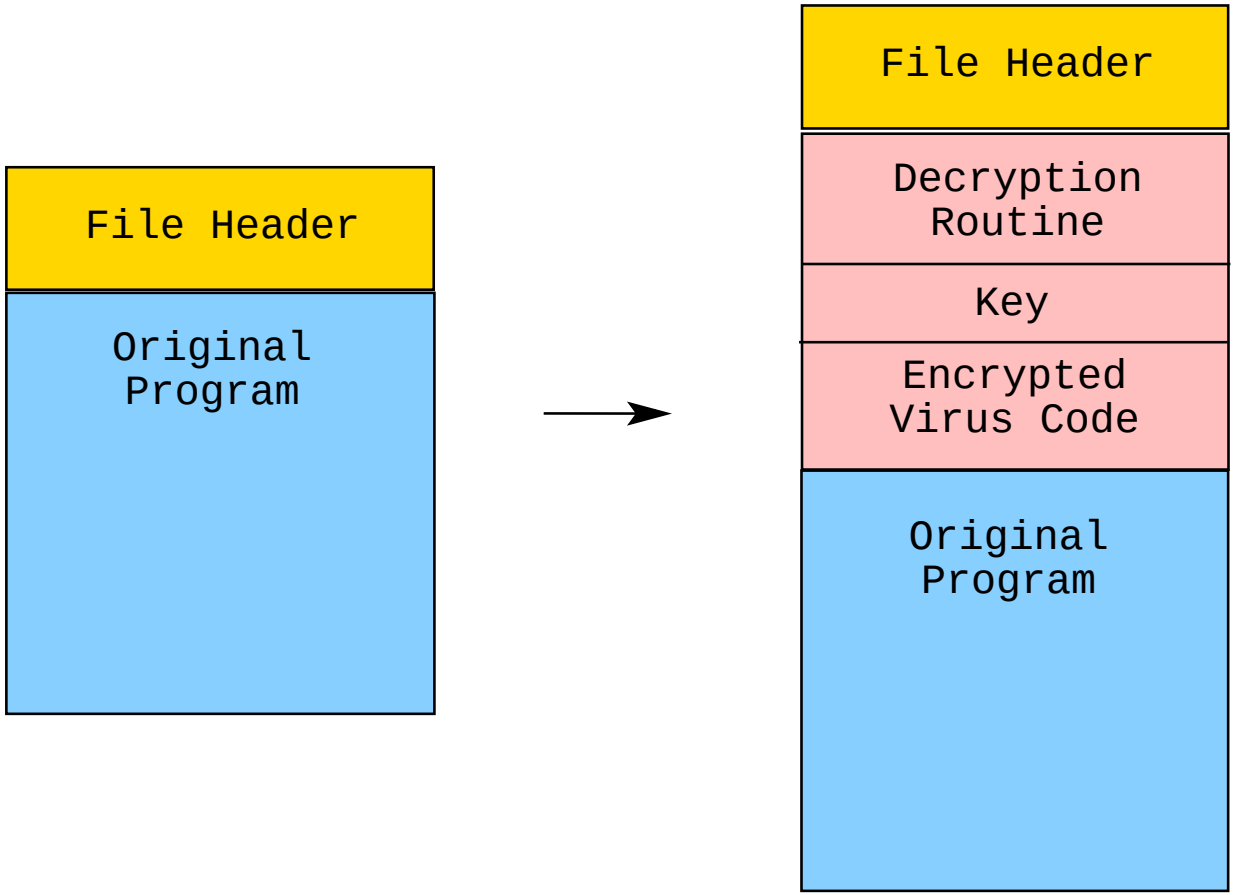
Defenses Against Virus Infection...

- Antivirus or AV: software that checks for
- Signatures: Regular expressions over the virus code used to detect if files have been infected.
- Checking can be done
  1. periodically over the entire filesystem;
  2. whenever a new file is downloaded.

Virus Countermeasures

- Viruses need to protect themselves against detection.
- This means hiding any distinguishing features, making it hard to construct signatures.

Virus Countermeasures: Encryption

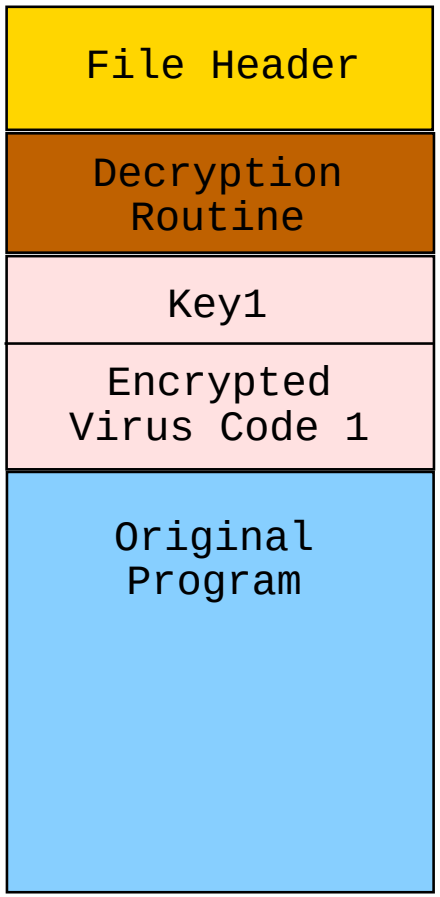


- By encrypting its payload, the virus hides its distinguishing features.
- Encryption: often just xor with a constant.

Virus Countermeasures: Encryption...

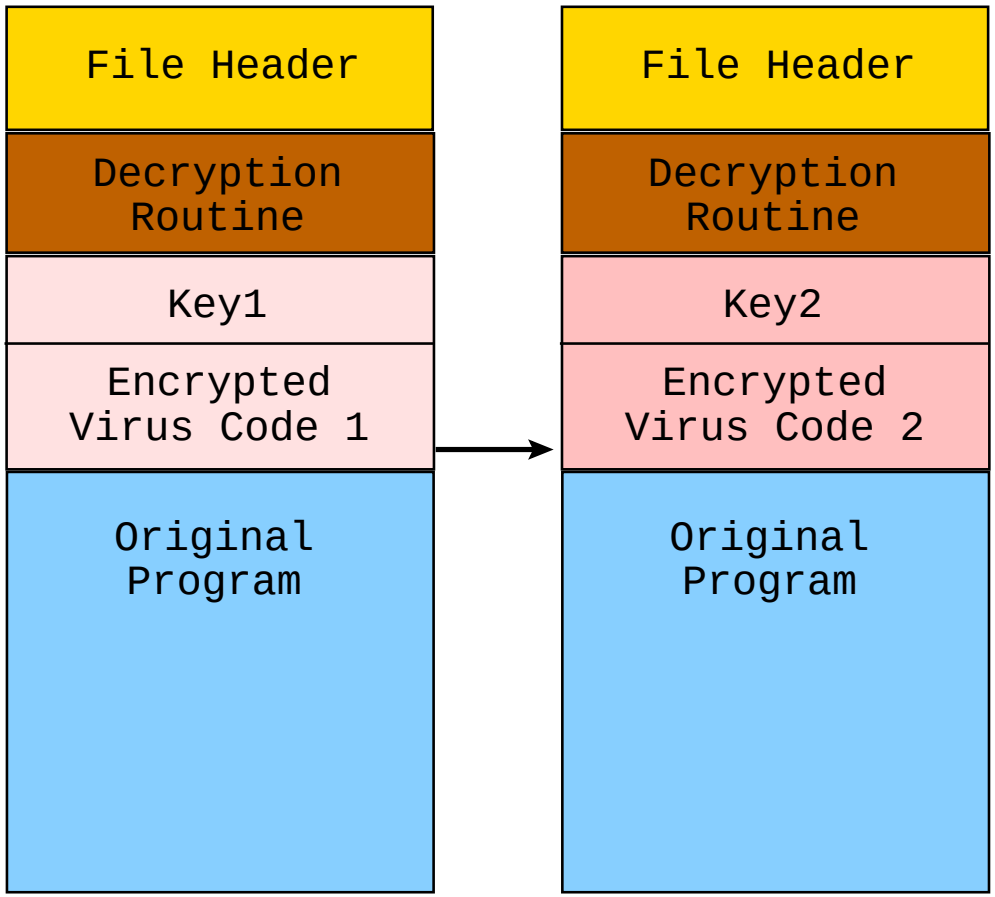
```
... file header ...
int key = 0xFEEDFACE;
char virus[] = {0x42,0x99,0xff,...}
char payload[1000];

void decrypt_virus() {
    for(i=0=;i<length(virus);i++) {
        payload[i] = virus[i] ^ key
    }
}

... original code ...

int main() {
    decrypt_virus();
    (*payload)();
    ... original main ...
}
```

Virus Countermeasures: Polymorphism



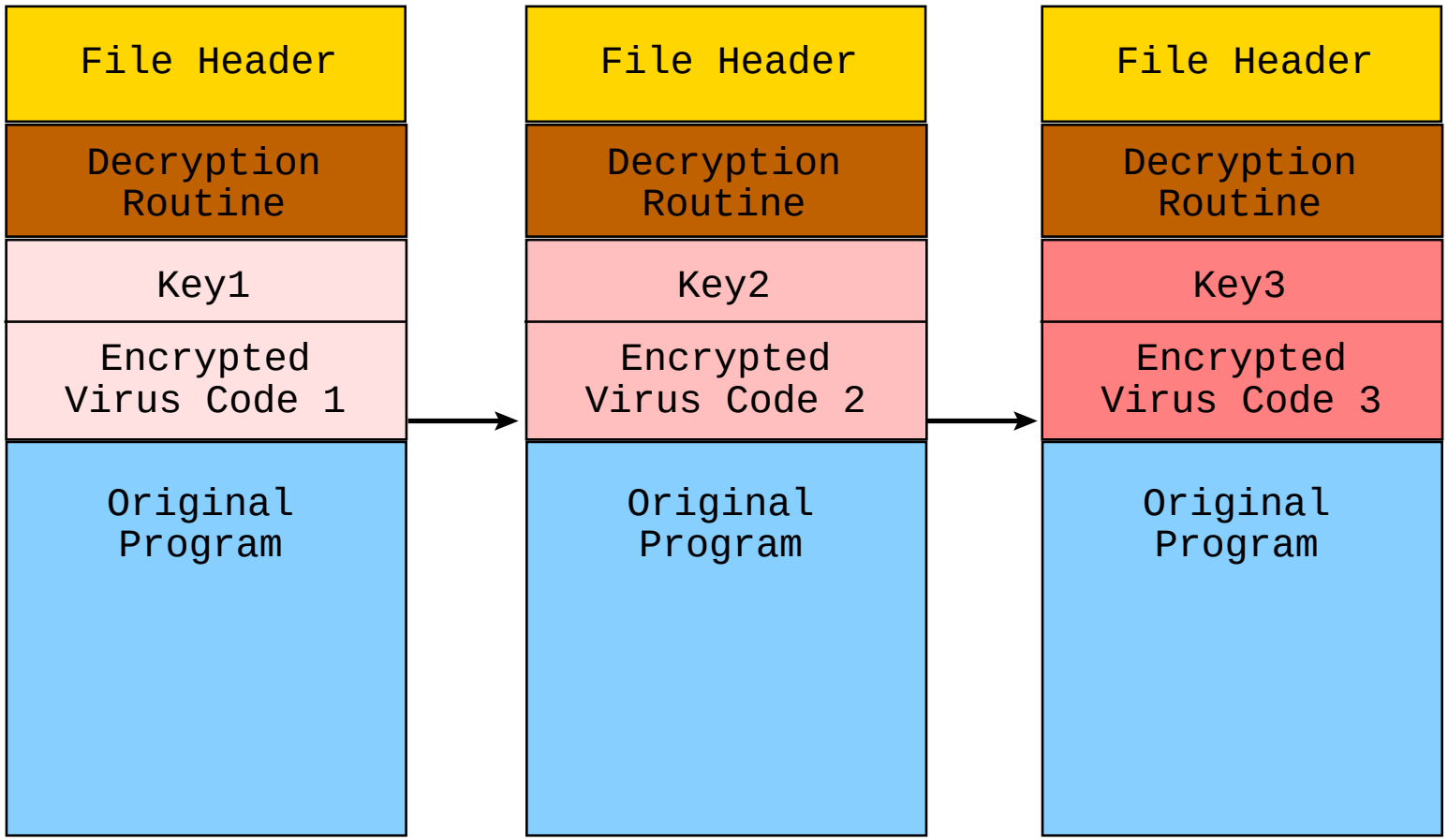
- Each variant is encrypted with a different key.
- The decryption routine itself, however, can be used to create a signature!

Virus Countermeasures: Polymorphism



- Each variant is encrypted with a different key.
- The decryption routine itself, however, can be used to create a signature!

Virus Countermeasures: Polymorphism



- Each variant is encrypted with a different key.
- The decryption routine itself, however, can be used to create a signature!

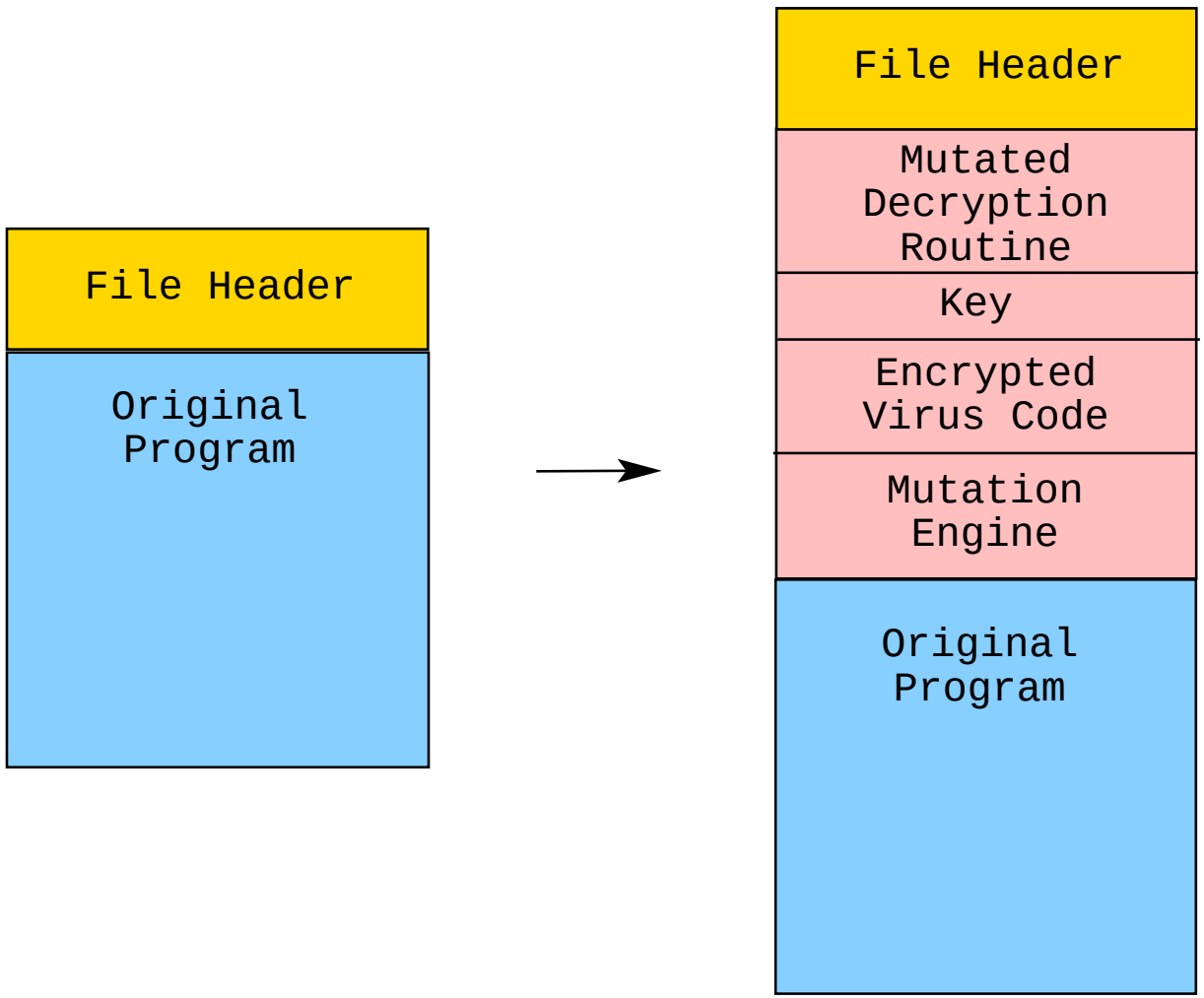
Virus Countermeasures: Polymorphism...

```
int my_key = 0xFEEDFACE;
void decrypt() {... my_key...}

char[] decrypted_virus_payload () = {
    do_bad_stuff();
    for(every file on system) {
        int new_key = rand();
        payload1 = encrypt payload with new_key;
        new_code = [new_key; decrypt; payload];
        insert new_code in file;
    }
}

int main() {
    decrypt();
    (*payload)();
    ... original main ...
}
```

Virus Countermeasures: Metamorphism

- To prevent easy creation of signatures for the decryption routine, metamorphic viruses will mutate the decryptor, for each infection.
- The virus contains a mutation engine which can modify the decryption code while maintaining its semantics.
- Mutation engine = (simple) obfuscator!

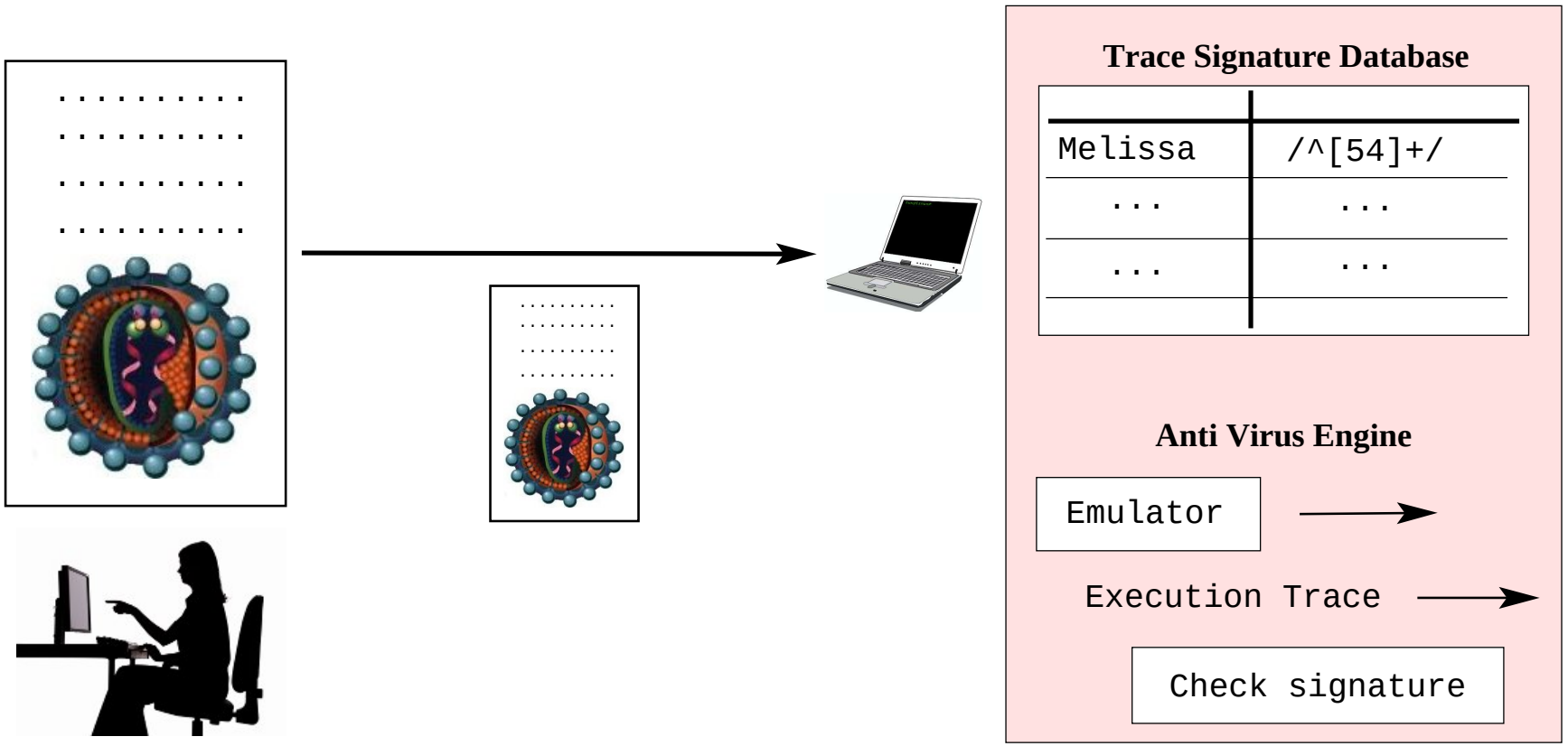
Virus Countermeasures: Metamorphism...



Virus Countermeasures: Metamorphism...

```
char* mutate(char * v) { ... obfuscate v! ... }

char[] decrypted_virus_payload = {
  do_bad_stuff();
  for(every file on system) {
     int key = rand();
     decrypt1 = mutate(decrypt);
     virus1 = encrypt [payload;mutate] with key;
     code = [key; decrypt1; virus1];
     insert code in file;
  }
}

int main() {
   decrypt();
   (*payload)();
}
```

Anti Virus: Emulation



- To counter metamorphism, virus detectors can run the virus in an emulator.

Anti Virus: Emulation...

- The emulator gathers a trace of the execution.
- A virus signature is then constructed over the trace.
- This makes it easier to ignore garbage instructions the mutation engine may have inserted.

Quiz

- What is the difference between an encryption and a polymorphic virus?

Quiz

- What is the difference between an encryption and a polymorphic virus?
- What is the difference between a polymorphic and a metamorphic virus?

Quiz

- What is the difference between an encryption and a polymorphic virus?
- What is the difference between a polymorphic and a metamorphic virus?

Exam Problem

- You have discovered a new virus that embeds itself in a host program P like on the next slide.

P

$$K = \langle K_0, K_1, K_2, K_3, K_4, K_5 \rangle$$

.....

$$C^{obf} = \begin{bmatrix} C_0 \oplus K_0, & C_1 \oplus K_1, & C_2 \oplus K_2, & C_3 \oplus K_3, & C_4 \oplus K_4, & C_5 \oplus K_5, \\ C_6 \oplus K_0, & C_7 \oplus K_1, & C_8 \oplus K_2, & C_9 \oplus K_3, & C_{10} \oplus K_4, & C_{11} \oplus K_5, \\ C_{12} \oplus K_0, & C_{13} \oplus K_1, & C_{14} \oplus K_2, & C_{15} \oplus K_3, & C_{16} \oplus K_4, & C_{17} \oplus K_5, \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

```
void run_virus() {
    extract K from P;
    extract C^obf from P;
    for(i=0; i<6·n; i++)
        C_i ← C_i^obf ⊕ K_{i mod 6};
    jump to the beginning of the virus
        code C_0;
}
```

Exam Problem...

- C^{obf} is an obfuscated version of the virus body C,
- K is a random secret key.
- `run_virus()` is a function that loads, de-obfuscates, and executes the virus body.
- All are inserted at random positions in the host program P.

Exam Problem...

- You have learned the following facts about the virus:
  1. The virus body C is a multiple of 6 bytes: $C = \langle C_0, C_1, C_2, C_3, \ldots, C_{6 \cdot n-1} \rangle$
  2. K is 6 bytes long, and different for each host program.
  3. C^{obf} is obtained by XOR-ing the cleartext virus body C with K, where K is repeated the necessary number of times.

Exam Problem...

- Assume that you have obtained the cleartext body C of the virus and a set of programs that you suspect are infected.
- You want to write a virus scanner that returns `true` if C occurs in host program P (consisting of bytes $\langle P_0, P_1, P_2, \ldots \rangle$) and `false` otherwise:

```
boolean detect(
    C = ⟨C_0, C_1, C_2, ...⟩, P = ⟨P_0, P_1, P_2, ...⟩) {
    ...
}
```

Exam Problem...

- You can't look for a signature of `run_virus()` in P (because it's too similar to code that occurs in normal programs).
- You can't examine `run_virus()` for the location of C^{obf} (because you don't have tools that are precise enough to analyze the `run_virus()`'s binary code).
- You can't execute P to look for the behavior of `run_virus()` or the virus body (because it would take too much time).
- Your algorithm should be fast, i.e. polynomial-time in the length of P.

Outline

Trojan Horses

```
main(){
   play_game();
   send_spam();
}
```

- A trojan horse is a program that appears to perform a useful task, but, in addition, performs a malicous task.
- Example:
  - Useful task: A better `ls`.
  - Malicious task: Exfiltrate company secrets.

Trojan Horses: The AIDS Trojan

- A disk that said it was an AIDS Information Introductory Diskette.
- When the boot count reaches 90, AIDS encrypts the names of all files.
- The user is asked to *renew the license*.
- To recover the files the user needs to send $189 to a P.O. box in Panama.

The AIDS Trojan License Agreement

Source: http://en.wikipedia.org/wiki/AIDS_(trojan_horse)

If you install [this] on a microcomputer... then under terms of this license you agree to pay PC Cyborg Corporation in full for the cost of leasing these programs... In the case of your breach of this license agreement, PC Cyborg reserves the right to take legal action necessary to recover any outstanding debts payable to PC Cyborg Corporation and to use program mechanisms to ensure termination of your use...

The AIDS Trojan License Agreement...

These program mechanisms will adversely affect other program applications... You are hereby advised of the most serious consequences of your failure to abide by the terms of this license agreement; your conscience may haunt you for the rest of your life... and your [PC] will stop functioning normally... You are strictly prohibited from sharing [this product] with others...

Outline

Readings and References

- Chapter 4 in *Introduction to Computer Security*, by Goodrich and Tamassia.