CSCI262 : System Security

Week 6: Malware (Malicious Software)

Schedule

- What is Malware?
- Classifications
- Propagation Mechanisms
- Payload Actions
- Countermeasures

Malware

- NIST SP 800-83 (Guide to Malware Incident Prevention and Handling for Desktops and Laptops, July 2013):

“Malware (Malicious software) is a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of victim’s data, applications or operating systems or otherwise annoying or disrupting the victim.”

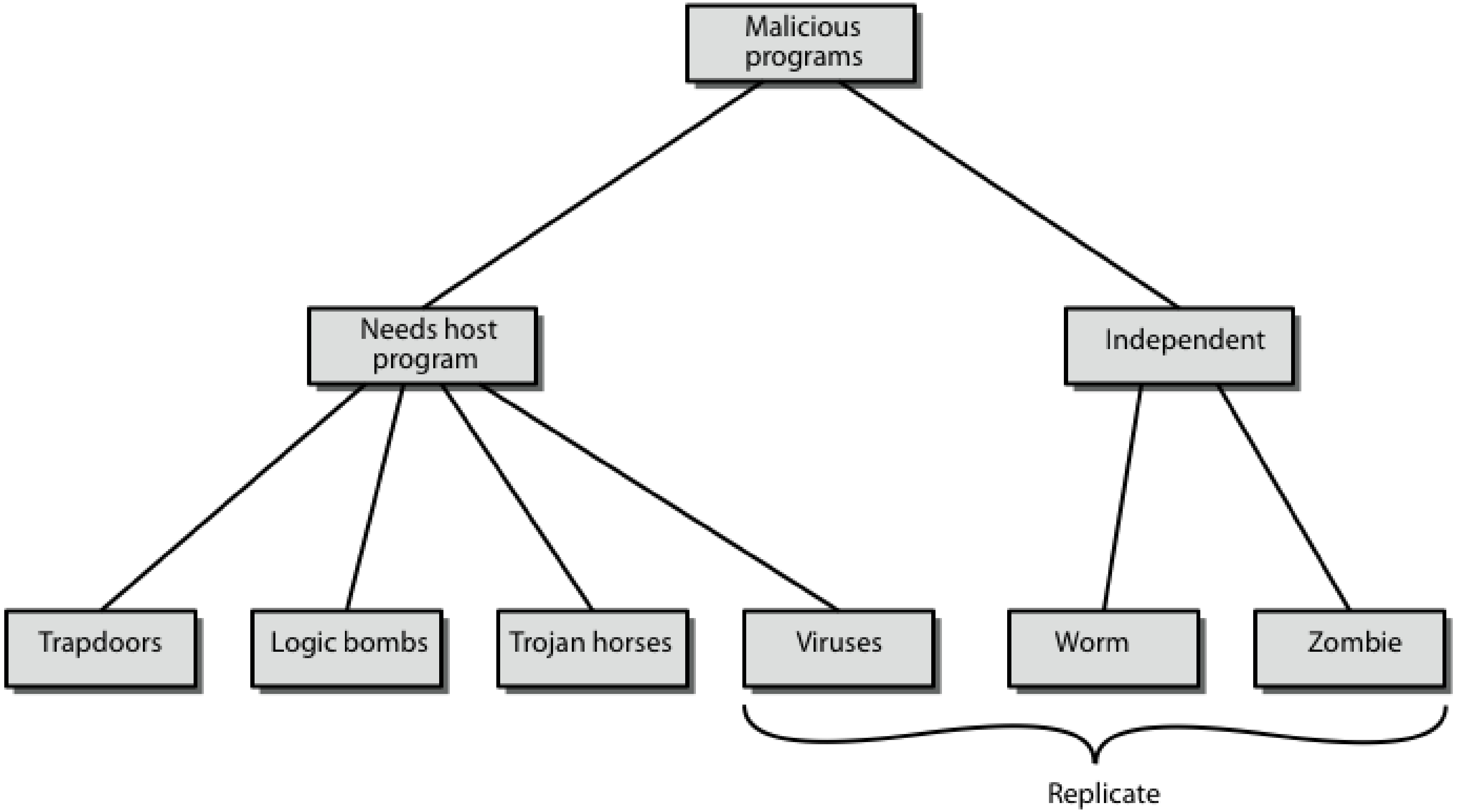
- Malicious mobile code is malware that is furthermore designed to move:
    - From computer to computer and network to network

Some numbers

- According to Malware statistics 2023 (https://dataprot.net/statistics/malware-statistics/) :
  - There are more than 1 billion malware programs out there
  - 560,000 new pieces of malware are detected every day
  - SonicWall has registered more than 3.2 billion malware attacks in the first half of 2020.
  - Symantec: In 2020, the number of detected malware variants rose by 62%.
  - Google: Each week, Google detects 50 websites containing malware.
- Over 30 million new malware samples found in 2022 as cyber threats evolve
  - https://atlasvpn.com/blog/over-30-million-new-malware-samples-found-in-2022-as-cyber-threats-evolve

Classification

- Early approach



From Stallings (Cryptography & Network Security)

- In this lecture, we classify malware into two broad categories:
- How it spreads/propagates
  - infection of existing executable or interpreted content by viruses that is subsequently spread to other systems
  - exploit of software vulnerabilities either locally or over a network by worms or drive-by-downloads to allow the malware to replicate
  - social engineering attacks that convince users to bypass security mechanisms to install Trojans, or to respond to phishing attacks
- The actions/payloads it performs
  - corruption of system or data files
  - theft of service in order to make the system a zombie agent of attack as part of a botnet
  - theft of information from the system, especially of logins, passwords, or other personal details by keylogging or spyware programs
  - stealthing where the malware hides its presence on the system from attempts to detect and block it

Viruses

- A computer virus, first appeared in the early 1980s, is a piece of software that can “infect” other programs, or indeed any type of executable content, by modifying them.
  - Injecting the original code with a routine to make copies of the virus code, which can then go on to infect other content
- In early years, viruses dominated the malware scene due to lack of user authentication and access control
- Tighter access controls on modern operating systems resulted in development of macro viruses that exploited the active content supported by some documents types, such as Microsoft Word or Excel files, or Adobe PDF documents

Viruses “in the wild”

- When a virus is in general circulation we refer to it as being "in the wild".
- This implies computing environments outside of
  - … the development environment where the virus was created and tested.
  - … the collections of antivirus vendors, researchers, and collectors.
- “For a virus to be considered “In the Wild”, it must be spreading as a result of normal day-to-day operations on and between the computers of unsuspecting users.” Ducklin in 'Counting Viruses'

- Viruses may not stay in the wild, but could end up disappearing because of good antivirus techniques and technology updates.
    - For example, a virus for 360kB floppy disks (such as the Brain virus, the first for IBM PC’s) would not normally be a problem now.
    - It is also possible for viruses to return to the wild.

- The Wild List was started in 1993 by Joe Wells, with the associated organization founded in April 1996 by Joe Wells & Sarah Gordon.
  - http://www.wildlist.org
  - It includes an extensive list of currently Wild viruses.
  - Virus scanners should be able to detect all of the viruses in the Wild, as well as many older ones.

Virus logic

- A virus is often quite a simple program, not just conceptually but also in implementation.
- A direct action virus can be modelled in terms of an algorithm such as the following:

```
begin
  Look for (one or more infectable objects)
    If (none found)
    then
      exit
    else (infect object or objects).
    endif
end
```

- Direct action viruses are only active when an infected object is active.

- A lot of viruses install themselves into the memory of the host computer when the original virus program is executed.
- This means that even after the original virus program is closed, new objects can be infected without having to run anything else.
- These are referred to as memory resident viruses.
- Hybrid viruses are both direct action and memory resident.

Viruses’ components

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

Virus lifetime phases

- **Dormant phase**: The virus is idle. It may be waiting for a trigger before propagation begins. Not all viruses have this.
- **Propagation phase**: The virus places a copy of itself into other programs or into certain system areas on the disk.
- **Triggering phase**: The virus is activated to perform the function.
- **Execution phase**: The function is performed.

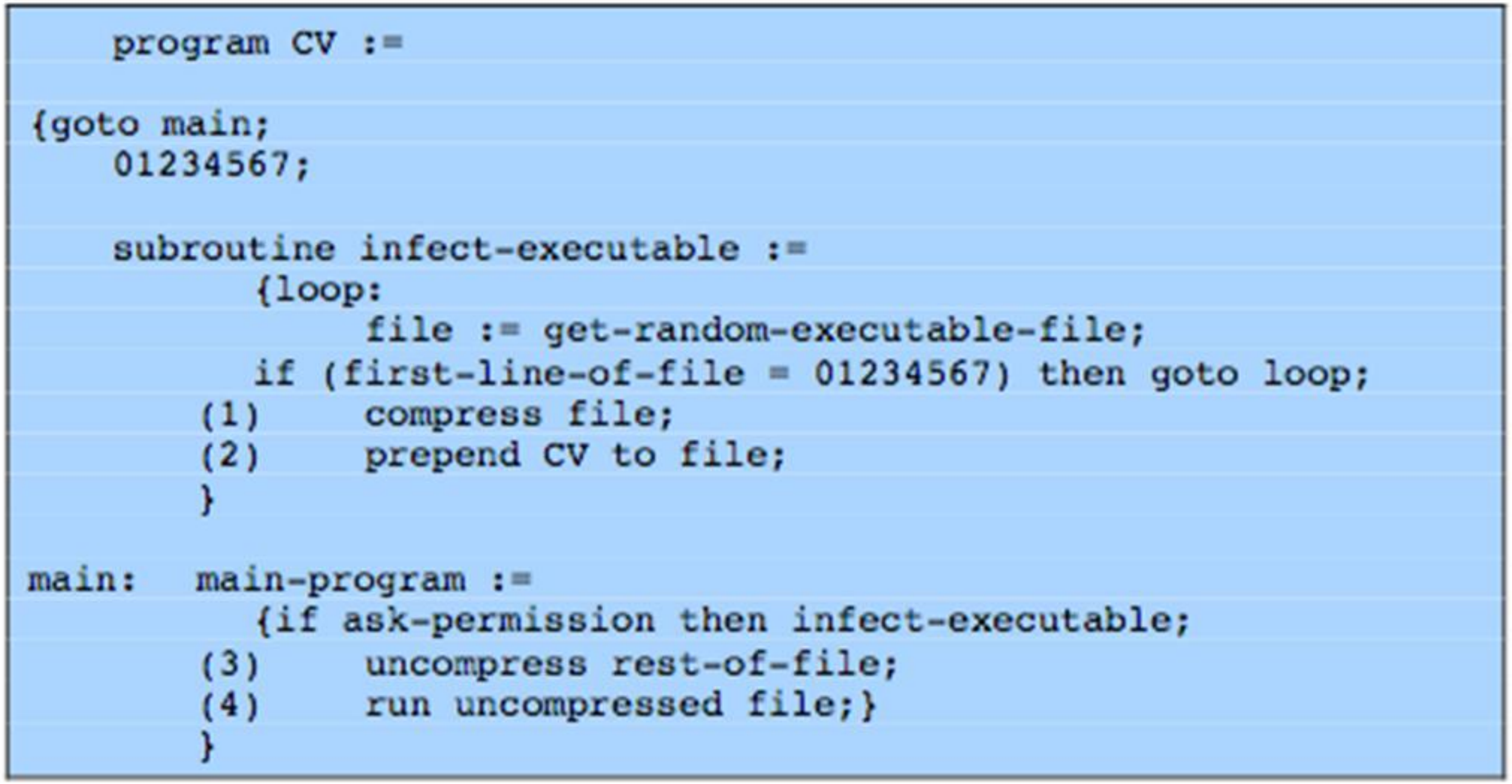
Classification of Viruses

- Viruses can be classified **according to the target**
    - **Boot sector infector** : Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.
    - **File infector**: Infects files that the operating system or shell consider to be executable.
    - **Macro virus**: Infects files with macro or scripting code that is interpreted by an application. Example in [SB18]: Melissa
    - **Multipartite virus**: Infects files in multiple ways, infecting multiple type of files

- … and according to the method of concealment
  - **Stealth virus:** designed to hide itself from detection by anti-virus software. Thus, the entire virus, not just a payload, is hidden.
  - **Encrypted virus**: use encryption to obscure its content
  - **Polymorphic virus**: change form each time they are inserted into another program.
  - **Metamorphic virus**:
    - a higher order of polymorphic viruses.
    - Not only do they change in form between transitions they can be completely re-written

Stealth viruses

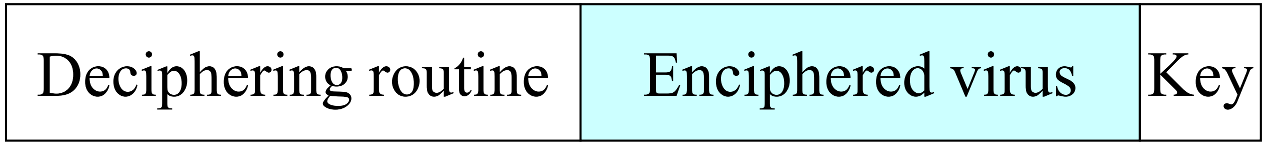
- Viruses with this technology explicitly try to hide all of themselves from detection.
- One typical approach is to include compression.
  - Detecting that a file has changed by checking the length won’t work anymore.
- The virus can also include placing “intercept logic” to capture dangerous queries (from the viewpoint of the virus) and provide acceptable responses.



```
    program CV :=

{goto main;
    01234567;

    subroutine infect-executable :=
          {loop:
               file := get-random-executable-file;
          if (first-line-of-file = 01234567) then goto loop;
        (1)    compress file;
        (2)    prepend CV to file;
        }

main:   main-program :=
          {if ask-permission then infect-executable;
        (3)    uncompress rest-of-file;
        (4)    run uncompressed file;}
        }
```

Encrypted Viruses

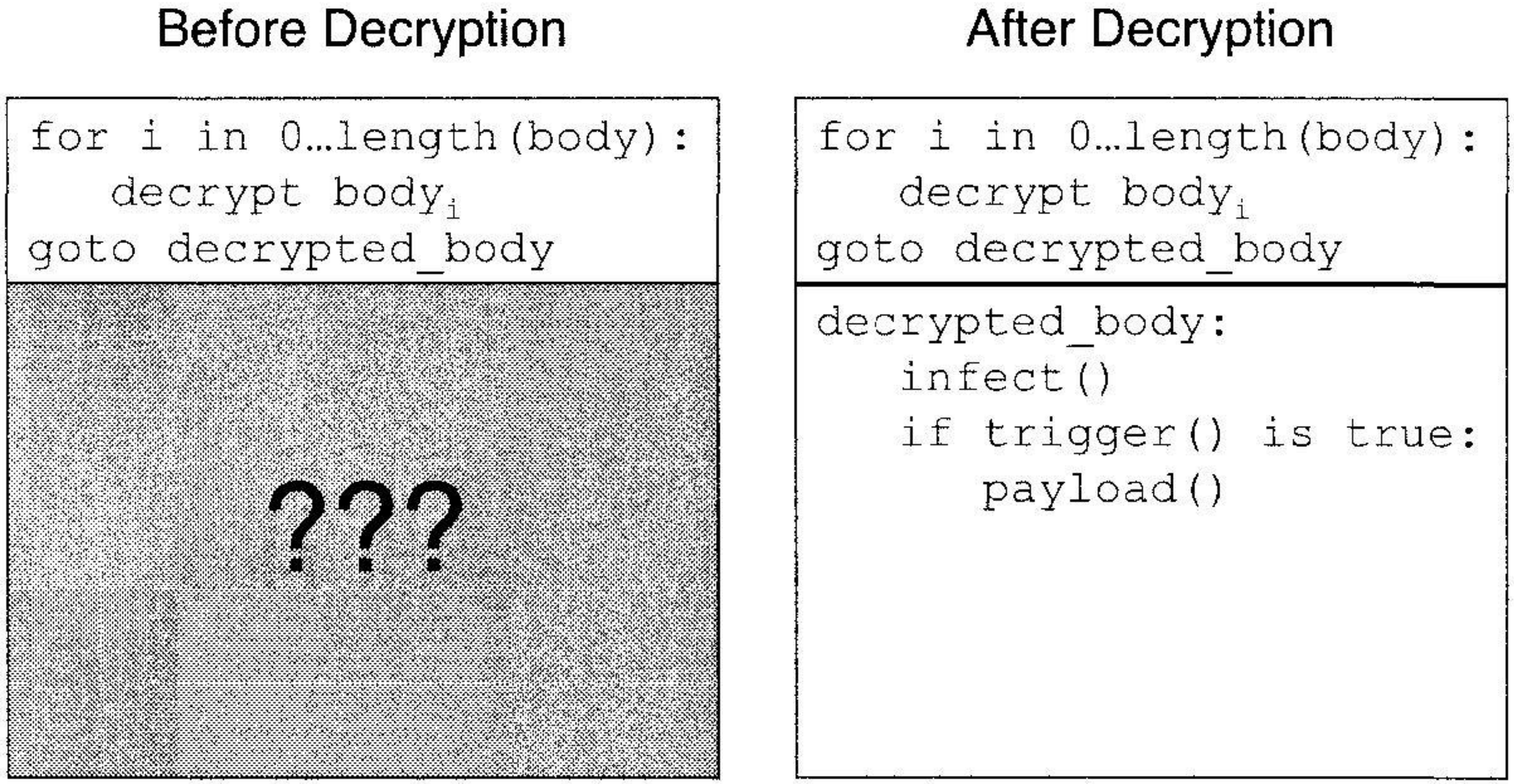
- Viruses which are encrypted with a cipher.
- Why?
  - To avoid detection. The virus code is hidden!
- How?
  - The virus code is encrypted, except for the decryption routine and key.
- Does it work?
  - We will look at detection a little later.



Deciphering routine

Enciphered virus

Key



Before Decryption

```
for i in 0…length(body):
    decrypt $body_i$
goto decrypted_body
```

???

After Decryption

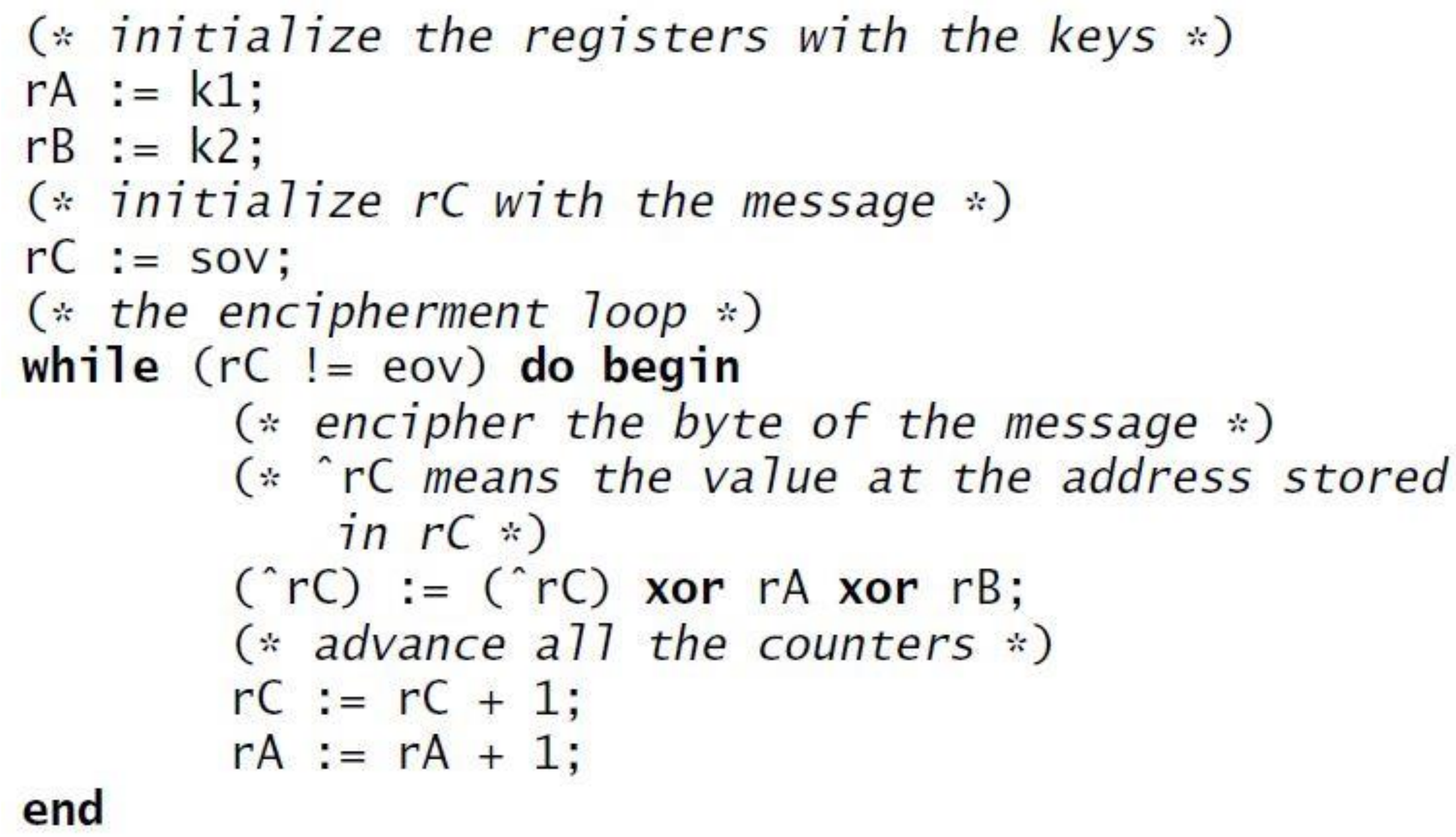
```
for i in 0…length(body):
    decrypt $body_i$
goto decrypted_body
```

```
decrypted_body:
    infect()
    if trigger() is true:
        payload()
```

Figure 3.5. Encrypted virus pseudocode

- From the book: “A Pathology of Computer Viruses” by Ferbrache.

An example of encrypted virus



```
(* initialize the registers with the keys *)
rA := k1;
rB := k2;
(* initialize rC with the message *)
rC := sov;
(* the encipherment loop *)
while (rC != eov) do begin
        (* encipher the byte of the message *)
        (* ˆrC means the value at the address stored
            in rC *)
        (ˆrC) := (ˆrC) xor rA xor rB;
        (* advance all the counters *)
        rC := rC + 1;
        rA := rA + 1;
end
```

D. Ferbrache. *A Pathology of Computer Viruses*, Springer 1992

Polymorphic Viruses

- Polymorphic viruses change the form of its decryption routine each time it inserts itself into another program.
- If the viruses is encrypted, as is fairly common, the decryption code is the segment of the virus that is changed.
  - For example, at the instruction level, all the following have exactly the same effect.

```
add 0 to operand
Logical AND 1 with operand
no operation
subtract 0 from operand
```

```
(* initialize the registers with the keys *)
rA := k1;
rA := rA + 0;                   (* random line *)
rB := k2;
rD := k1 + k2;                  (* random line *)
(* initialize rC with the message *)
rC := sov;
rC := rC + 1;                   (* random line *)
(* the encipherment loop *)
while (rC != eov) do begin
        rC := rC - 1;           (* random line *)
        (* encipher the byte of the message *)
        (* ^rC means the value at the address stored in rC *)
        (^rC) := (^rC) xor rA xor rB;
        (* advance all the counters *)
        rC := rC + 2;           (* counter incremented ... *)
        (* to handle random line X *)
        rD := rD - 0;           (* random line *)
        rA := rA + 1;
end
(* the next block does nothing *)
while (rC != sov) do begin
        rD := rD - 1;
        rC = rC - 1;
end
```

Metamorphic viruses

- These are, in some sense, a higher order of polymorphic viruses.
- Not only do they change in form between transitions they can be completely re-written.
  - They can re-write in a version suitable for executables on a different platform too.

Worms

- **Worm**: A program/virus that copies itself from one computer to another.
- Host computer worms are entirely contained in the computer they run on and use network connections only to copy themselves to other computers.
- The original may terminate itself after launching a copy on another host - sometimes called "rabbits."

Worms’ replication

- **Electronic mail or instant messenger facility**: A worm e-mails a copy of itself to other systems, or sends itself as an attachment via an instant message service
- **File sharing**: A worm either creates a copy of itself or infects other suitable files as a virus on removable media such as a USB drive
- **Remote execution capability**: A worm executes a copy of itself on another system, either by using an explicit remote execution facility or by exploiting a program flaw in a network service to subvert its operations
- **Remote file access or transfer capability**: A worm uses a remote file access or transfer service to another system to copy itself from one system to the other
- **Remote login capability**: A worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other

Worms’ Propagation

- **Propagation phase** generally performs the following functions:
  - Search for appropriate access mechanisms on other system to infect
  - Use the access mechanisms found to transfer a copy of itself to the remote system, and cause the copy to be run
- The worm may also attempt to determine whether a system has previously been infected before copying itself to the system
  - It can also disguise its presence by naming itself as a system process or using some other name that may not be noticed by a system operator

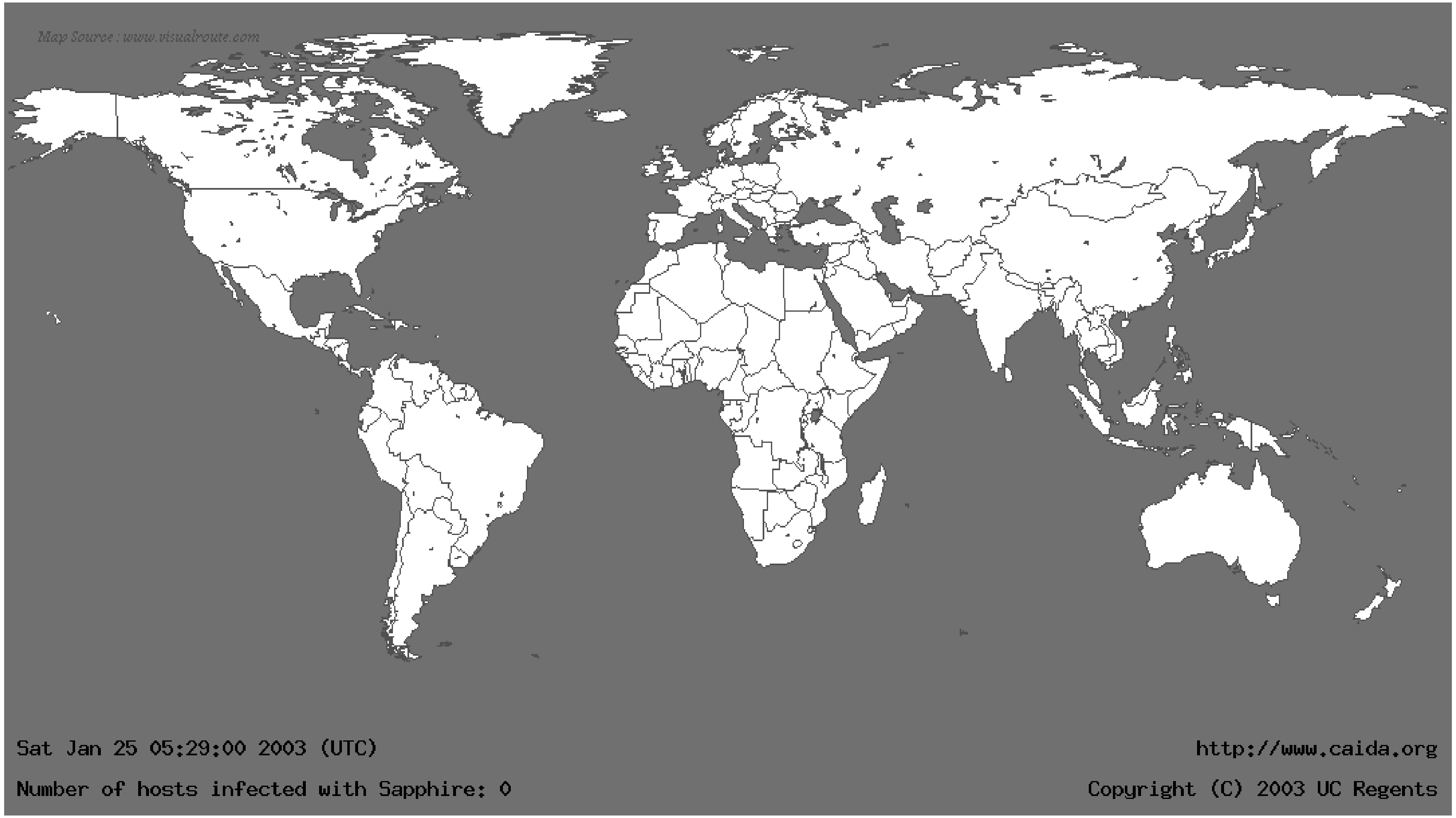
The Morris Worm

- One of the first computer worms distributed via the internet
- Was released onto the Internet by Robert Morris in 1988
- Was designed to spread on UNIX systems
- Within 24 hours, an estimated 6,000 of the approximately 60,000 computers that were then connected to the Internet had been hit.
- The worm did not damage or destroy files, but:
  - vital military and university functions slowed to a crawl
  - Emails were delayed for days.
- More detail: [SB18] and https://www.fbi.gov/news/stories/morris-worm-30-years-since-first-major-attack-on-internet-110218

SQL Slammer

- Appeared in early 2003
- Exploited a buffer overflow vulnerability in Microsoft SQL Server
  - We will learn buffer overflow later in the course

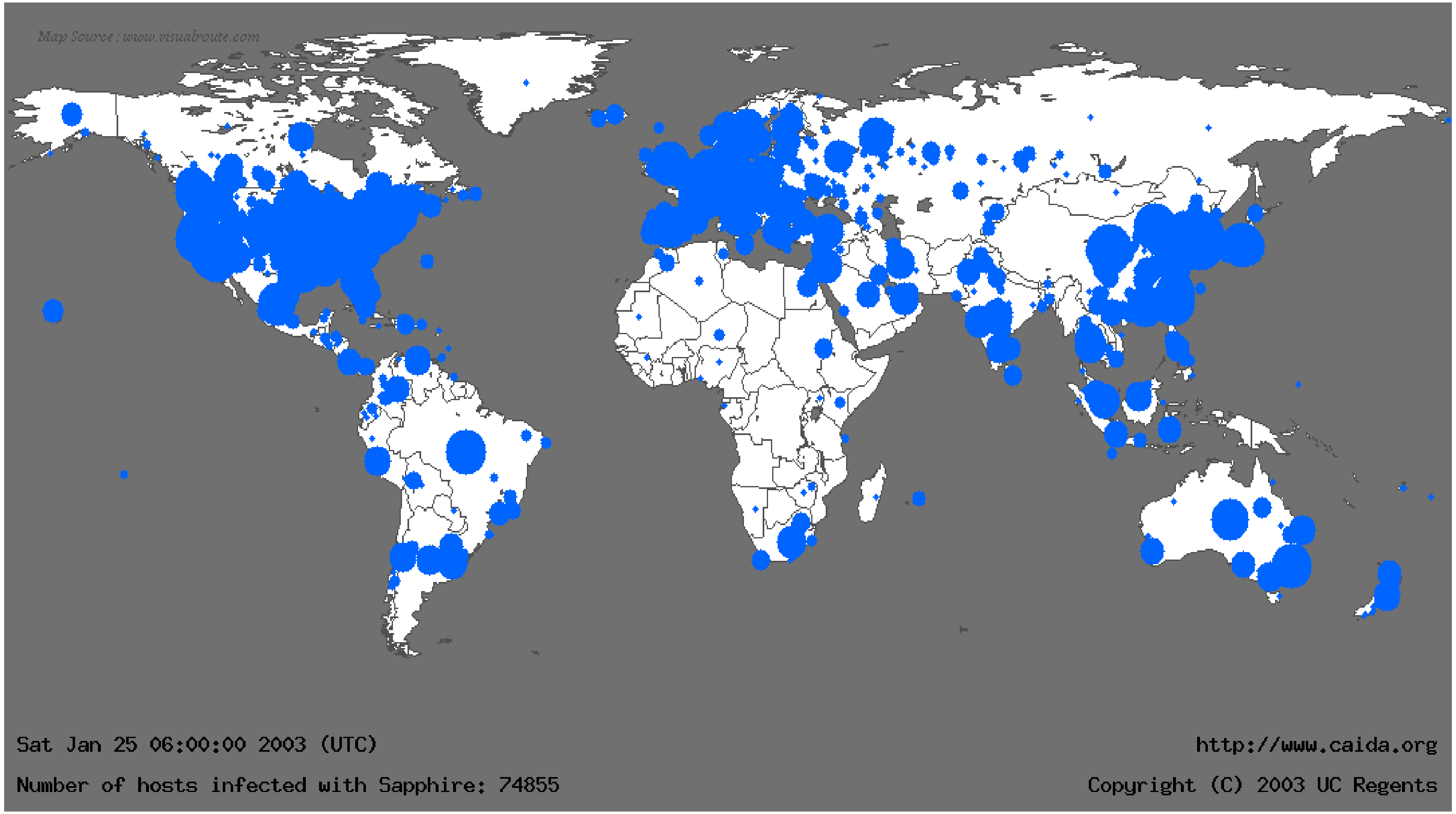
The world: Pre-Slammer Worm


Map Source : www.visualroute.com
Sat Jan 25 05:29:00 2003 (UTC)
http://www.caida.org
Number of hosts infected with Sapphire: 0
Copyright (C) 2003 UC Regents

From www.caida.org: Lots of interesting things.

Exploits buffer overflow in Microsoft SQL Server 2000.

31 minutes later ☹


Map Source : www.visualroute.com
Sat Jan 25 06:00:00 2003 (UTC)
Number of hosts infected with Sapphire: 74855
http://www.caida.org
Copyright (C) 2003 UC Regents

Why was Slammer so fast?

- 90% of the vulnerable hosts were infected within 10 minutes.
- The spread doubled every 8.5 seconds and the highest scanning rate of 55 million scans per second was reached in 3 minutes.
- Slammer used UDP, which is connectionless, unlike TCP which involves responses.
- As such it wasn’t limited by the latency in communicating back to the source host, it was limited just by the bandwidth from an attack station and the speed of transport to other hosts.

The broadcast mechanism

- The Slammer code randomly generated an IP address and sent out a copy of itself to that location, using UDP.
  - Slammer exploited a vulnerability in MS SQL Server Resolution Service, for which a patch had been available for 6 months.
  - If a UDP packet arrived on port 1434 with the first byte 0x04, the rest of the packet is interpreted as a registry key to be opened, and is stored in the buffer for that purpose later.
  - But without length checking the rest of the packet can be written into the buffer.
- The packet was just 376 bytes long.
- https://www.giac.org/paper/gcih/414/attack-slammer-worm-practical-case-study/103632

Trojan horses

- A friend passes you an interesting game.
  - You run it and it is lots of fun.
  - The license says it’s freeware so you pass it to your friends.
- What could happen?
  - It could be a Trojan Horse.
  - It could be doing something malicious in addition to the obvious game.

Trojan Technology

- Some concealment methods:
  - A Trojan renames itself to the name of a valid system file.
  - They can also be encrypted and polymorphic, and could install themselves in different ways to escape detection.

- Hiding as source code:
  - They transfer themselves as ASCII text source code onto the host machine.
  - They are then compiled or interpreted to bypass malicious code scanners.
  - The executable code or a batch file is included to assemble or interpret the code on the fly.
  - The companion programs that assemble or link the source code into its runtime form are usually legitimate programs and will not be flagged by scanners either.
  - Other Trojans use tools already available on most Windows PCs (*DEBUG.EXE or WSCRIPT.EXE*) to launch their programs.

- Remote Administration Trojans (RAT's):
  - These allow a hacker to take complete control of a PC. Very nasty with keyboard & screen capture, and the ability to directly manipulate your computer.
- Backdoor Programs:
  - In March 2001 it was revealed that a group of Eastern European hackers had spent over a year exploiting an NT vulnerability and installing backdoor Trojans to steal more than a million credit cards from over 40 top e-commerce and e-banking web sites.
- Network Redirection:
  - Many Trojans, including the most successful RATs, allow network redirection. This allows a hacker to redirect specific attacks through a compromised intermediate host machine toward a new target.
    - This can, for example, be used to avoid firewall filtering.

Bacteria

- A bacterium or rabbit (or wabbit ☺) is a program that creates many instances of itself to burn up resources of some type.
  - A bacterium is not required to use all resources on the system, but is aiming to result in some level of denial of service.
- An example …

```
while true
do
    mkdir x
    chdir x
done
```

Logic Bombs

- A program that performs an malicious action when some external event occurs.
- Plant Trojan horses in system using a logic bomb.
- The most common logic used is date matching, but it could be anything.

Backdoor

- Backdoor (Trapdoor) is a secret entry point into a program that allows someone to gain access without going through the usual security access procedures
  - Programmers used backdoor to debug and test programs
  - But can be used to gain unauthorised access
- A backdoor is usually implemented as a network service listening on some non-standard port that the attacker can connect to and issue commands through to be run on the compromised system
  - The WannaCry ransomware included such a backdoor
  - See: https://www.kaspersky.com.au/resource-center/threats/ransomware-wannacry
- It is difficult to implement operating system controls for backdoors in applications.

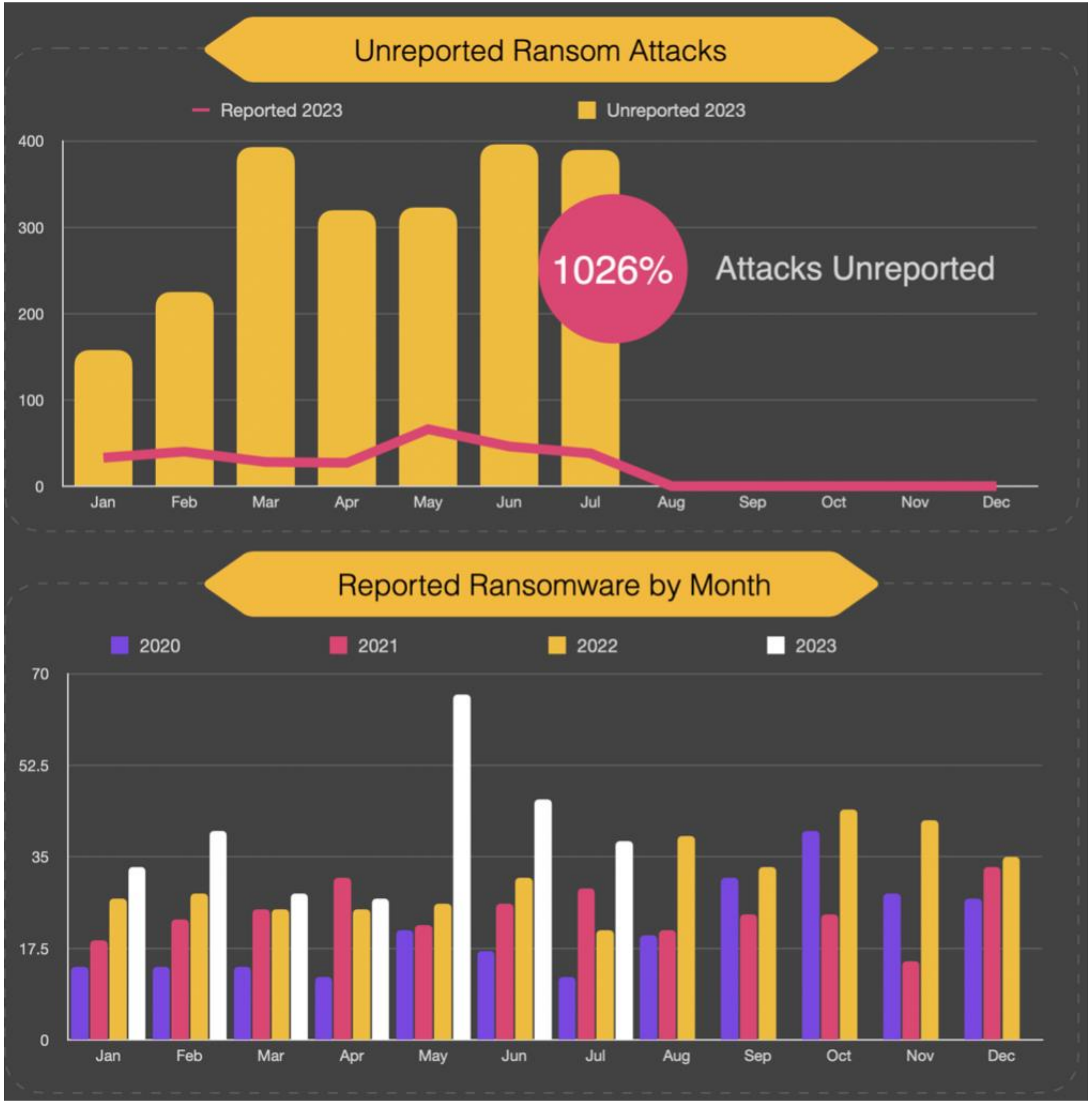
Rootkits

- These are collections of tools designed to “govern” a particular operating system.
  - Why rootkit? Because root is the system administration account on Unix systems.
- More specifically, a installed rootkit allows an attacker to change the system functionality. This includes:
- How do rootkits get there?
  - Trojan horses are one way.
  - Adding and removing programs.
  - Transmit and receive network traffic.
  - Changing the state of system monitors.
- How do rootkits get there?
  - Trojan horses are one way.

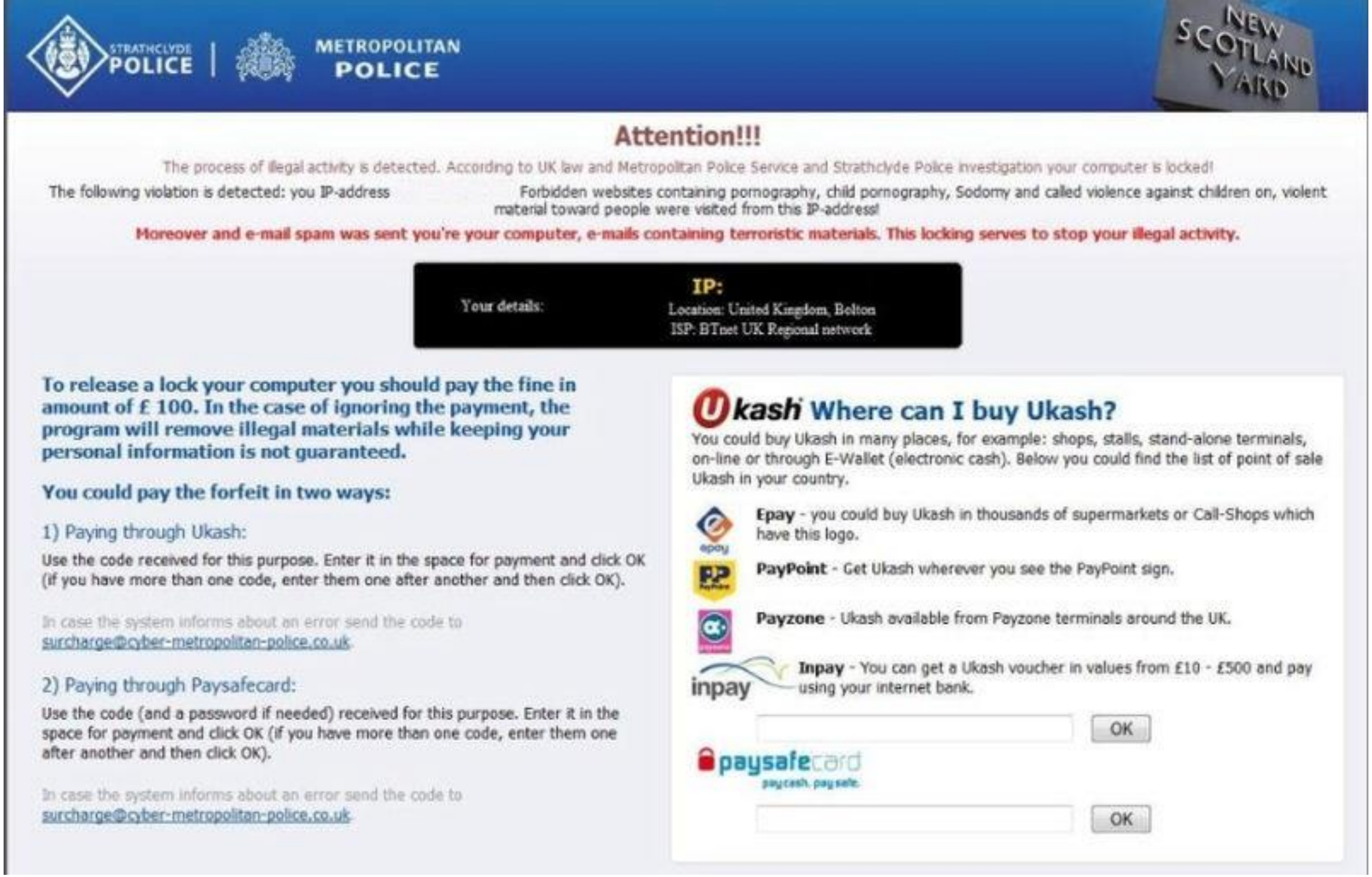
- In the scenario on the previous slide the rootkit tries to hide itself.
- This is typical.
    - Disabling scanners.
    - Hiding malicious processes, …
        - Either at the user/application level so the user doesn’t receive accurate information.
        - Or at the kernel level by removing the malicious processes from the kernel’s active list.
- Rootkits are very useful for increasing the effectiveness of other attacks.

Ransomware

- From the Symantec report (April 2015)
- “Ransomware attacks grew 113 percent in 2014, along with 45 times more crypto-ransomware attacks.” ~ 24,000 per day…
- Randomware typically involves an attack on availability that will not be stopped until some payment is made.
    - Crypto-ransomware involves encrypting files.
    - When you pay, you get the password/key.



From: https://www.blackfog.com/the-state-of-ransomware-in-2023/



Attention!!!

The process of illegal activity is detected. According to UK law and Metropolitan Police Service and Strathclyde Police investigation your computer is locked!

The following violation is detected: you IP-address Forbidden websites containing pornography, child pornography, Sodomy and called violence against children on, violent material toward people were visited from this IP-address!

Moreover and e-mail spam was sent you're your computer, e-mails containing terroristic materials. This locking serves to stop your illegal activity.

Your details: **IP:**
Location: United Kingdom, Bolton
ISP: BTnet UK Regional network

To release a lock your computer you should pay the fine in amount of £ 100. In the case of ignoring the payment, the program will remove illegal materials while keeping your personal information is not guaranteed.

You could pay the forfeit in two ways:

1) Paying through Ukash:

Use the code received for this purpose. Enter it in the space for payment and click OK (if you have more than one code, enter them one after another and then click OK).

In case the system informs about an error send the code to surcharge@cyber-metropolitan-police.co.uk

2) Paying through Paysafecard:

Use the code (and a password if needed) received for this purpose. Enter it in the space for payment and click OK (if you have more than one code, enter them one after another and then click OK).

In case the system informs about an error send the code to surcharge@cyber-metropolitan-police.co.uk

Ukash Where can I buy Ukash?

You could buy Ukash in many places, for example: shops, stalls, stand-alone terminals, on-line or through E-Wallet (electronic cash). Below you could find the list of point of sale Ukash in your country.

Epay - you could buy Ukash in thousands of supermarkets or Call-Shops which have this logo.

PayPoint - Get Ukash wherever you see the PayPoint sign.

Payzone - Ukash available from Payzone terminals around the UK.

Inpay - You can get a Ukash voucher in values from £10 - £500 and pay using your internet bank.

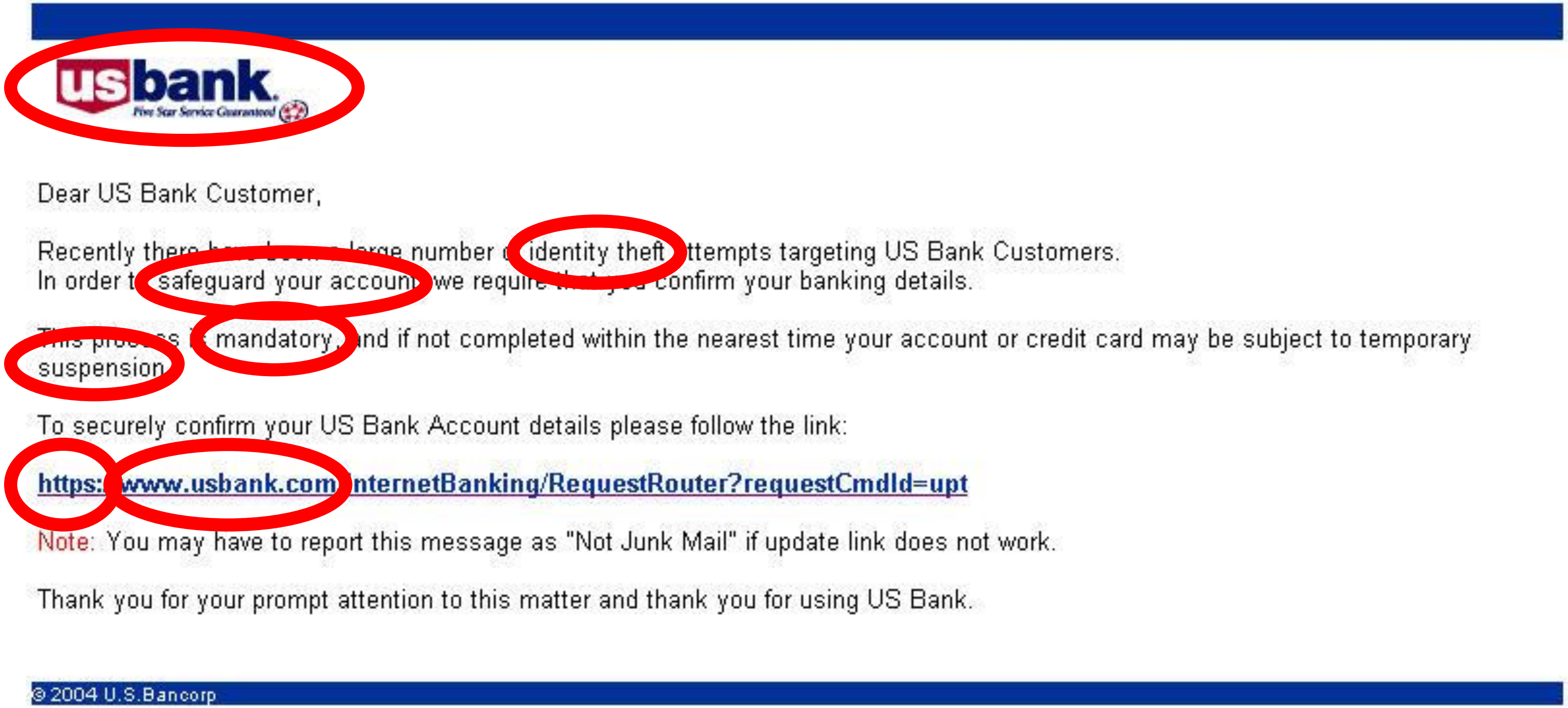
- This one uses some social engineering too …
- From https://en.wikipedia.org/wiki/Ransomware

Phishing

- **Phishing**: Cybercriminal attempts to steal personal and financial information or infect computers and other devices with malware and viruses
  - It’s a form of social engineering.
  - Designed to trick you into clicking a link or providing personal or financial information
  - Often in the form of emails and websites
  - May appear to come from legitimate companies, organizations or known individuals
  - Take advantage of natural disasters, epidemics, health scares, political elections or timely events

- According to the Anti-Phishing Working Group (www.antiphishing.org),
  > "Phishing attacks use both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials."
- Things like keyloggers, screenloggers and web Trojans can be considered phishing tools.
- The Anti-Phishing Working Group(www.antiphishing.org) produces quarterly reports on trends in, primarily, Phishing.
  - In the fourth quarter of 2022: 1,350,037 phishing attacks were observed
  - http://www.antiphishing.org/trendsreports/
- The State of Phishing Attacks 2022: https://www.slashnext.com/the-state-of-phishing-2022/

A typical Phishing Email…



Dear US Bank Customer,

Recently there have been a large number of identity theft attempts targeting US Bank Customers.
In order to safeguard your account we require that you confirm your banking details.

This process is mandatory, and if not completed within the nearest time your account or credit card may be subject to temporary suspension.

To securely confirm your US Bank Account details please follow the link:

https://www.usbank.com/internetBanking/RequestRouter?requestCmdId=upt

Note: You may have to report this message as "Not Junk Mail" if update link does not work.

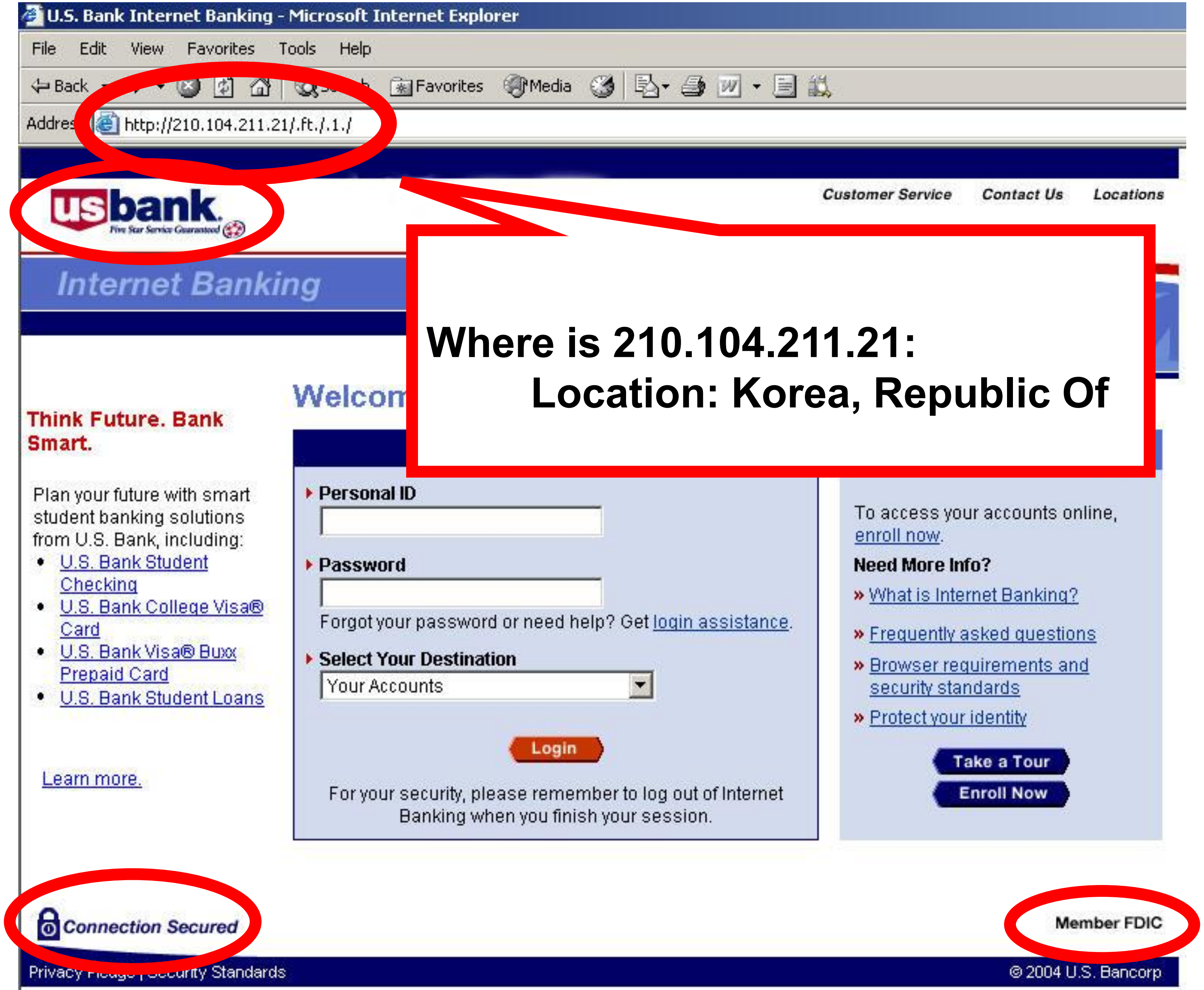
Thank you for your prompt attention to this matter and thank you for using US Bank.

© 2004 U.S.Bancorp

Images from Anti-Phishing Working Group's Phishing Archive

… and then…

- … the next page requests more information:
  - Name.
  - Address.
  - Telephone.
  - Credit Card Number, Expiration Date, Security Code.
  - PIN.
  - Account Number.
  - Personal ID.
  - Password.



http://210.104.211.21/.ft./.1./

Welcom

Think Future. Bank Smart.

Plan your future with smart student banking solutions from U.S. Bank, including:

- U.S. Bank Student Checking
- U.S. Bank College Visa® Card
- U.S. Bank Visa® Buxx Prepaid Card
- U.S. Bank Student Loans

Learn more.

Personal ID

Password

Forgot your password or need help? Get login assistance.

Select Your Destination

Your Accounts

For your security, please remember to log out of Internet Banking when you finish your session.

To access your accounts online, enroll now.

Need More Info?

» What is Internet Banking?

» Frequently asked questions

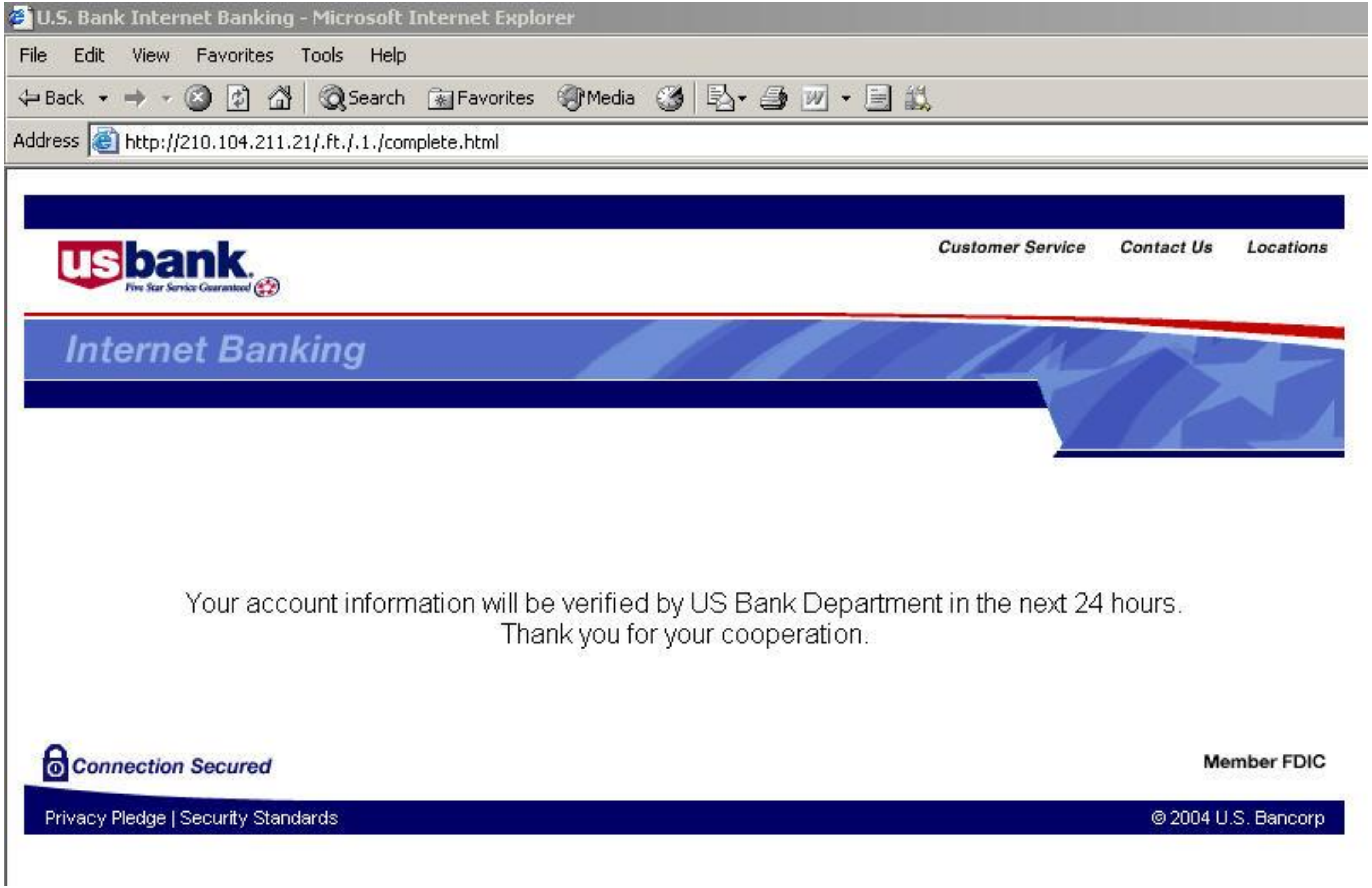
» Browser requirements and security standards

» Protect your identity

Where is 210.104.211.21:

Location: Korea, Republic Of

Images from Anti-Phishing Working Group's Phishing Archive


U.S. Bank Internet Banking - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
usbank.
Five Star Service Guaranteed
Customer Service
Contact Us
Locations
Internet Banking
Connection Secured
Member FDIC
Privacy Pledge | Security Standards
© 2004 U.S. Bancorp

http://210.104.211.21/.ft./.1./complete.html

Your account information will be verified by US Bank Department in the next 24 hours.
Thank you for your cooperation.

Countermeasures

- Malware acts as both data and instructions:
  - A virus inserts code (a set of instructions) into another program.
  - The set of instructions is treated as data.
  - The virus executes itself, the set of instructions is treated as an executable.
  - Protection:
    - Treat all programs as type "data“.
    - Some certifying authority can change the type to executable, after verification takes place.

- Against Malicious code assuming the identity of a user:
  - When a user executes malicious code, that code can access and affect objects within the user's protection domain.
  - Protection: Limiting the objects accessible to a given process run by the user.
    - Information Flow Metrics.
    - Reducing the Rights.
    - Sandboxing.

- **Information flow metrics**:
- Define the flow distance metric *fd*(*x*) for some information x as follows:
    - Initially, all information has *fd*(*x*) = 0.
    - Whenever *x* is shared, *fd*(*x*) increases by 1.
    - Whenever *x* is used as input to a computation, the flow distance of the output is the maximum of the flow distances of the input.
- Information is accessible only while the distance is **less** than some value V.
- Information vs data?

- Example of applying the flow distance metric.
  - A, B, and C work on the same computer.
  - $V_A = 3$. $V_B = V_C = 2$.
  - A creates a program P containing a virus. ☺
  - B executes P.
    - The contents of P have a flow distance of 0, so when the virus infects B's file Q, the flow distance of the virus is 1, and so B can access it.
      - Hence, the copying succeeds.
  - C executes Q, when the virus tries to spread to her files, its flow distance increases to 2.
    - Hence, the infection is not permitted, because C can only access information with a flow distance 0 or 1.
    - C can however execute P and it will flow. ☹

- **Reducing the Rights**:
  - The user can reduce their associated work domain when running a suspect program.
  - This follows from the principle of least privilege:

> A subject should be given only those privileges that it needs in order to complete its task.

More on defence

- **Sandboxing** can be used.
    - Virtual environments.
        - From Symantec report (April 2015).
            - "In 2014, up to 28 percent of all malware was "virtual machine aware.""
- Restrict sharing by controlling the domain boundaries:
    - Restrict users in different protecting domains from sharing programs or data.
    - Programs to be protected should be placed at the lowest level of an implementation of a multilevel security policy.

Detection

- Normal behaviour of a system is usually different from the activity profile of an infected system.
    - Virus monitors monitor known methods of virus activity, such as attempts to write to a boot sector, modify interrupt vectors, write to system files... and detect abnormal behaviour of the system.
- **Advantages:**
    - Works for all viruses.
    - Detection is before (complete) infection.
- **Disadvantages:**
    - To detect a high percentage of viruses, the sensitivity of the monitor must be set high and this may generate many false alarms.

- Theorem (by Cohen):

  It is undecidable whether an arbitrary program contains a computer virus.

There are formal definitions of viruses that allow this type of result to be derived.

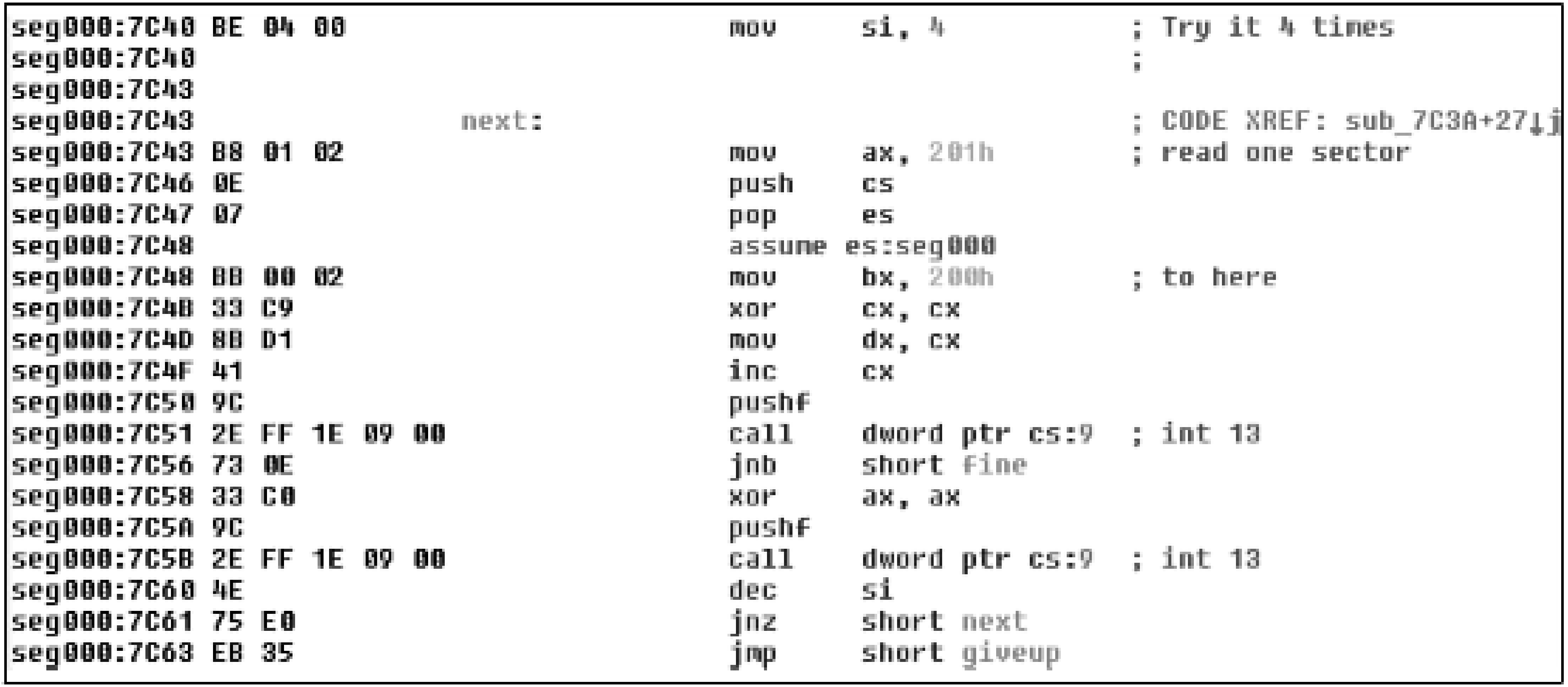
Multiple copy testing …

- Run several copies of the “same program or algorithm”.
    - This is not simply running the same program several times.
- Majority rules…
- The check could be based on results, calls made, efficiency …
- Majority still might not be trusted, because they might all be corrupted.
- But different performances can imply action needs to be taken.

Signature scanning

- Signature scanning (signature= search string= scan string) is the simplest and the most common approach to virus detection.
- Signature extraction is a non-trivial process:
  - The infection is disassembled and the key portions are identified.
  - The key portions are combined to form a signature.
  - The signature is checked against a large library of programs to reduce the chance of false positives occurring when signature accidentally matches some library code.

11.1 First-Generation Scanners



```
seg000:7C40 BE 04 00                          mov     si, 4           ; Try it 4 times
seg000:7C40                                                           ;
seg000:7C43
seg000:7C43                next:                                      ; CODE XREF: sub_7C3A+27↓j
seg000:7C43 B8 01 02                          mov     ax, 201h        ; read one sector
seg000:7C46 0E                                push    cs
seg000:7C47 07                                pop     es
seg000:7C48                                   assume es:seg000
seg000:7C48 BB 00 02                          mov     bx, 200h        ; to here
seg000:7C4B 33 C9                             xor     cx, cx
seg000:7C4D 8B D1                             mov     dx, cx
seg000:7C4F 41                                inc     cx
seg000:7C50 9C                                pushf
seg000:7C51 2E FF 1E 09 00                    call    dword ptr cs:9  ; int 13
seg000:7C56 73 0E                             jnb     short fine
seg000:7C58 33 C0                             xor     ax, ax
seg000:7C5A 9C                                pushf
seg000:7C5B 2E FF 1E 09 00                    call    dword ptr cs:9  ; int 13
seg000:7C60 4E                                dec     si
seg000:7C61 75 E0                             jnz     short next
seg000:7C63 EB 35                             jmp     short giveup
```

Figure 11.2 A code snippet of the Stoned virus loaded to IDA.

- This is from page 428 of The Art of Computer Virus Research and Defense by Peter Szor (2005).

- **Advantages**:
  - Signature scanning can be used against Trojan horses, logic bombs and other malicious software.
- **Disadvantage:**
  - Scanning cannot find new viruses before their patterns are known.
  - It is also ineffective against polymorphic viruses.
- First-generation scanner's use virus signatures only.
- GPU's can be used to speed up signature processing! https://developer.nvidia.com/gpugems/gpugems3/part-v-physics-simulation/chapter-35-fast-virus-signature-matching-gpu

2nd, 3rd and 4th generation scanners

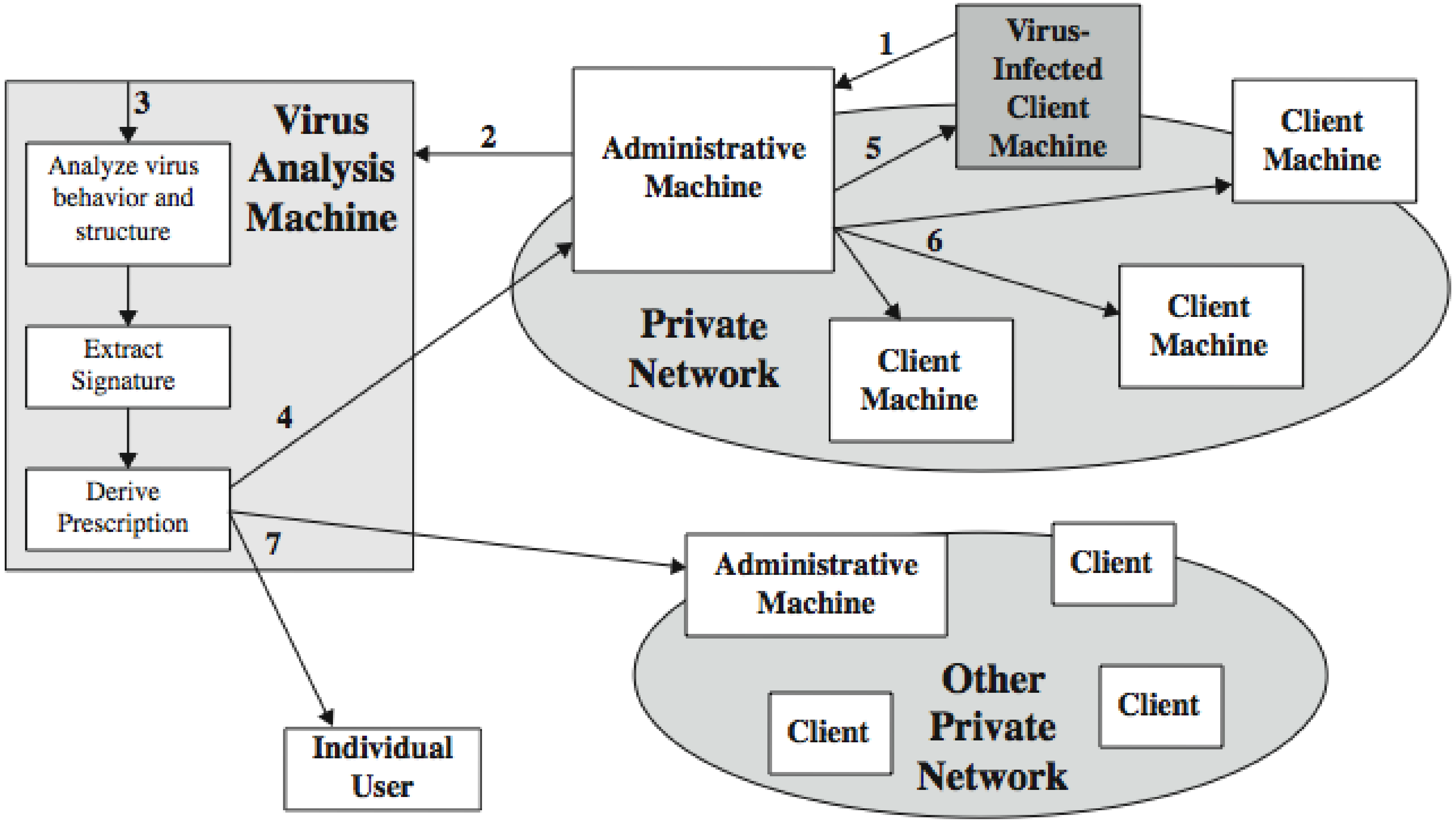
- **Second generation** scanners don’t just use specific signatures, since the signatures of many viruses change (polymorphic).
  - User heuristic rules to search for probable malware instances
  - Integrity checks (e.g. checksum) can also be applied

- Manipulation Detection Codes (or MDC's).
    - Apply some function to a file to obtain a set of bits called the signature block and then protect that block.
    - If the recomputing the signature block, the result differs from the stored signature block, the file has changed, possibly as a result of a malicious code.
    - We can use hash functions here!

- … and timestamps are useful too.
- The time of the last change to a program is kept separate from the environment that the program is stored.
- Timestamps should be frequently checked to ascertain the integrity of the program.

- Third generation scanners detect viruses by behaviour.
  - For example, attempts to interact inappropriately with certain system files could trigger detection.
- Fourth generation basically use a collect of antivirus techniques together.
- One quite important, and fairly new tool, is virtualisation.
  - We can run a virus in an isolated but realistic environment and see what happens.

Digital Immune System



Originally IBM, subsequently Symantec.

Figure 6.6, page 213 of Stallings and Brown. Edition 2

- Objective: To provide a rapid response so viruses can be stamped out soon after being introduced.
    - On detecting a new virus, the immune system captures and analyses it, adds detection and shielding information, removes it, and passes information about that virus to other systems so it can be detected before being allowed to run elsewhere.

1. Monitoring programs on the PC’s use heuristics to infer a virus may be present. A copy is passed to an administrative machine.

2. Admin machine encrypts the “virus” and sends it to a central virus analysis machine.

3. The central virus analysis machine provides a safe environment for analysis, and produces a prescription for virus identification and removal.

4. Prescription sent back to the admin machine.

5. Prescription forwarded to the infected client.

6. Prescription forwarded to other clients in the organization.

7. Worldwide subscribers get regular antivirus updates.