Introduction to Network Security

Chapter 9

Email

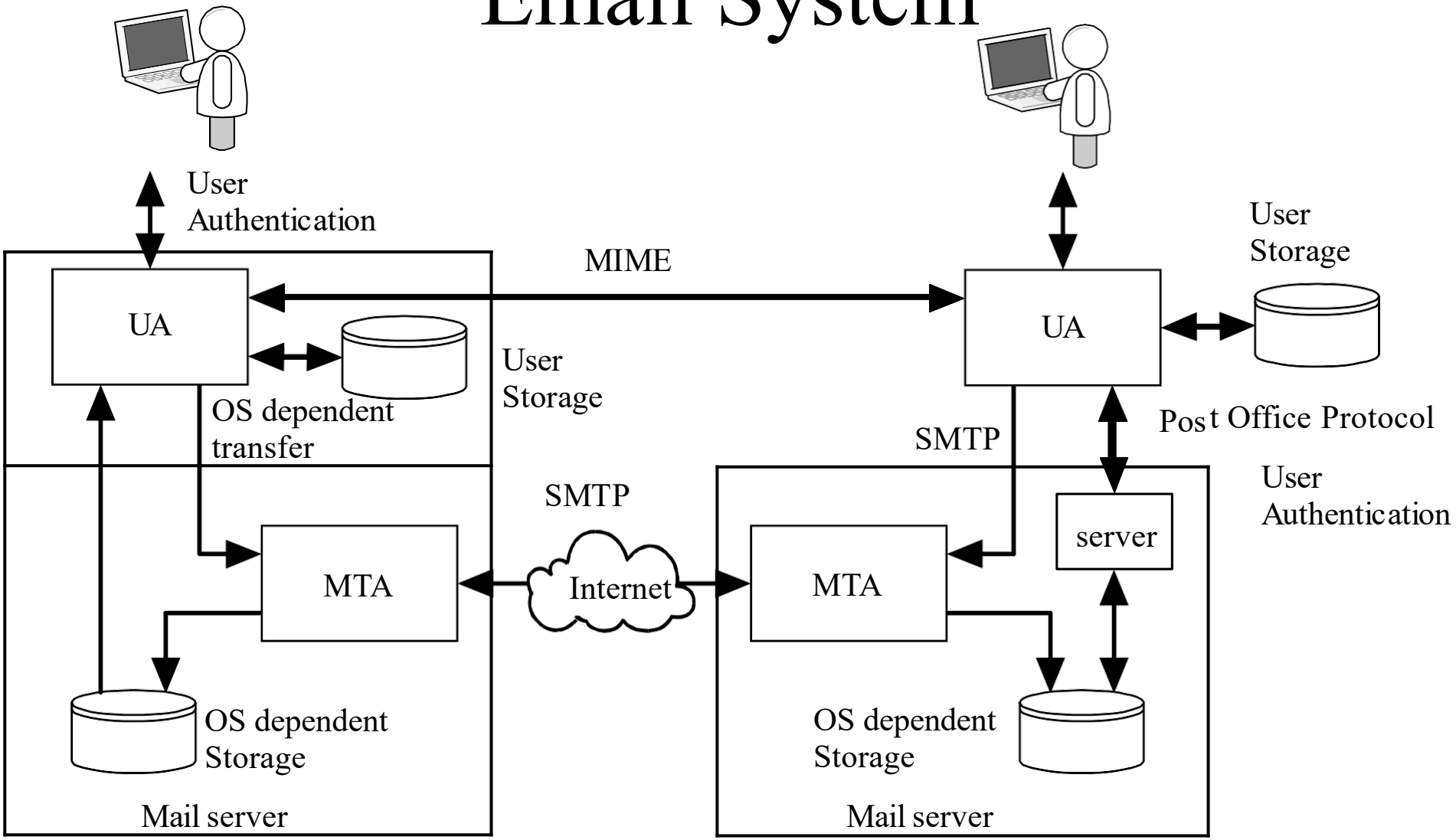
Email Topics

- SMTP
- POP & IMAP
- MIME
- Vulnerabilities
- General Email Countermeasures

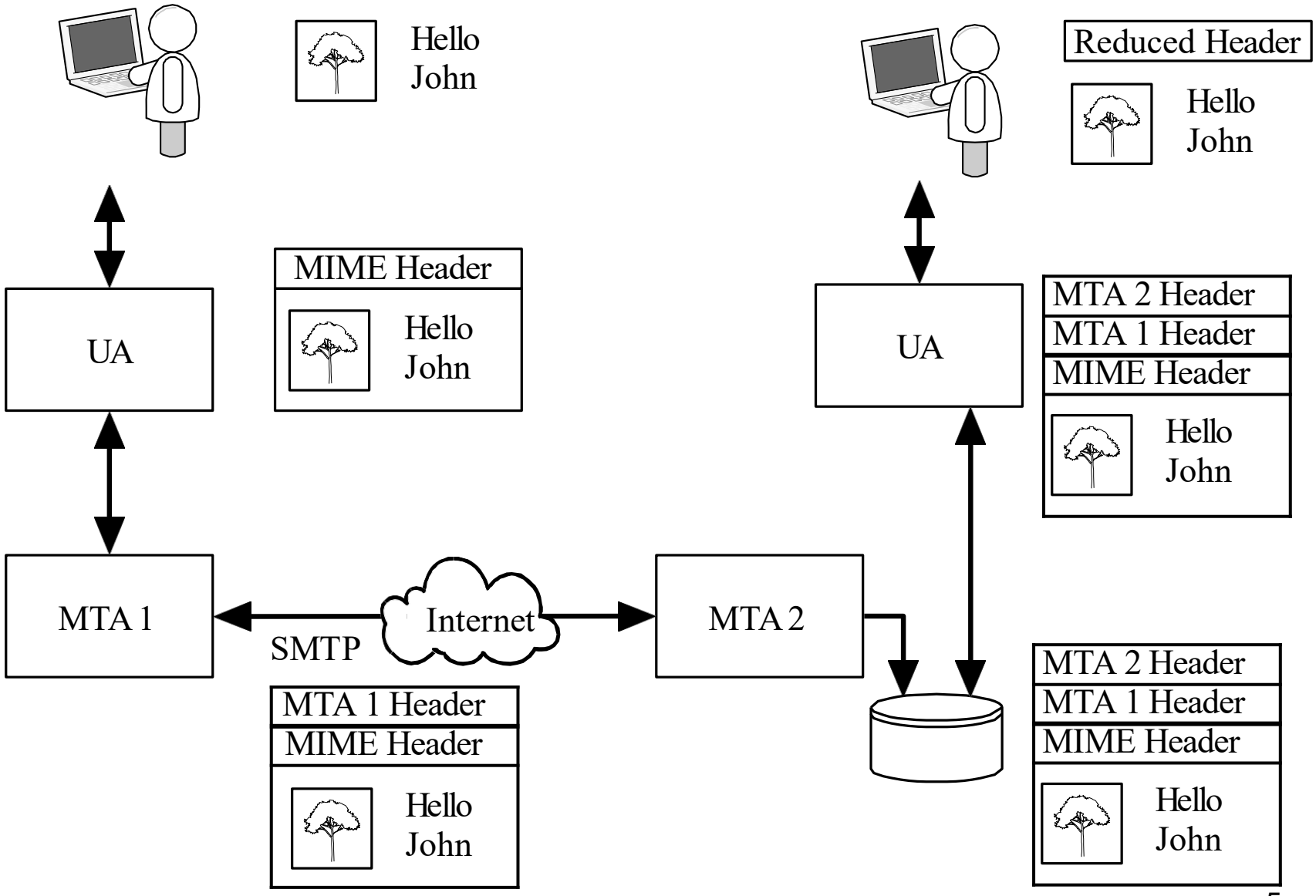
Email

- A basic electronic mail system performs four functions:
- Creation: A user creates and edits a message, generally using a rudimentary editing capability. Most systems also allow the user to create a message using the system editor or a word processor, and then incorporate the resulting file as the body of the message.
- Sending: The user designates the recipient (or recipients) of the message, and the facility stores the message in the appropriate mailbox(es)
- Reception: The intended recipient may invoke the electronic mail facility to access and read the delivered mail
- Storage: Both sender and recipient may chose to save the message in a file for permanent storage

Email System



Email Message Format



Email

The SMTP protocol is the standard protocol for transferring mail between hosts. The protocol was defined in RFC 821 and later formalized as MIL-STD-1781.

SMTP is not concerned with the format or content of the messages themselves, with two minor exceptions.

SMTP adds logging information to message that indicates the path the message took.

Email

The SMTP protocol attempts to provide reliable operation, but does not guarantee to recover from hosts that lose files. No end-to-end acknowledgment is returned to a message's originator when a message is successfully delivered, and errors are not guaranteed to be returned either. However, the mail system is sufficiently reliable that this is not an issue.

In most cases mail goes directly from the mail originator's machine to the destination machine. However, mail will occasionally go through intermediate systems.

The SMTP protocol is made up of a set of simple commands.

Email

SMTP has 14 commands.

Command syntax is a set of 4 letter commands with parameters
Not all commands need to be implemented

The commands are:

CMD Syntax	Action
HELO <domain>	Used by the sending system to identify itself (HELO eeclass.ee.iastate.edu)

Email

CMD Syntax	Action
MAIL FROM: <path>	Identifies who the message is from. (MAIL FROM doug@iastate,edu) error messages have a NULL from field to prevent answers.
RCPT TO: <forward path>	Identifies who the message should be mailed to. There is separate RCPT for each recipient.

Email

CMD Syntax	Action
DATA	Indicates that the next transmission contains the message text. Terminated with a line containing <CR LF>.<CR LF>
RSET	Terminate current transaction
SEND FROM: <path>	Used instead of MAIL if message should be displayed on user's terminal.

Email

The reply codes are designed to make implementation of SMTP easier. Each digit of the three digit code has a unique purpose.

First digit specifies whether the response was good, bad, or or incomplete.

The second digit specifies what type of error occurred.

The third digit details specific failures.

The values for the codes are given on the next slide.

Email

1XX Positive Preliminary Reply - The command has been accepted, but the receiver requires more information. (not used by SMTP, used by other protocols)

2XX Positive Completion Reply - The requested action has been successfully completed. A new request may be initiated.

3XX Positive Intermediate Reply - The command has been accepted, but action is being held, pending receipt of further information. The SMTP sender should send another command specifying this information.

Email

4XX Transient Negative Completion Reply - The command was not accepted, however, the error condition is temporary

5XX Permanent Negative Completion Reply - The command was not accepted.

Email

X0X Syntax Error or unimplemented commands

X1X Information: reply to requests for information

X2X Connections - reply to the request for connection

X3X Unspecified

X4X Unspecified

X5X Mail System - indicates the status of the receiver during, for example, a transfer.

Email

211 System status or system help reply

214 help message

220 service ready

221 Service closing transmission channel

250 Requested mail action okay, completed

251 User not local; will forward to <forward path>

354 Start mail input

421 Service not available; closing channel

450 Mail box busy

451 requested action terminated; local error in processing

452 Requested action not taken; insufficient system storage

Email

500 Syntax Error, command unrecognized

501 Syntax Error in parameters or arguments

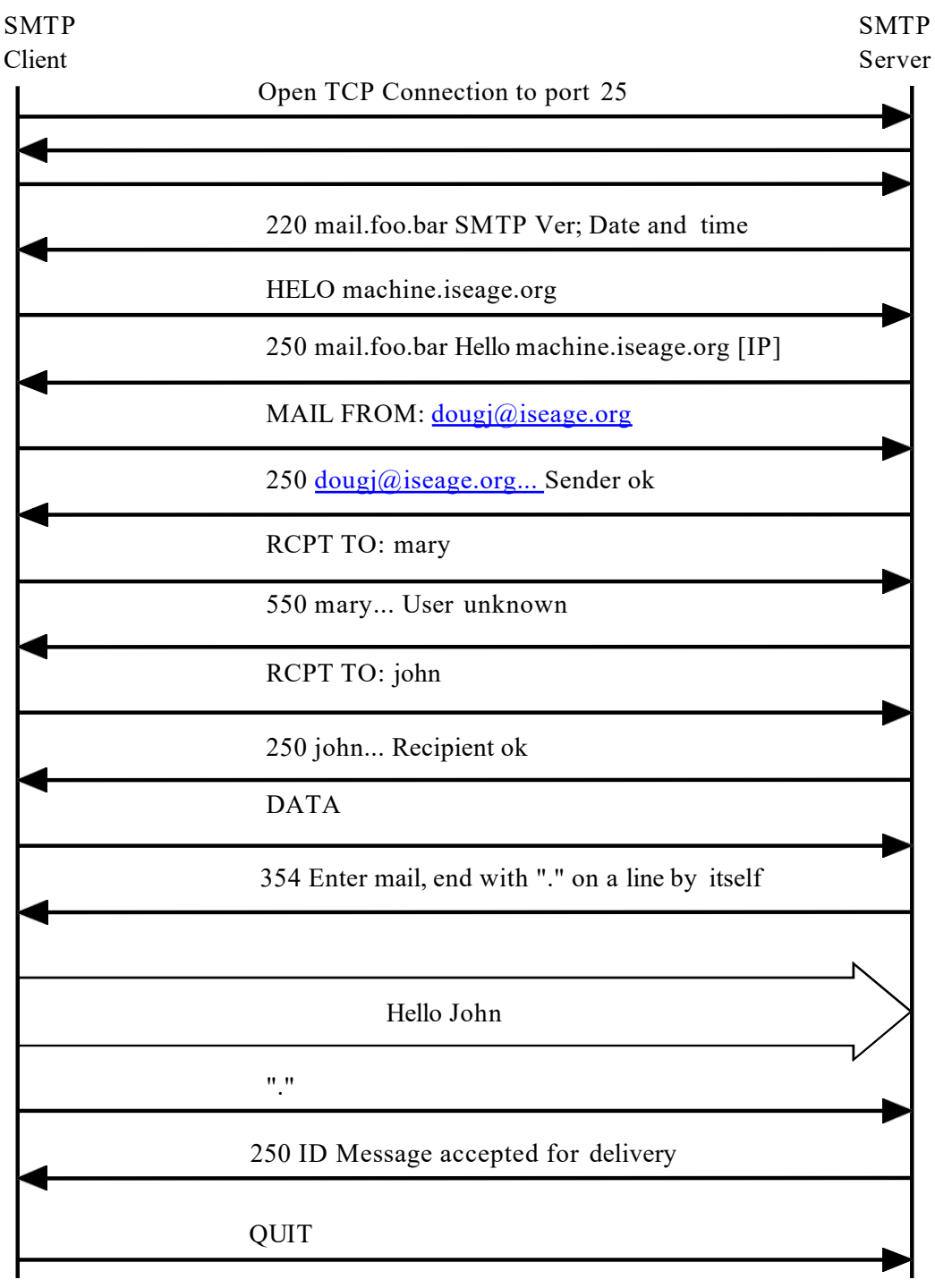
502 Command not implemented

550 mailbox not found

551 user not local; please try <forward path>

554 transaction failed

SMTP


SMTP
Client
SMTP
Server
Open TCP Connection to port 25
220 mail.foo.bar SMTP Ver; Date and time
HELO machine.iseage.org
250 mail.foo.bar Hello machine.iseage.org [IP]
MAIL FROM: dougj@iseage.org
250 dougj@iseage.org... Sender ok
RCPT TO: mary
550 mary... User unknown
RCPT TO: john
250 john... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Hello John
"."
250 ID Message accepted for delivery
QUIT

POP

Post Office Protocol

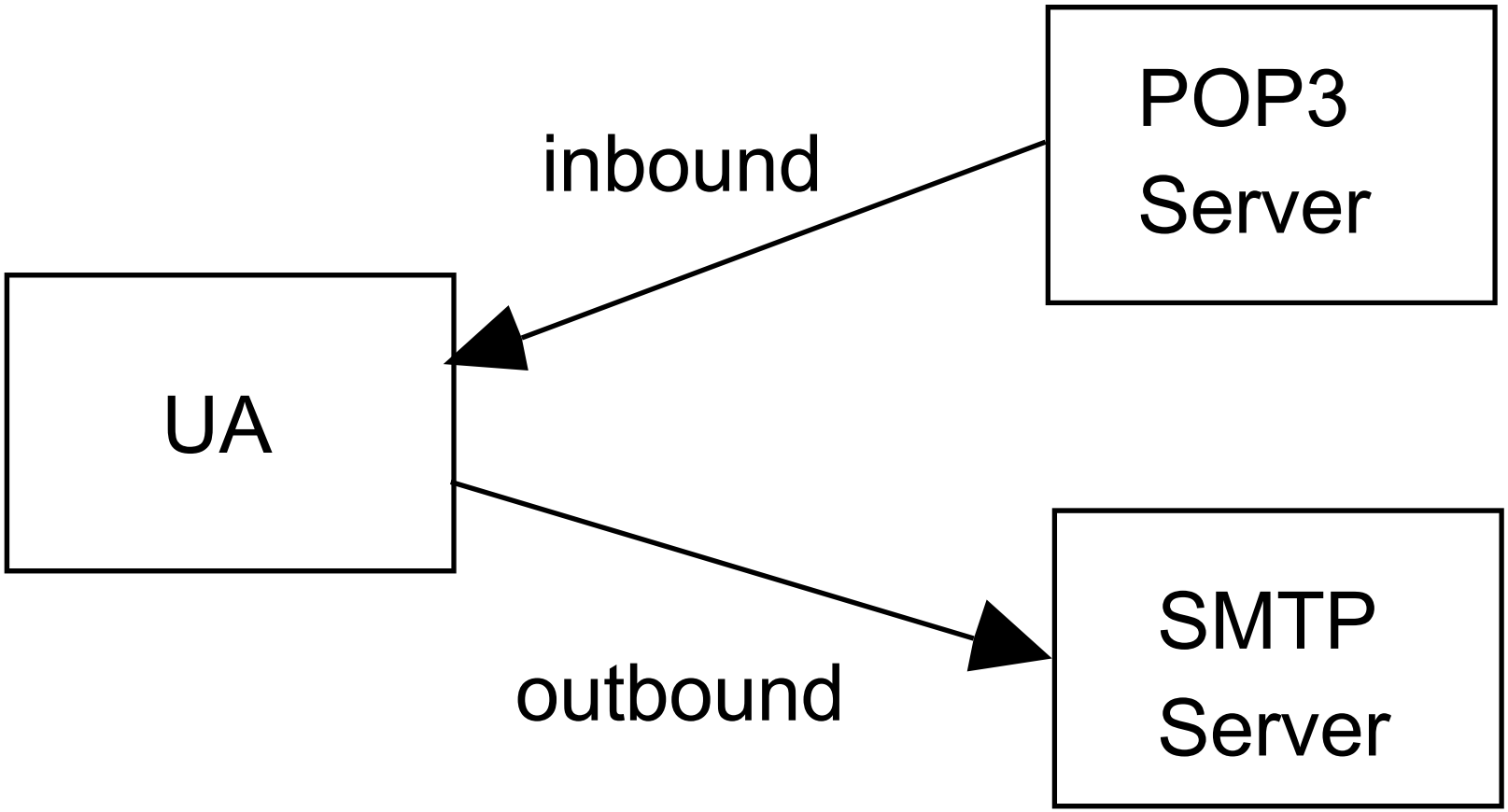
Used to transfer mail between the mail server and a PC

Provides user Authentication

POP3 protocol

- POP3 client “logs in” to a POP3 server (TCP port 110)
- Login name and password in clear text
- User can configure how often mail is checked
  - this means the login and password can be sent many times a day
  - easy to capture since when there is no mail there are only a few packets exchanged.

POP3 block diagram



POP3 Commands

- USER name — Login name
- PASS string — User password
- STAT — returns number of messages
- LIST [msg] — returns the size of msg or all messages if [msg] is not supplied
- RETR msg — send client the full message [msg]
- DELE msg — Delete message from server
- NOOP — No operation
- RSET — Reset deletion indicators

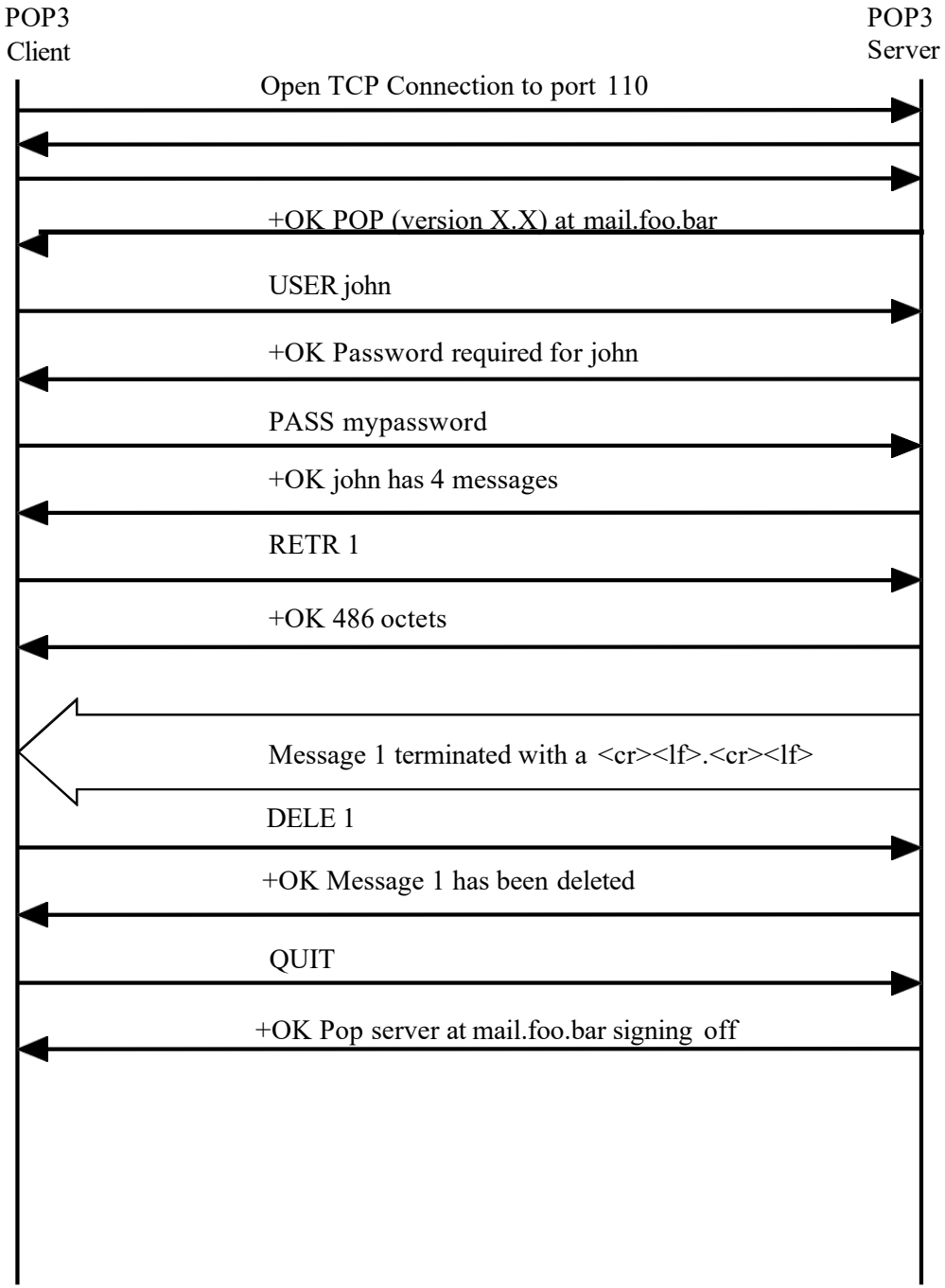
POP3 Commands

- Quit — Quit the session
- APOP name digest — Optional authentication
- TOP msg n — return first n lines of message
- UIDL — returns a unique ID string for the requested message, does not change during session. Message ID can used to request message.

POP3 Responses

- Two response codes
  - -ERR message
  - +OK message

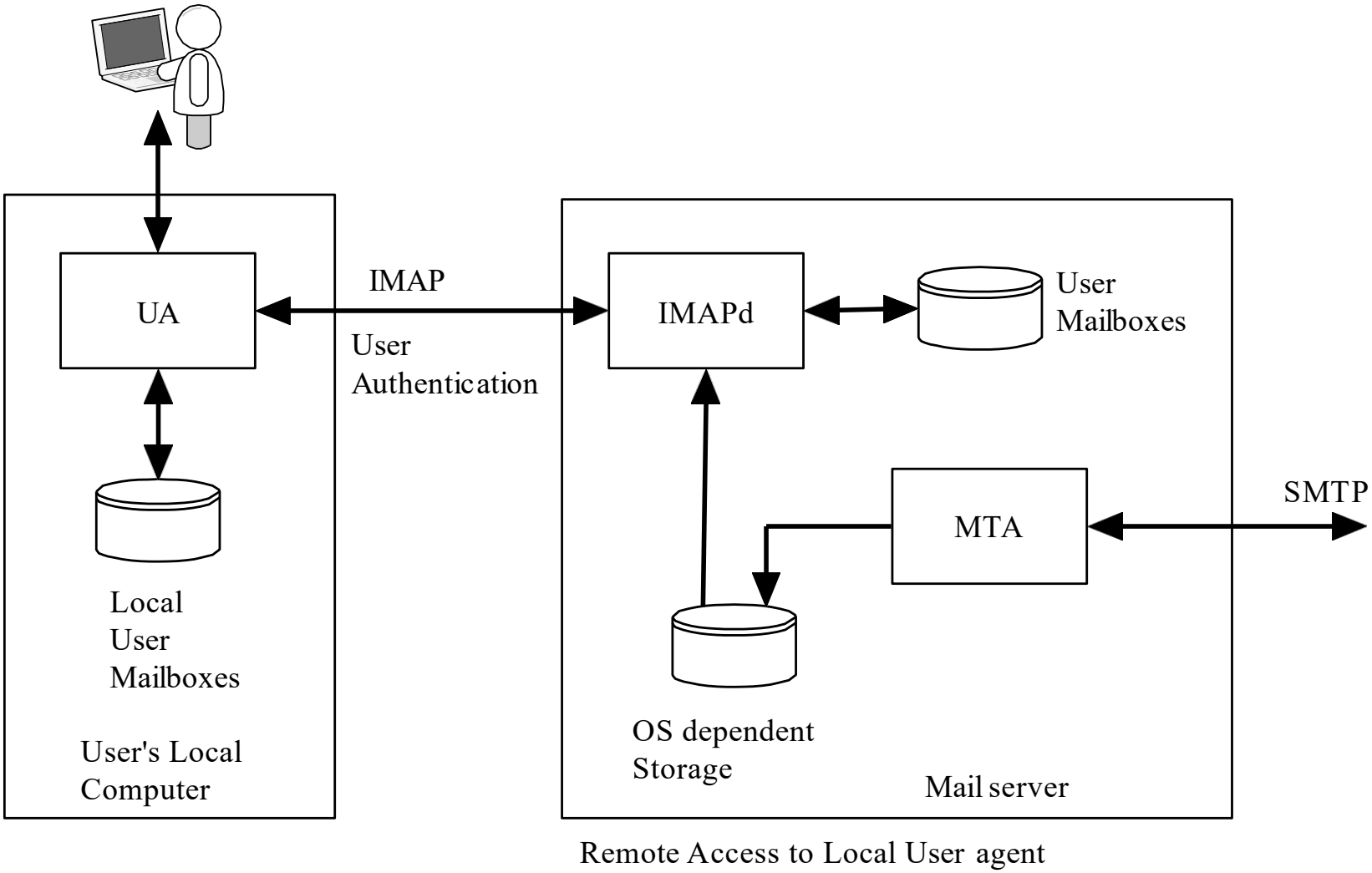
POP3 Protocol



IMAP

- Supports message retrieval
- POP, does not work well in a multiple client configuration since mail is deleted after it is read.
- IMAP can keep messages on the server and can be used by multiple clients.

IMAP Mail Boxes



Remote Access to Local User agent

IMAP Commands

- CAPABILITY List server capabilities
- NOOP No operation
- LOGOUT
- AUTHENTICATE type
- LOGIN name passwd
- SELECT mailbox
- EXAMINE mailbox read only version of select
- CREATE mailbox
- DETELE mailbox

IMAP COMMANDS

- RENAME current-name new-name rename mailbox
- SUBCRIBE mailbox add mailbox to servers list of active mailboxes
- UNSUBSCRIBE mailbox
- LIST ref mailbox provide a list of mailboxes
- LSUB provide a list based on subscribe
- APPEND mailbox mess Append the message to the mailbox
- CHECK Flush mailboxes to disk
- CLOSE Close mailbox, all messages marked as deleted are removed

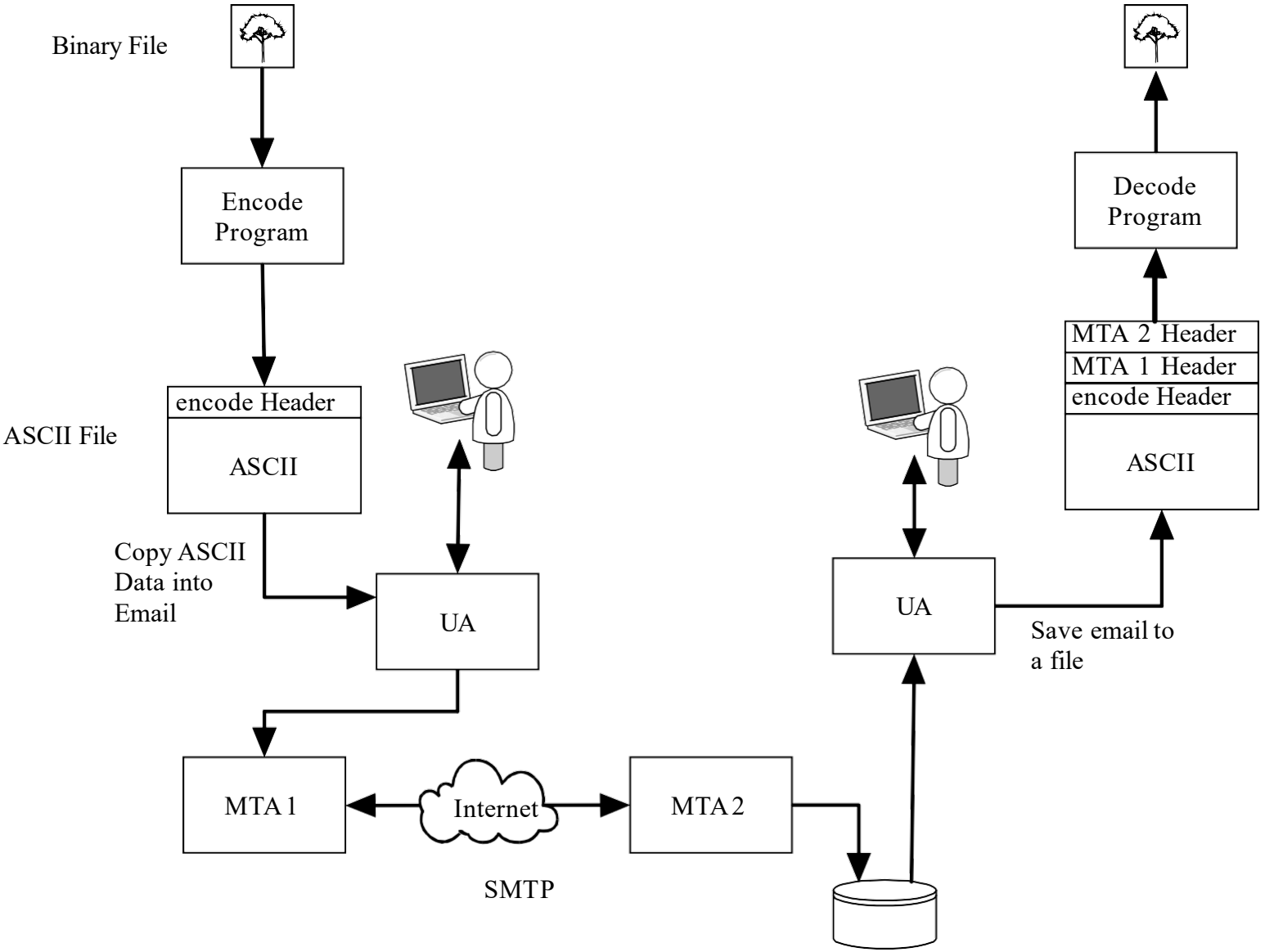
IMAP Commands

- EXPUNGE Remove messages marked as deleted
- SEARCH criteria Search the mailbox for messages that match
- FETCH message-set get message
- PARTIAL message len get partial message
- STORE
- COPY message-set Mailbox copy a message to another mailbox
- UID gets unique ID for messages

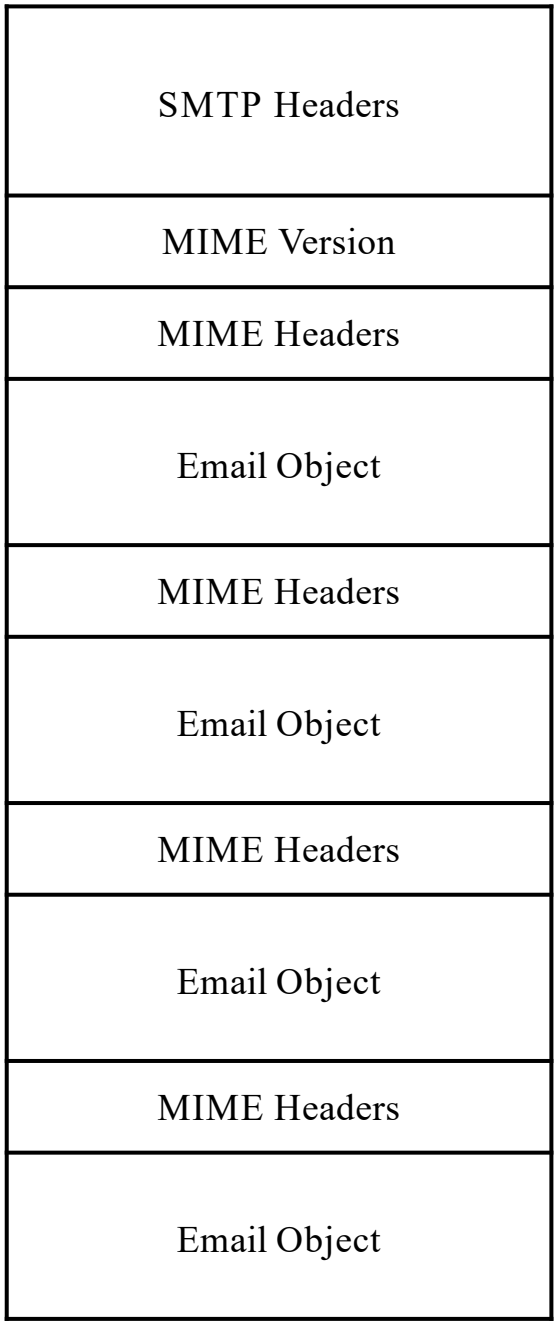
MIME

- Multipurpose Internet Mail Extensions
- Email message format
  - Embedded pictures
  - Embedded code
  - Attachments

Encode and Decode



MIME Structure



MIME Headers

MIME Header	**Function**
MIME-Version:	Indicates a MIME message. The current version is 1.1
Content-Type:	Indicates the type of content contained in the message
Content-Transfer-Encoding:	Indicates how the content is encoded
Content-Id:	Optional Identifier used for multiple messages
Content-Description:	Optional description of the object that can be displayed by the user agent
Content-Disposition:	Optional description of the method to use to display the object in receiving the user agent

Content-Type

Type	Subtype	Description
	Plain	Unformatted text
Text	Html	Text in HTML format
Multipart	Mixed	Multiple ordered objects
	Parallel	Multiple object, not ordered
	Digest	Multiple ordered RFC822 objects
	Alternative	Alternate methods of representing the same object
Message	RFC822	Encapsulated message
	Partial	Part of a larger message
	External-Body	Object is a reference to an external message
Image	JPEG	JPEG Image
	GIF	GIF Image
Video	MPEG	MPEG movie
Audio	Basic	Audio object
Application	Postscript	Adobe Postscript object
	Octet-stream	8 bit binary object

```
Email Header
MIME-Version: 1.0
UA Header
Content-Type: multipart/mixed;
 boundary="------------090603080000040609050705"

This is a multi-part message in MIME format.
```

```
----------090603080000040609050705
Content-Type: multipart/alternative;
 boundary="------------000407030803000901080005"

  --------------000407030803000901080005
  Content-Type: text/plain; charset=ISO-8859-1;
  format=flowed
  Content-Transfer-Encoding: 7bit

  ASCII text message

  --------------000407030803000901080005

  Content-Type: multipart/related;
   boundary="------------080803090003030603090002"


  --------------080803090003030603090002
  Content-Type: text/html; charset=ISO-8859-1
  Content-Transfer-Encoding: 7bit

  HTML Text
   <img src="cid:part1.09040604.05020804@iastate.edu"
  alt=""><br>
  HTML Text

  --------------080803090003030603090002
  Content-Type: image/gif;
   name="logo.gif"
  Content-Transfer-Encoding: base64
  Content-ID: <part1.09040604.05020804@iastate.edu>
  Content-Disposition: inline;
   filename="logo.gif"

  GIF File in base64

  --------------080803090003030603090002--

  --------------000407030803000901080005--
```

OR

```
--------------090603080000040609050705
Content-Type: image/gif;
 name="logo.gif"
Content-Transfer-Encoding: base64
Content-Disposition: inline;
 filename="logo.gif"

GIF File in base64

--------------090603080000040609050705--
```

Security Issues of Emails

- Phishing
  - Draw confidential information from victim (e.g. passwords)
- Privacy breach
  - Sender wants to track email recipients
- SPAM
  - Unwanted emails (e.g. advertisement)
- Malware

Security Issues of Emails

- Phishing
  - Draw confidential information from victim (e.g. passwords)
- Privacy breach
  - Sender wants to track email recipients
- SPAM
  - Unwanted emails (e.g. advertisement)
- Malware

Phishing

- Phishing = "Password Fishing"
  - Victim receives email with link to fake Web site and clicks link
  - Victim enters confidential data (e.g. passwords) assuming he is on a trusted Web site
  - Attacker misuses the entered data
- The tricks …
  - Sending mass emails is very easy and cheap
  - Sender addresses in emails are not authenticated
  - Creating Web sites and mails impersonating a trusted source is easy
  - Hyperlinks to fake Web sites can be hidden in HTML mails

Phishing Emails

提升优化–邮寄系统通知!

guoshanqing@sdu.edu.cn 发给 guoshanqing

提升优化–邮寄系统通知!

guoshanqing@sdu.edu.cn 发给 guoshanqing

发件人：guoshanqing@sdu.edu.cn<abouse@xxdzj.info>
收件人：guoshanqing<guoshanqing@sdu.edu.cn>
时间：2022年11月9日(周三) 03:31
大小：7.1KB

尊敬的用户：

您好!

为加强网络安全管理，提高系统的安全性和稳定性，保障收发畅通，为用户提供优质的服务，现即将启用新版系统，有关事项通知如下：

1.用户需登陆新邮件系统将原有数据迁移至新版系统。

点此登录

2.未迁移数据的用户，其服务将被停止。

3.升级后用户名和密码均不变，用户无需修改客户端软件设置。

特此通知。

3:31:05

Security Issues of Emails

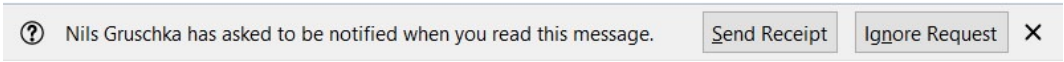
- Phishing
  - Draw confidential information from victim (e.g. passwords)
- Privacy breach
  - Sender wants to track email recipients
- SPAM
  - Unwanted emails (e.g. advertisement)
- Eavesdropping
  - Disclosure of email content on servers or during transport between servers

Security Issues of Emails

- Phishing
  - Draw confidential information from victim (e.g. passwords)
- Privacy breach
  - Sender wants to track email recipients
- SPAM
  - Unwanted emails (e.g. advertisement)
- Malware
  - Infiltrating malicious programs into recipient's computer
- Fraud
  - Contact medium for deception (e.g. financial fraud)

Email Tracking

- The sender might want to know:has the recipient received/read the email?
- Possibility 1: explicit request + receipt
  - user must confirm mail receipt for finishing a business process
  - hardly used any more



- Possibility 2: implicit tracking (mainly for SPAM or phishing)
  - does the email address exist?
  - does the email bypass SPAM filters?
  - is the recipient viewing the mail (or deleting it)?

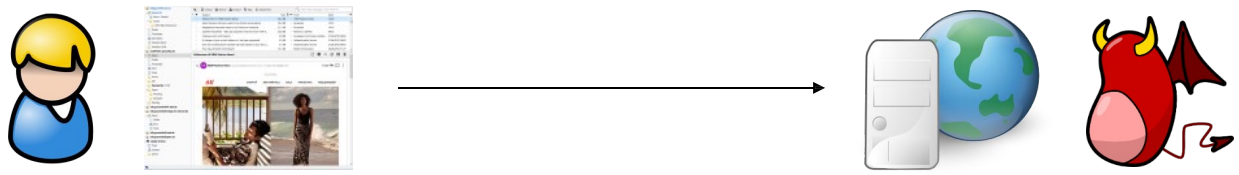
Email Tracking: Images

- Many newsletters contain HTML content:

```
This is a multi-part message in MIME format.

------=_NextPart_655_E1CC256C.E1CC256C
Content-Type: text/plain; charset="windows-1252"  Content-Transfer-
Encoding: 8bit

Hvis du ikke kan se HTML-version af nyhedsbrevet, skal du klikke på dette link:  http://rt1-
t.autoemail.hm.com/r/?id=t36ae2b3,c895a0b,c895a47

------=_NextPart_655_E1CC256C.E1CC256C
Content-Type: text/html; charset="windows-1252"  Content-Transfer-
Encoding: quoted-printable

<!DOCTYPE html>
<html style="border:0;margin:0;outline:0;padding:0">
<head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
</head>
<body style="background:#fff;border:0;color:#000;line-height:1;margin:0>
    <img src="https://s1-cdn.hm.com/global/assets/1.0.112/images/social/twitter-box.png">
```

Email Tracking: Images

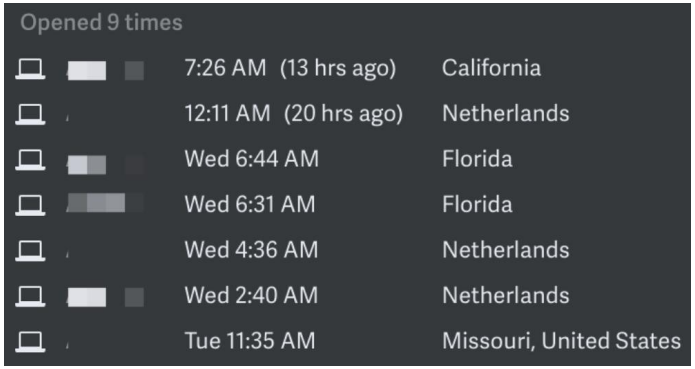
- Mail program receives email in HTML format
- HTML document contains image tags (located on Web server of the mail sender)
  - e.g.: `<img src="https://cdn.hm.com/logo/hm-logo-light-red.png">`
- Mail program downloads the images for rendering the HTML mail
- Web server owner (= mail sender) logs the request and can analyze the URL



Email Tracking ... as a Service

Mike Davidson
@mikeindustries

Superhuman is a surveillance tool that intentionally violates privacy by notifying senders every time their emails have been viewed by recipients. I would never trust this company. Only way to make sure your own privacy isn't violated is to disable images in your own email app.



Security Issues of Emails

- Phishing
  - Draw confidential information from victim (e.g. passwords)
- Privacy breach
  - Sender wants to track email recipients
- SPAM
  - Unwanted emails (e.g. advertisement)
- Malware

SPAM

- What are the reasons for the huge amount of SPAM?
  - Sending mass emails is very easy and cheap
  - Sender addresses are not authenticated
  - Sender domains are not authenticated
  - *Open relay* server accept and forward

Security Issues of Emails

- Phishing
  - Draw confidential information from victim (e.g. passwords)
- Privacy breach
  - Sender wants to track email recipients
- SPAM
  - Unwanted emails (e.g. advertisement)
- Malware

Malware

- Email is still the main infection source for malware
- Example: Locky

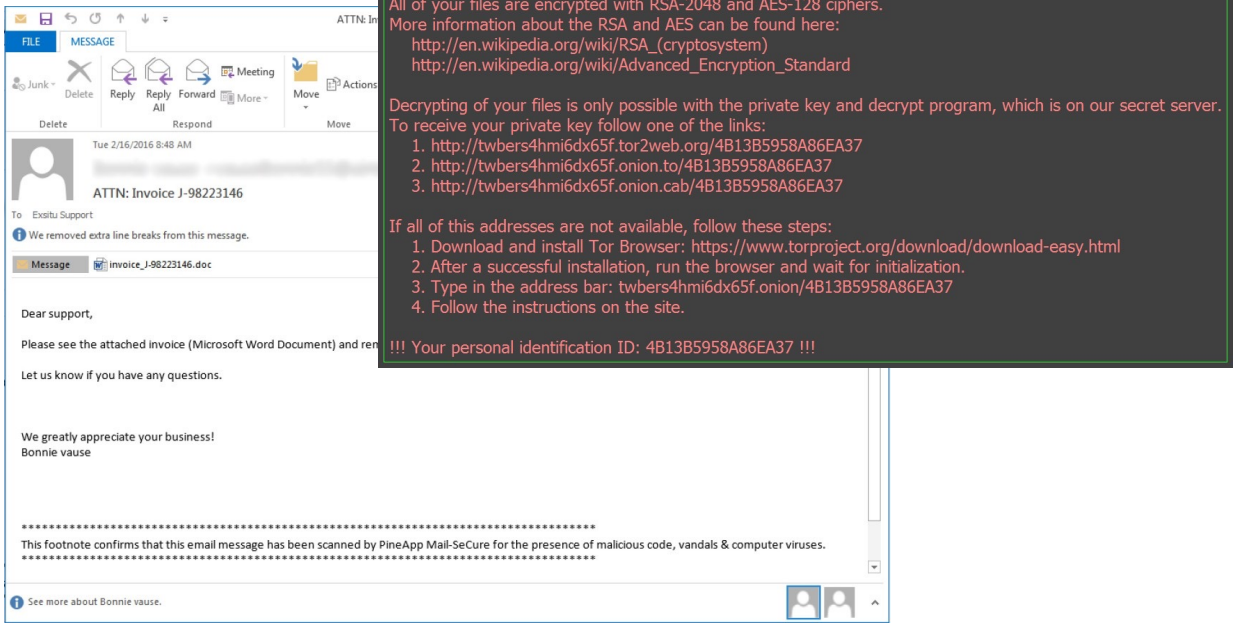
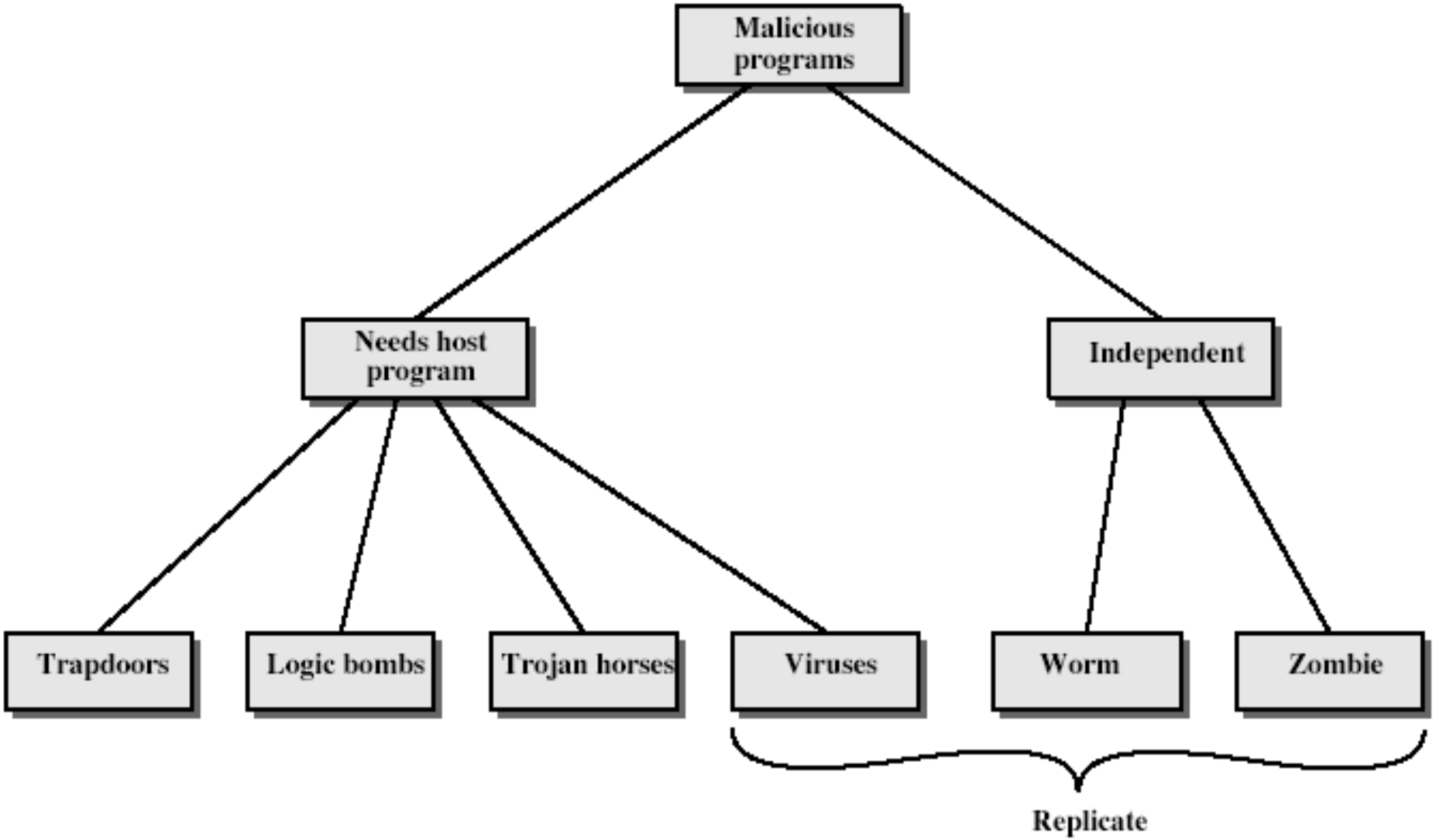


Image Source: mcafee.com

Malicious Program's Evolution

Malware

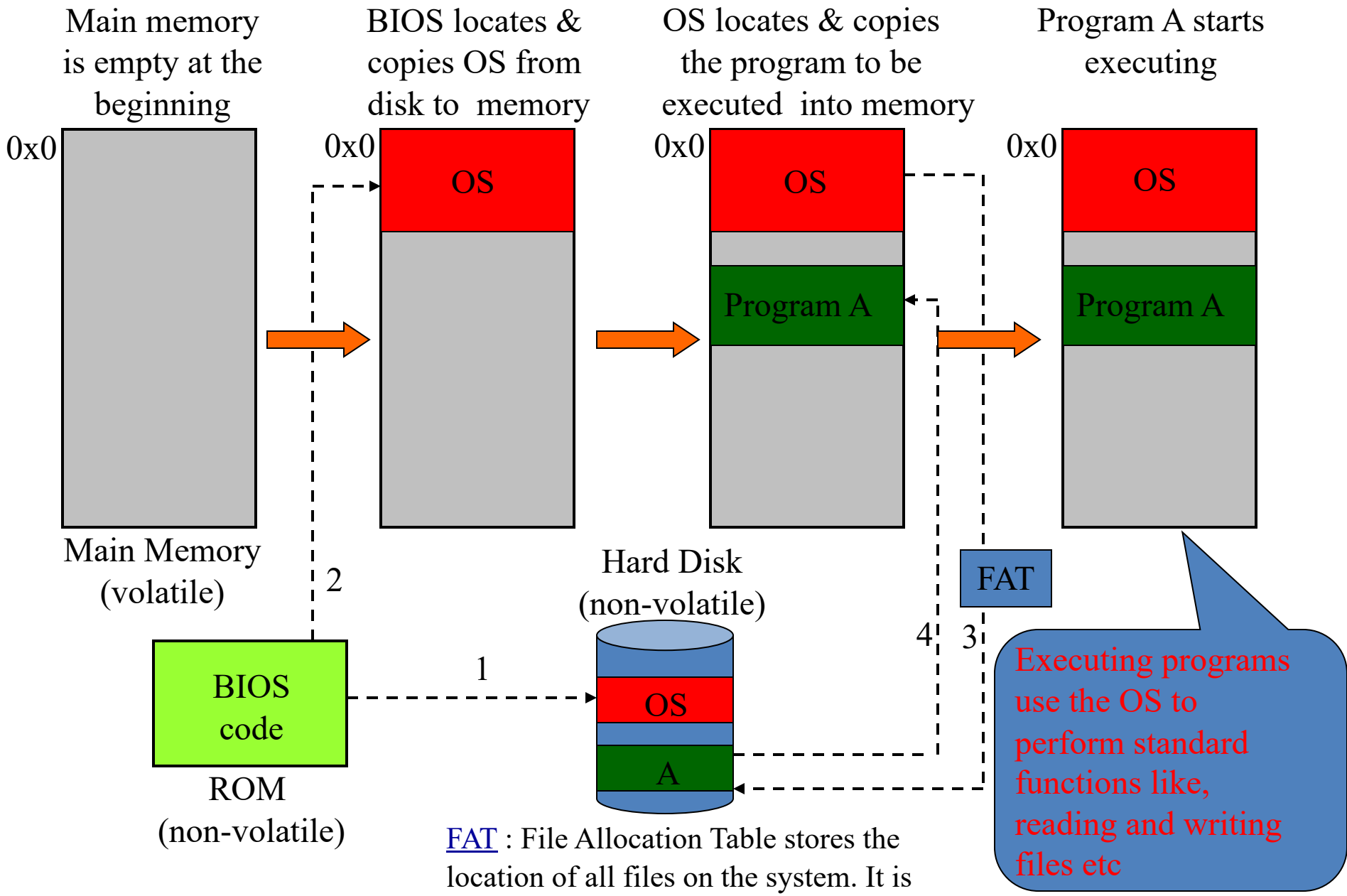

Malicious programs
Needs host program
Independent
Trapdoors
Logic bombs
Trojan horses
Viruses
Worm
Zombie
Replicate

Viruses

Viruses

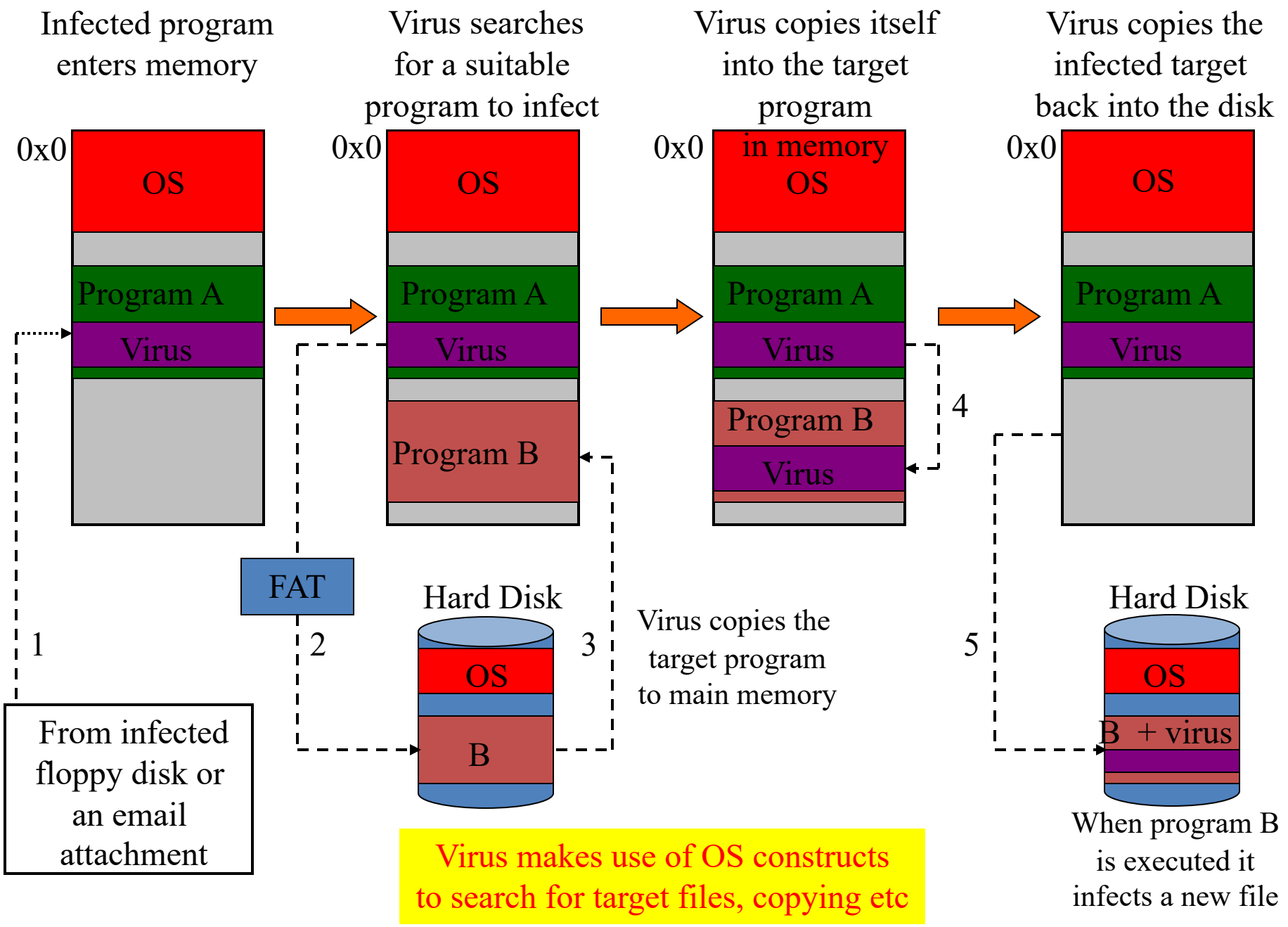
- Definition from RFC 1135: A *virus* is a piece of code that inserts itself into a host, including operating systems, to propagate. It cannot run independently. It requires that its host program be run to activate it.

Steps in Normal Program Execution



FAT : File Allocation Table stores the location of all files on the system. It is maintained by the OS.

Virus Infection Mechanism



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:

}
```

Trojan（Rootkits)

- A destructive program that masquerades as a benign application. Unlike viruses, Trojan horses do not replicate themselves but they can be just as destructive

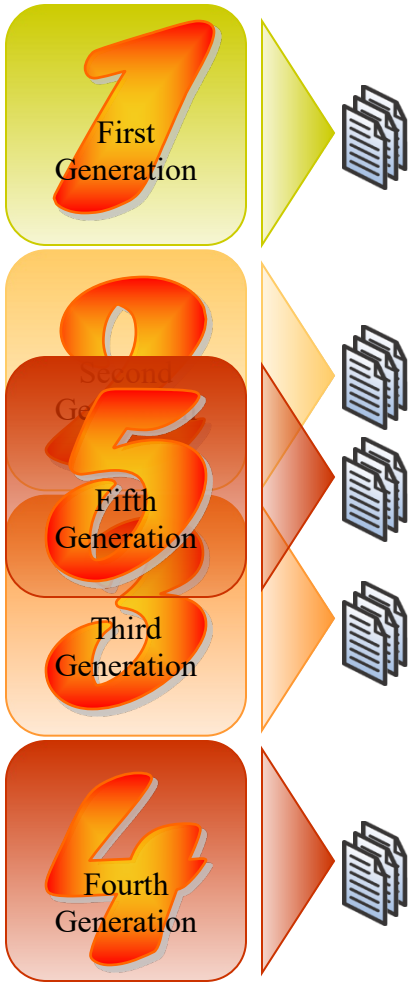
Trojan Horse

- Programs that appear to have one function but actually perform another.
- Modern Trojan Horse: resemble a program that the user wishes to run - usually superficially attractive
    - E.g., game, software upgrade etc
- When run performs some additional tasks
    - Allows attacker to indirectly gain access they do not have directly
- Often used to propagate a virus/worm or install a backdoor

Motivation

- Hackers want to keep access to a successfully compromized box.
- At the same time, they want to remain undetected and thus need to hide their presence and traces.
- All hacker activities and data related to those activities shall be invisible to legitimate users. Any permanent trace should be avoided, if possible.

History of Rootkits



Primitive
Binary file replacement (password logging / UNIX)
Hiding traces/tracks (log cleaners)

More advanced hiding - "stealthy"
(HxDef,HE4Hook)
Hooking techniques

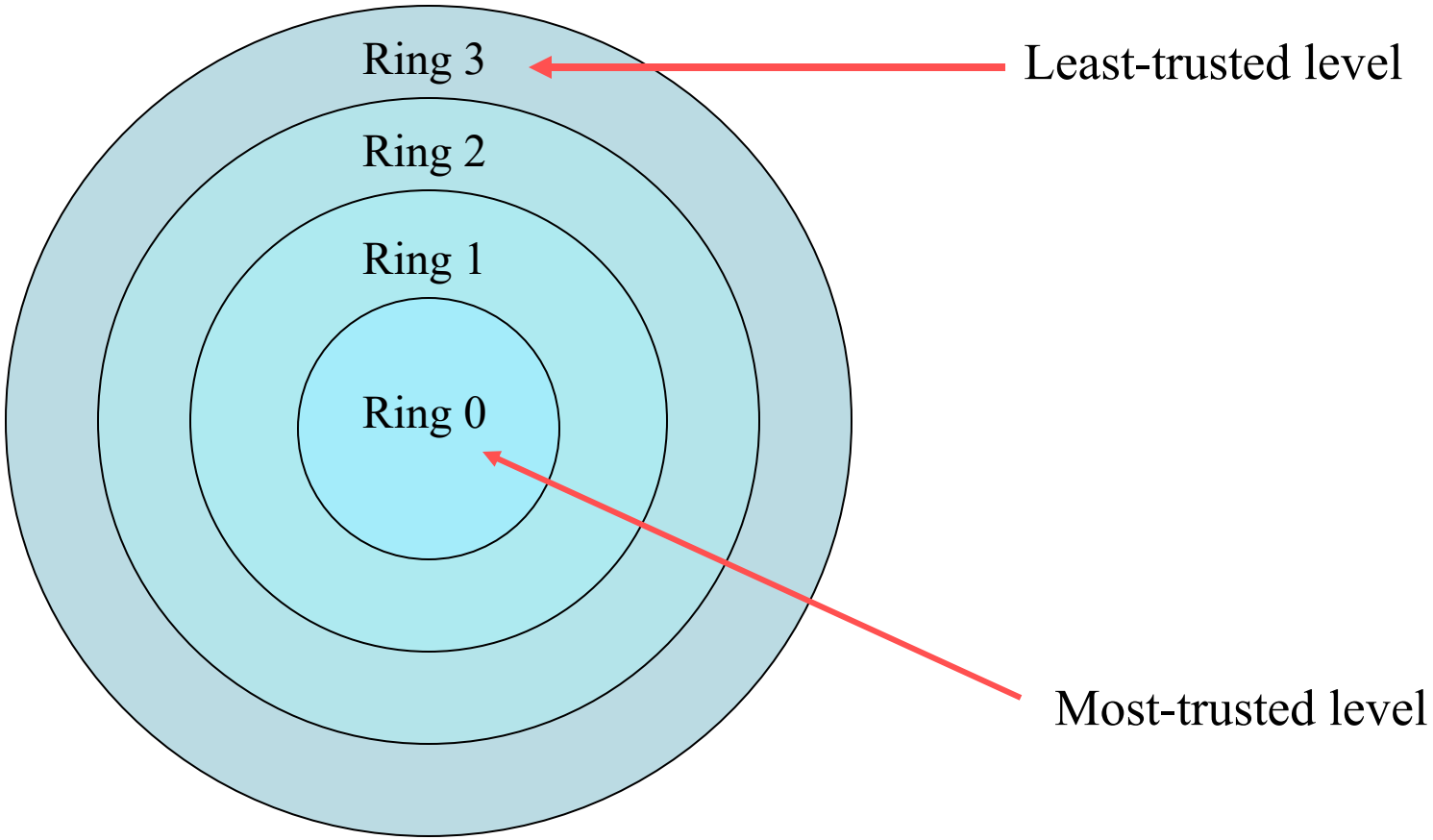
Hardware Virtualization

Direct dynamic manipulation of kernel structures (FU)
Difficult for detection software to identify

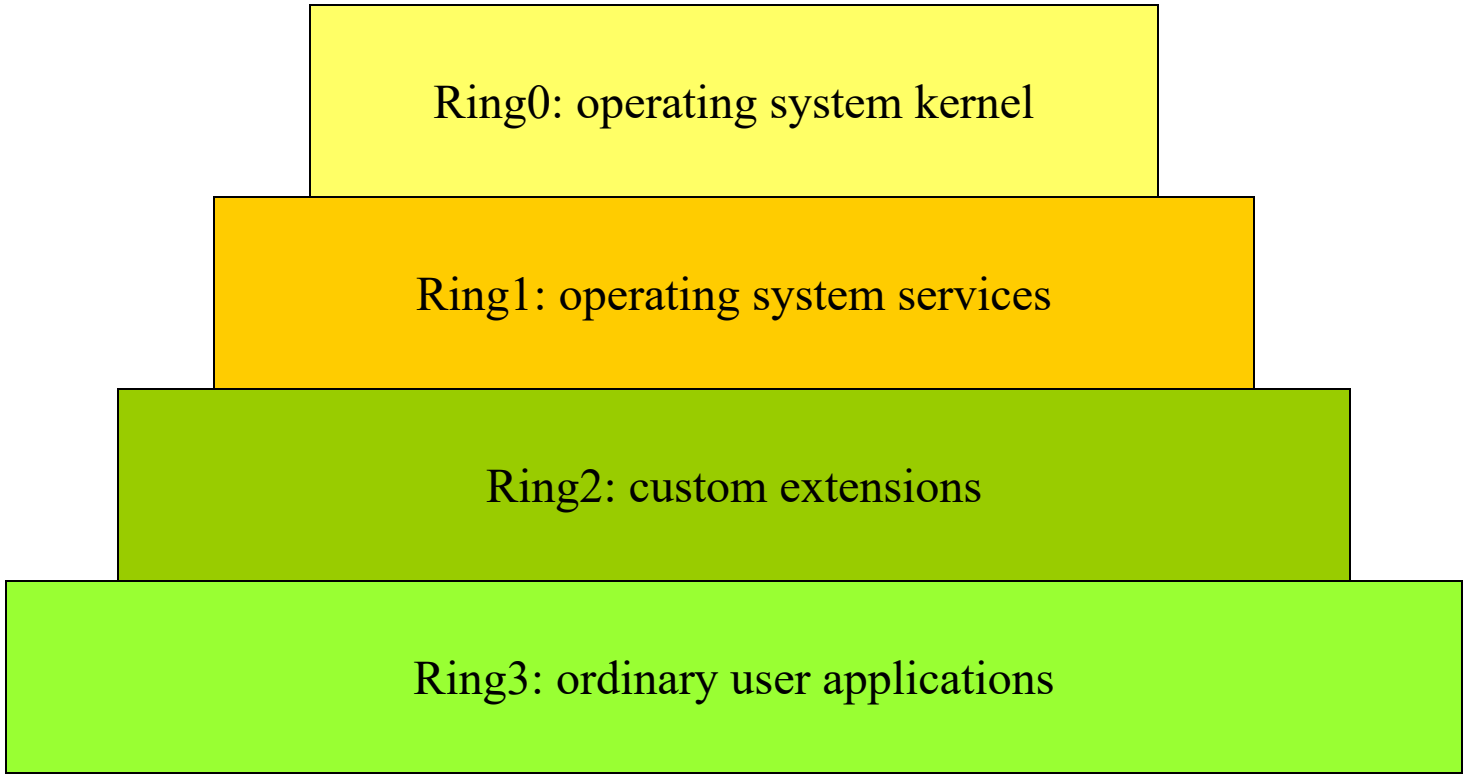
Advanced Memory hooking/hiding (Shadow Walker)
Used in collusion with 3rd Generation rootkit
Extremely "stealthy"

Reference: http://www.phrack.org/archives/63/p63-0x08_Raising_The_Bar_For_Windows_Rootkit_Detection.txt

Four Privilege Rings

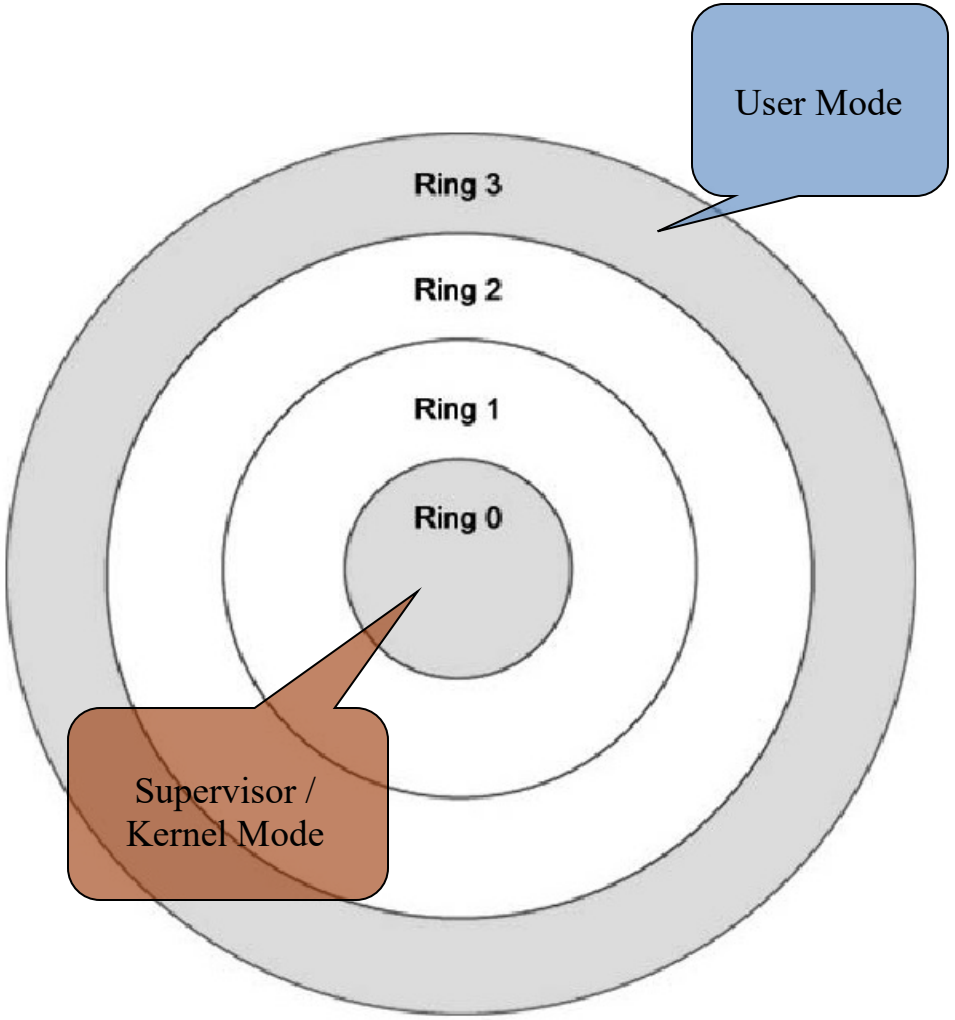


CPU

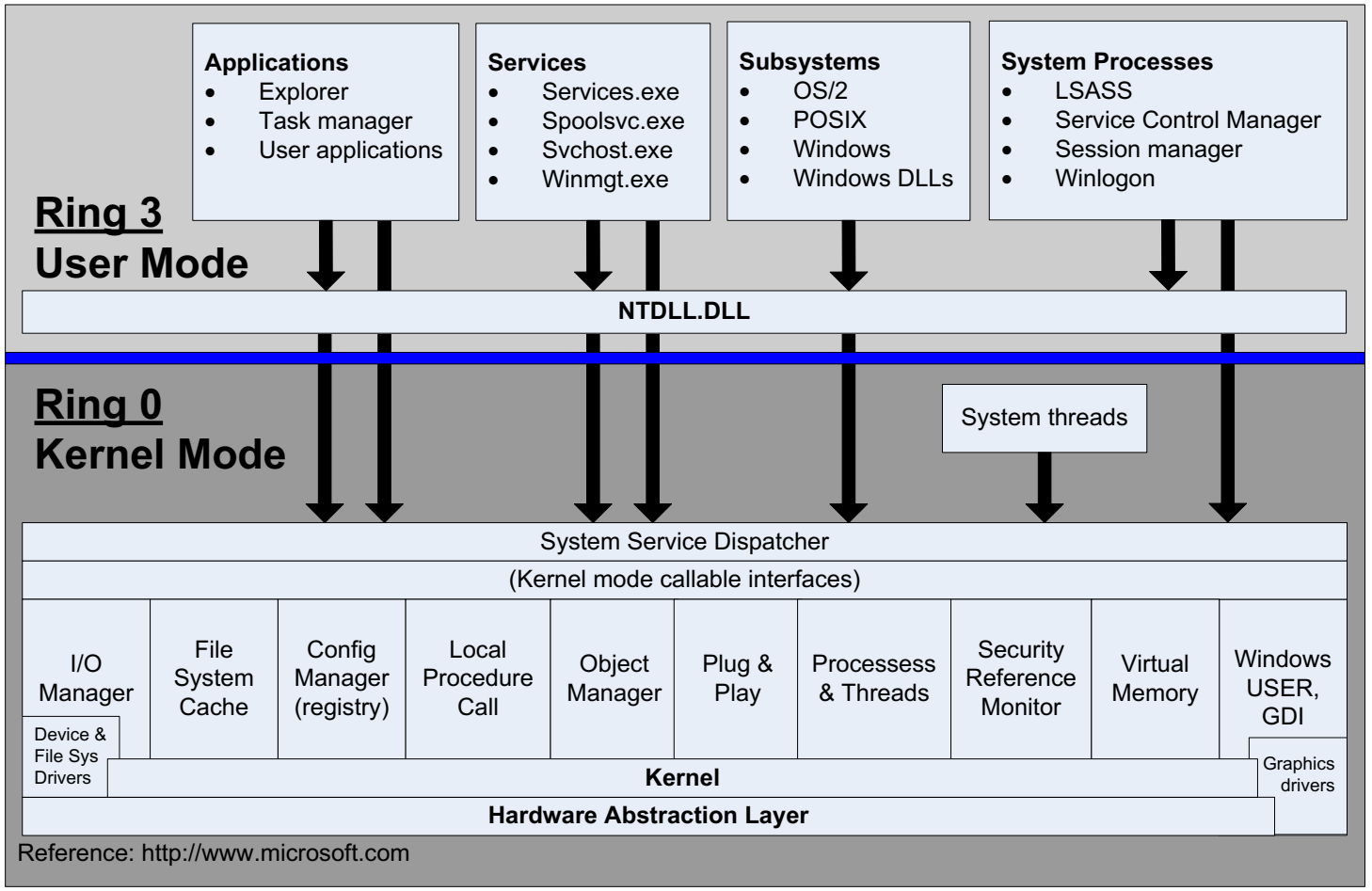

Ring0: operating system kernel
Ring1: operating system services
Ring2: custom extensions
Ring3: ordinary user applications

Operating System Design

- Intel has four privilege levels or rings
- windows and many other OS vendors use only two rings
  - **User Mode** : In this level some restriction in accessing the system hardware and certain memory regions apply. The address space a user program is restricted to the applicati memory maps.
  - **Kernel Mode** : Everything is allowed

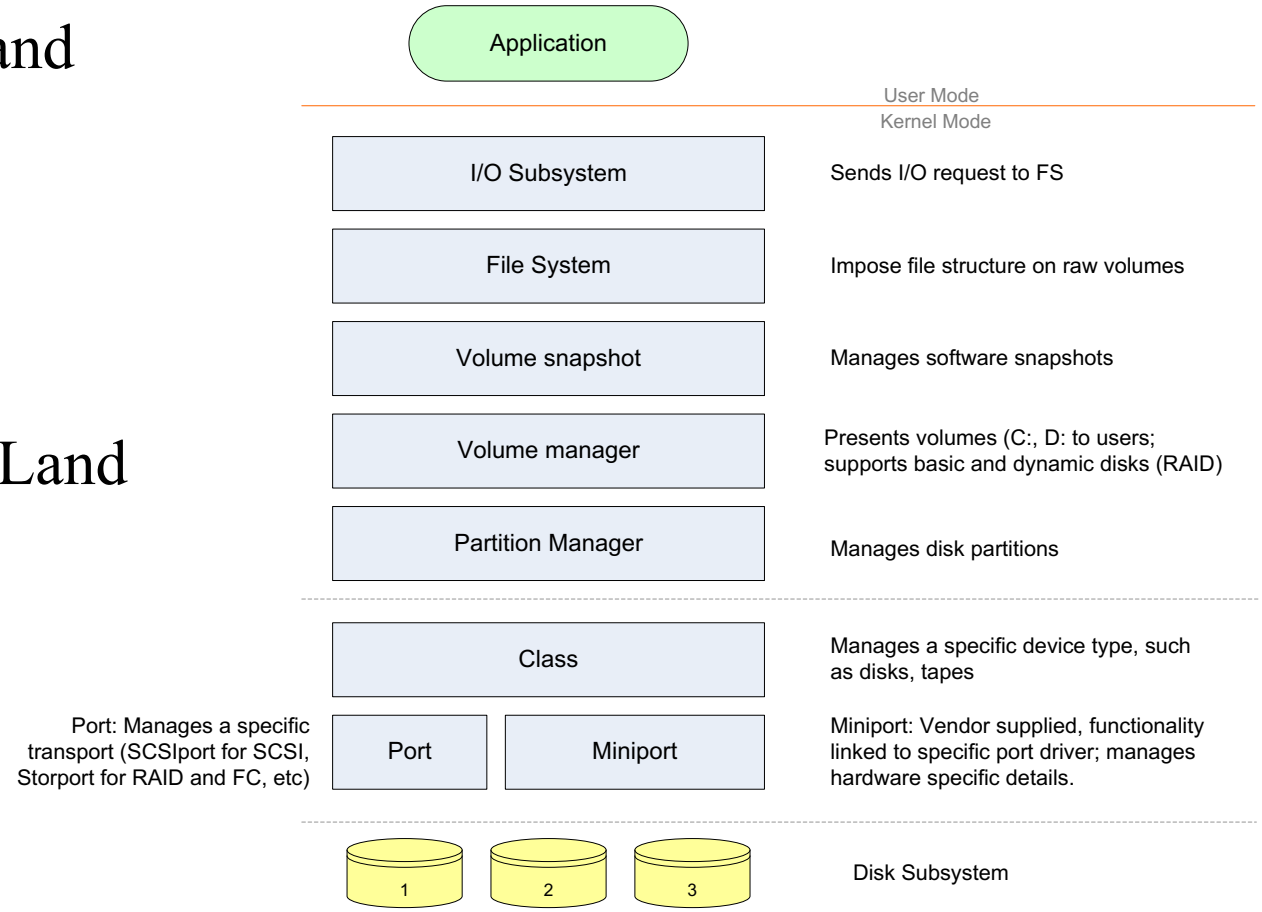


Windows Architecture

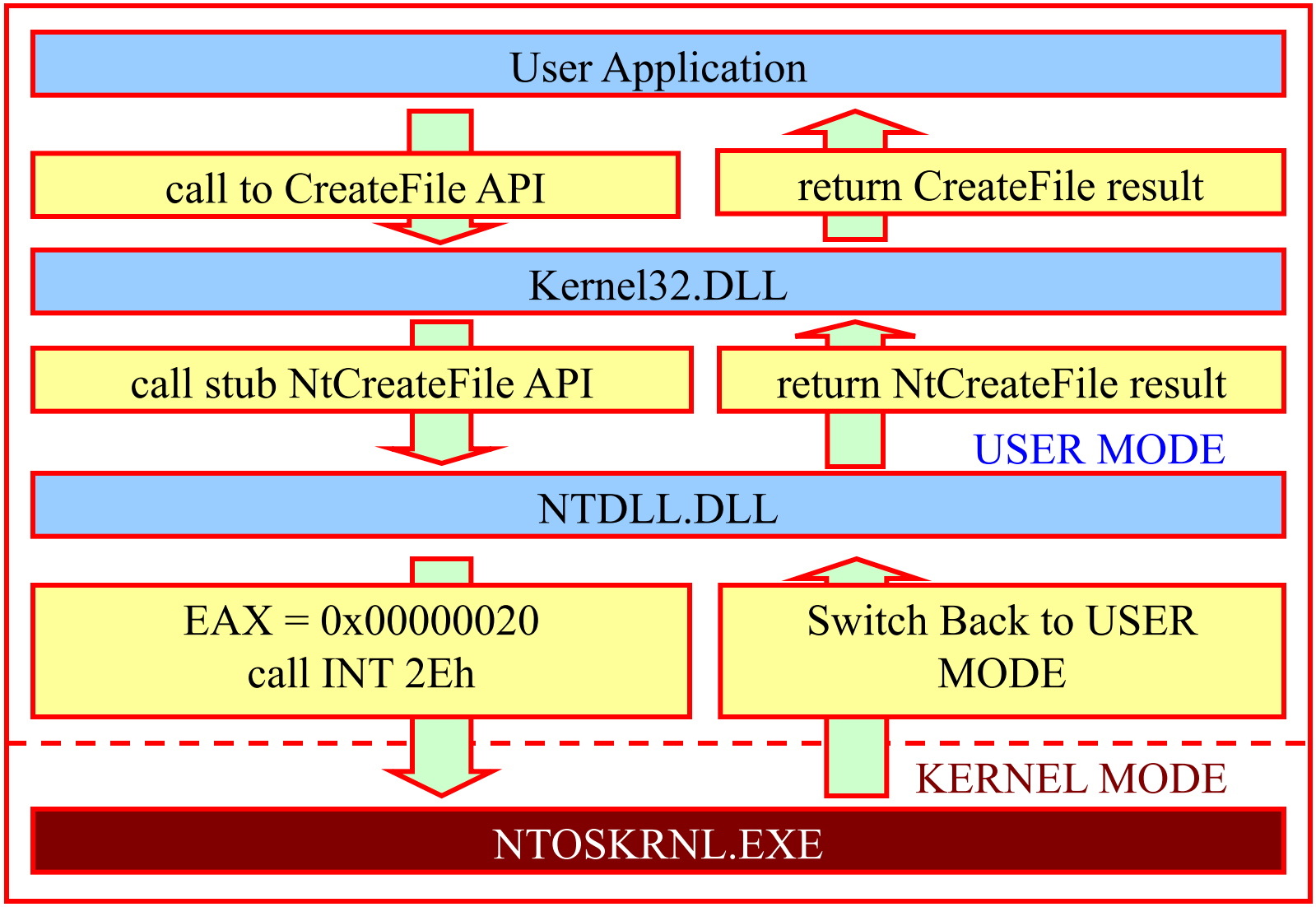


How does Rootkit work?

- Ring 3 – User Land
  - **User**
  - **Administrator**
  - **System**
- Ring 0 – Kernel Land
  - **Drivers**



System Service Call Cycle



NTDLL Interface

- Kernel32.DLL imports solely on the library NTDLL.DLL
- NTDLL.DLL is an interface to Int 2Eh function of Windows NT
- Int 2Eh signals a need to switch from user mode to kernel mode
- Int 2Eh is internally known as KiSystemService().

Getting Into The Root

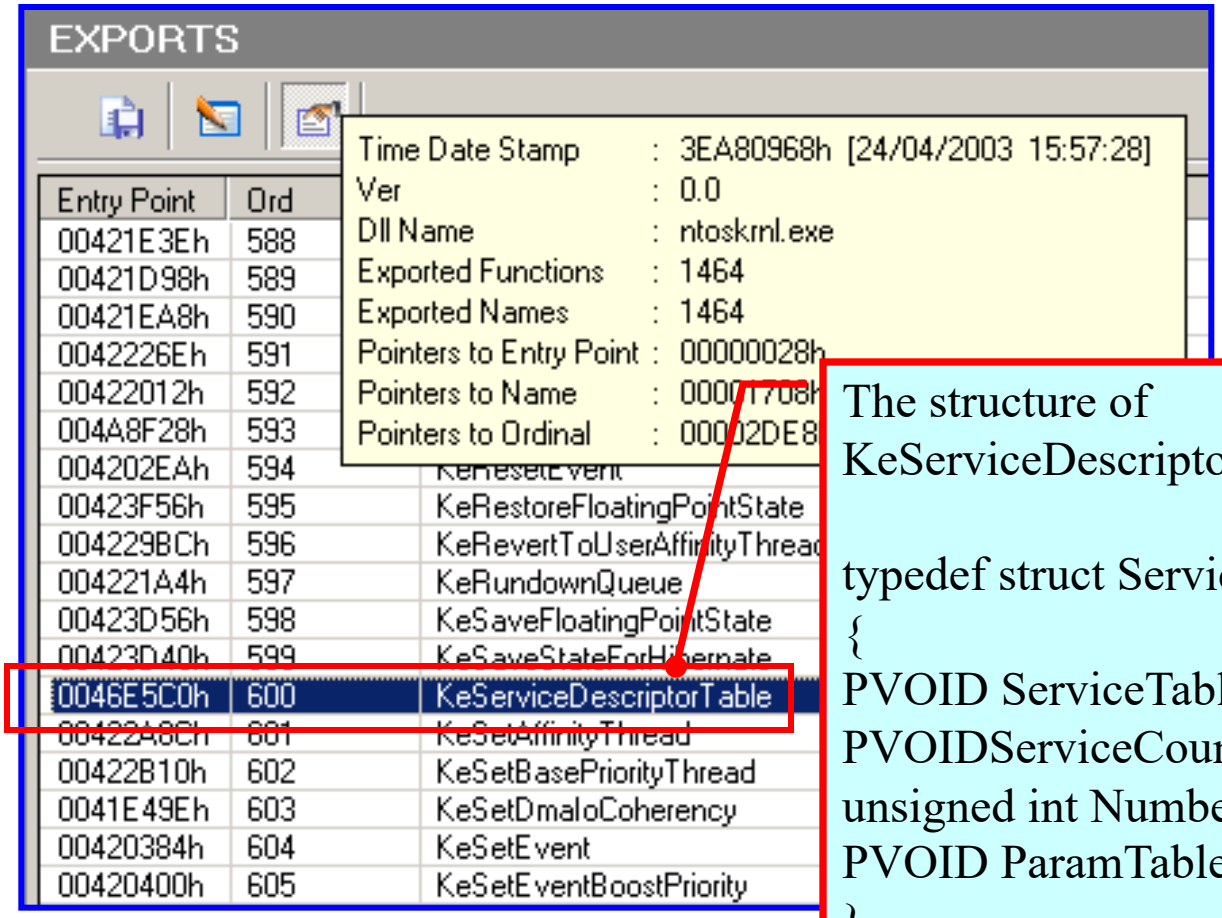
Application: Call to CreateFile() API

Kernel32.DLL: Call to NtCreateFile() -- Native API

Invokes KiSystemService()

NTOSKRNL.EXE: Call to KeServiceDescriptor Table

NTOSKRNL Exports



The structure of KeServiceDescriptorTable:

```
typedef struct ServiceDescriptorTable
{
PVOID ServiceTableBase;
PVOIDServiceCounterTable(0);
unsigned int NumberOfServices;
PVOID ParamTableBase;
}
```

KeServiceDescriptor Table

	ServiceTableBase	ParamTableBase
	…	…
0x20	@ NtCreateFile	0x2C bytes
	…	…
0x29	@NtCreateProcess	0x20 bytes
	…	
0x6A	@ NtOpenProcess	0x10 bytes
	…	
	…	…

Hooking System Service

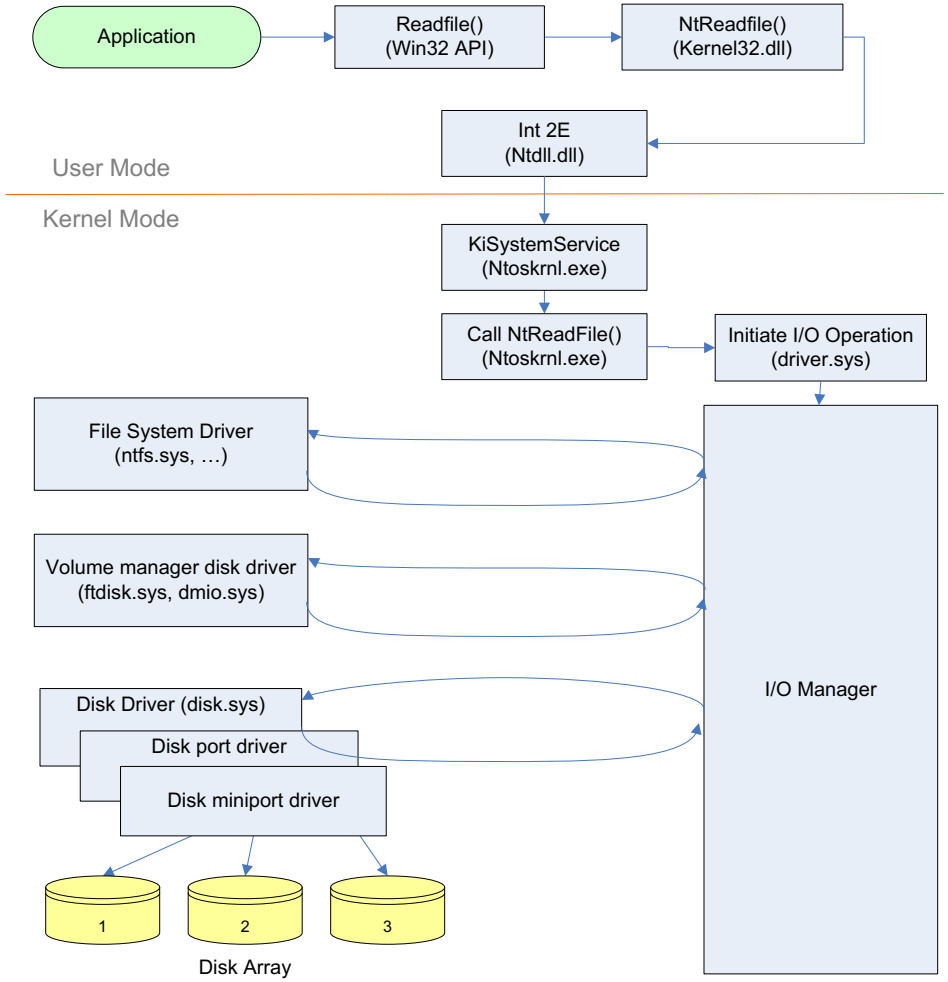
Example of NTDLL Exported Functions

```
ntdll!NtCreateFile
001B:77F95238   B820000000        MOV         EAX,00000020
001B:77F9523D   8D542404          LEA         EDX,[ESP+04]
001B:77F95241   CD2E              INT         2E
001B:77F95243   C22C00            RET         002C
```

```
ntdll!NtCreateProcess
001B:77F92D2C   B829000000        MOV         EAX,00000029
001B:77F92D31   8D542404          LEA         EDX,[ESP+04]
001B:77F92D35   CD2E              INT         2E
001B:77F92D37   C22000            RET         0020
```

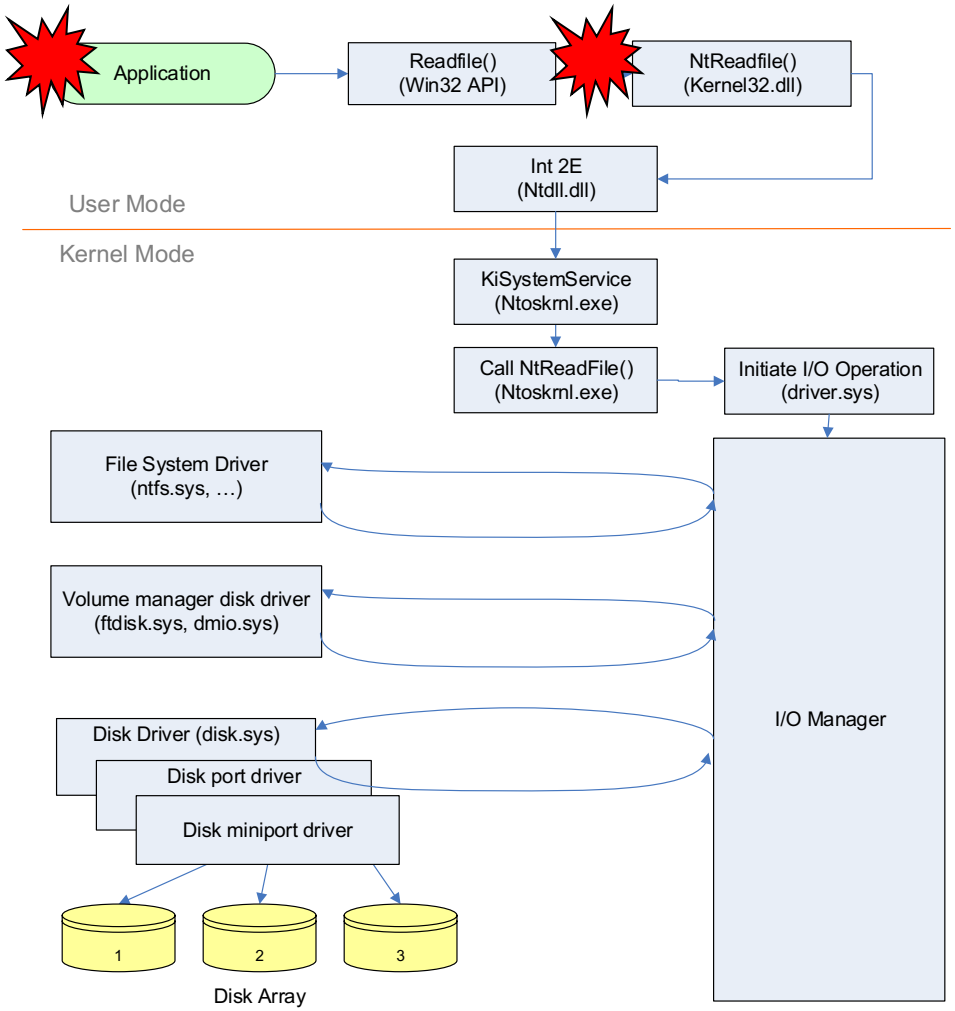
```
ntdll!NtOpenProcess
001B:77F8E5DE   B86A000000        MOV         EAX,0000006A
001B:77F8E5E3   8D542404          LEA         EDX,[ESP+04]
001B:77F8E5E7   CD2E              INT         2E
001B:77F8E5E9   C21000            RET         0010
```

What Happens When You Read a File?



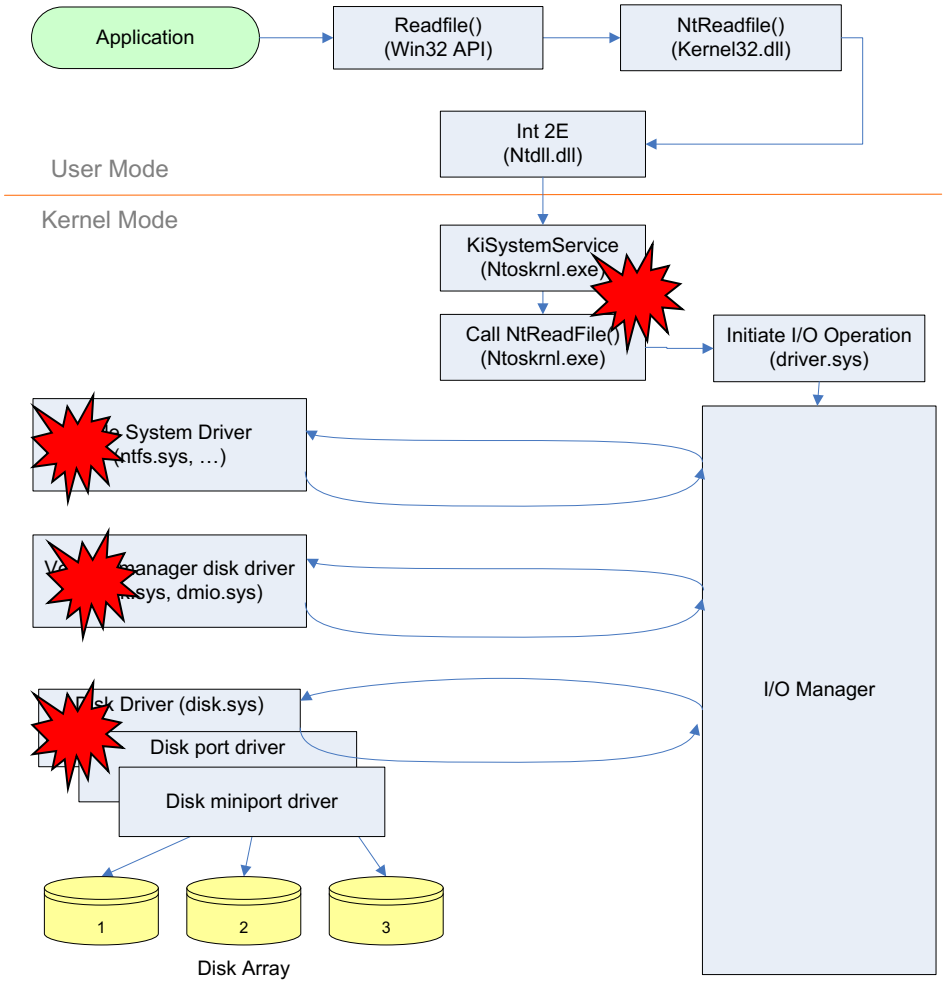
- Readfile() called on File1.txt
- Transition to Ring 0
- NtReadFile() processed
- I/O Subsystem called
- IRP generated

Userland (Ring 3) Rootkits



- Binary replacement eg modified Exe or Dll
- Binary modification in memory eg He4Hook
- User land hooking eg Hacker Defender
  - **IAT hooking**

Kernel (Ring 0) Rootkits



- Kernel Hooking

  E.g. NtRootkit

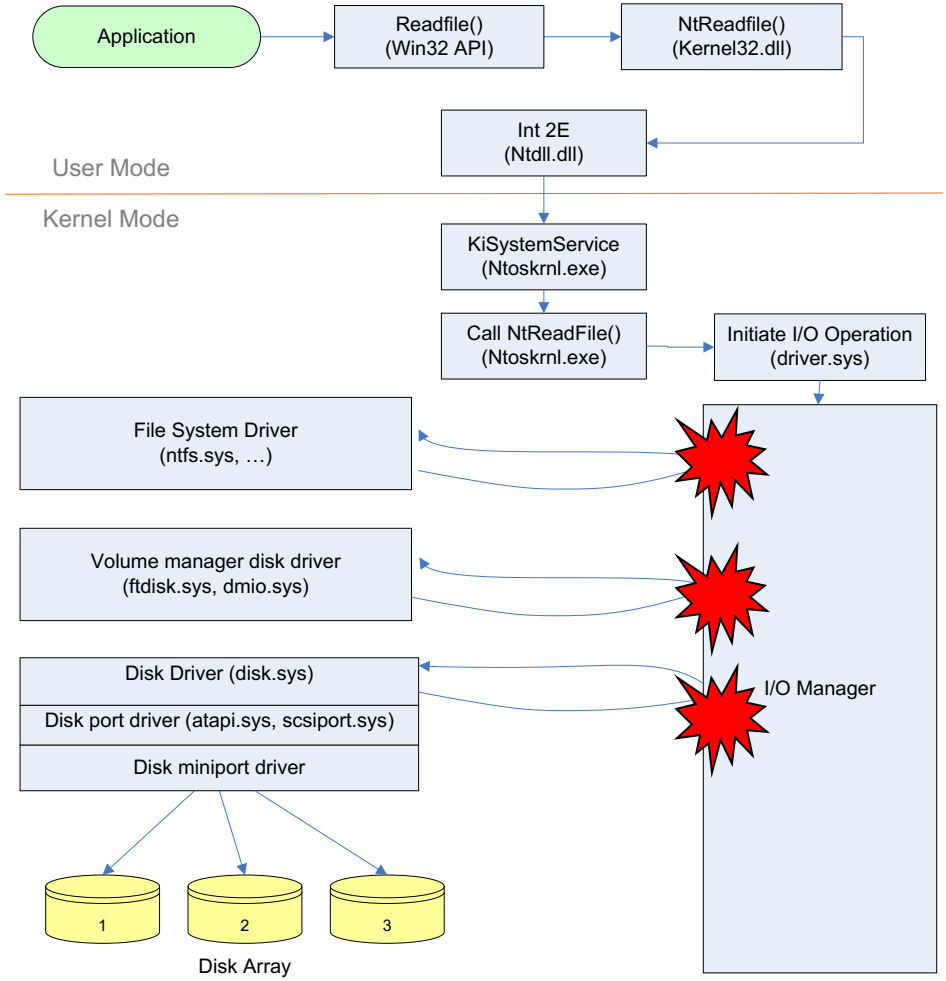
- Driver replacement

  E.g. replace ntfs.sys with ntfss.sys

- Direct Kernel Object Manipulation − DKOM

  E.g. Fu, FuTo

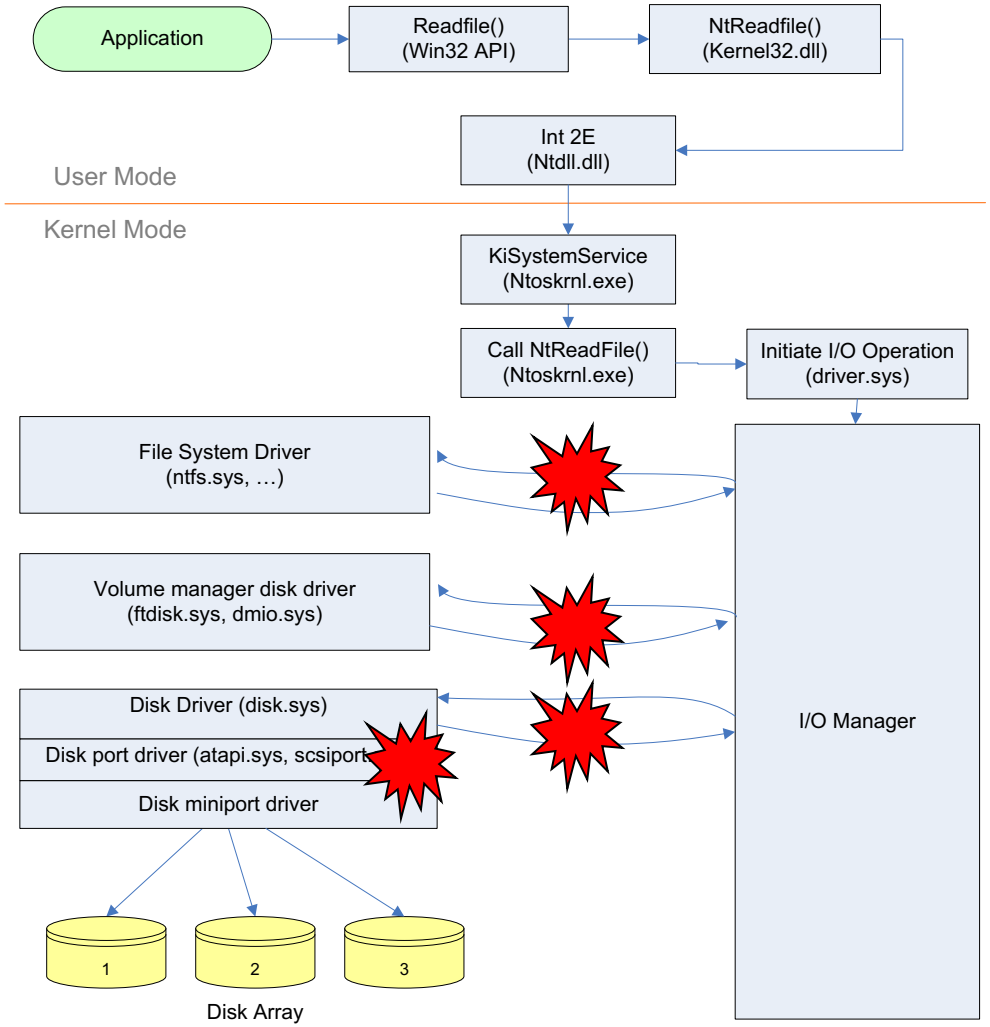
Kernel (Ring 0) Rootkits



- IO Request Packet (IRP) Hooking
  - IRP Dispatch Table

E.g. He4Hook

Kernel (Ring 0) Rootkits



- Filter Drivers
  - The official Microsoft method
- Types
  - File system filter
  - Volume filter
  - Disk Filter
  - Bus Filter

Classfical ways for hiding various objects

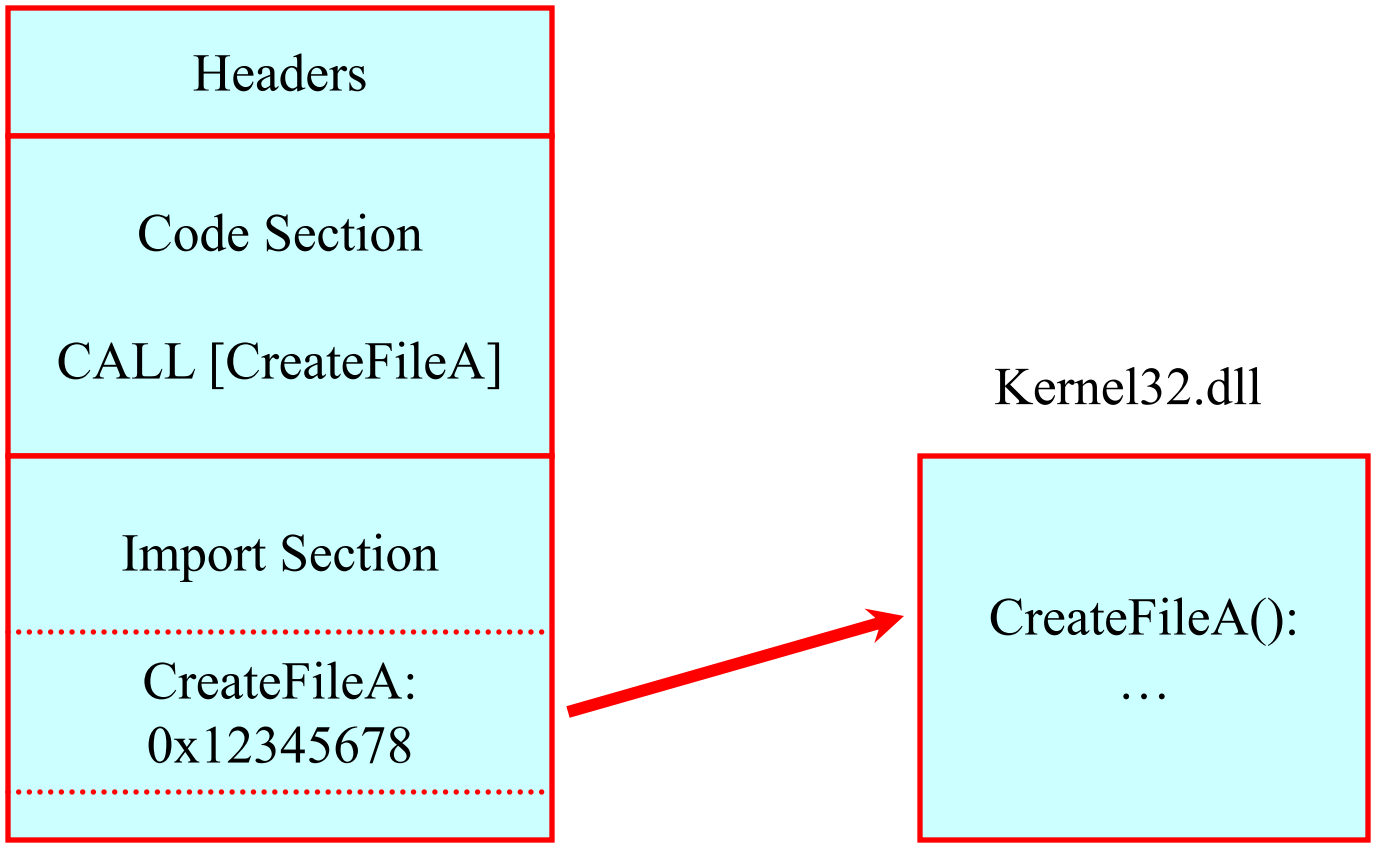
- Replacing files (e.g. DLLs)
- Hooking DLL's functions (API/IAT hooking)
- Modifying DLL's functions (Raw Code Change)
- Hooking entries in SST/KiServiceTable (very popular)
- Hooking IDT 2Eh entry
- Modifying Kernel Code (Raw Code Change)

IAT hooking

- Import Address Table (IAT) / Export Address Table (EAT)
  - Each process and module(DLL) have their own Import Address Table (IAT) that contains the entry-point addresses of the APIs that are used.
  - Every DLL has an Export Address Table (EAT) that contains the entry-point addresses of the APIs that are implemented within the DLL.

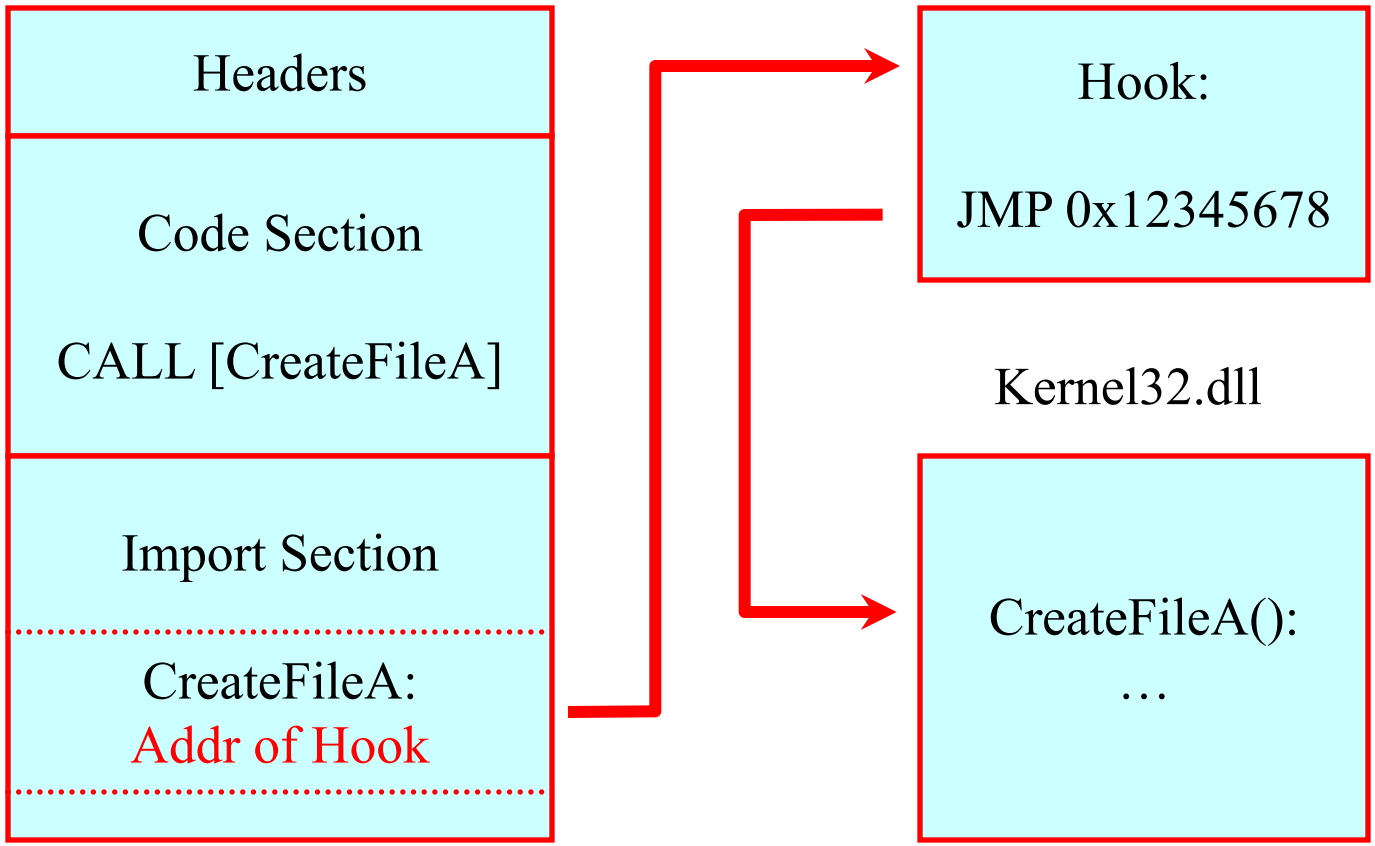
IAT Modification

PE File Before IAT Patching



IAT Modification

PE File Before IAT Patching



IAT hooking

- Powerful and simple
- Easy to detect, but
  - Legitimate hooking common
    - Methods such as DLL forwarding makes benign vs. malicious hooks hard to discern
  - Late binding
    - Applications do late-demand binding where function addresses are not resolved until called
    - Functions will not have addresses in IAT to hook!

Inline function hooking

- More powerful than IAT hooking
  - Do not have problems with binding time
  - Overwrite code bytes of target function so that no matter how it is resolved, your code will run
  - Can be used for both kernel and user functions

Inline function hooking

Original FindNextFile() API Function

```
FindNextFileA:
195D6: 55                   PUSH EBP
195D7: 8BEC                 MOV EBP, ESP
195D9: 81EC60020000         SUB ESP, 260
Continue_Here:
194DF: 53                   PUSH EBX
195E0: 8D85A0FDFFFF         LEA EAX, [EBP-260]
195DF: XX                   <…original code
                            continues…>
```

Dynamic Code Patching

Patched FindNextFile() API Function

```
FindNextFileA:
195D6: E9XXXXXXXX        JMP Hook
195DB: 90                NOP
195DC: 90                NOP
195DD: 90                NOP
195DE: 90                NOP
Continue_Here:
194DF: 53                PUSH EBX
195E0: 8D85A0FDFFFF      LEA EAX, [EBP-260]
195DF: XX                <…original code
                         continues…>

Hook: <process params>
      call Saved_Original
      <alter data>
      ret
```

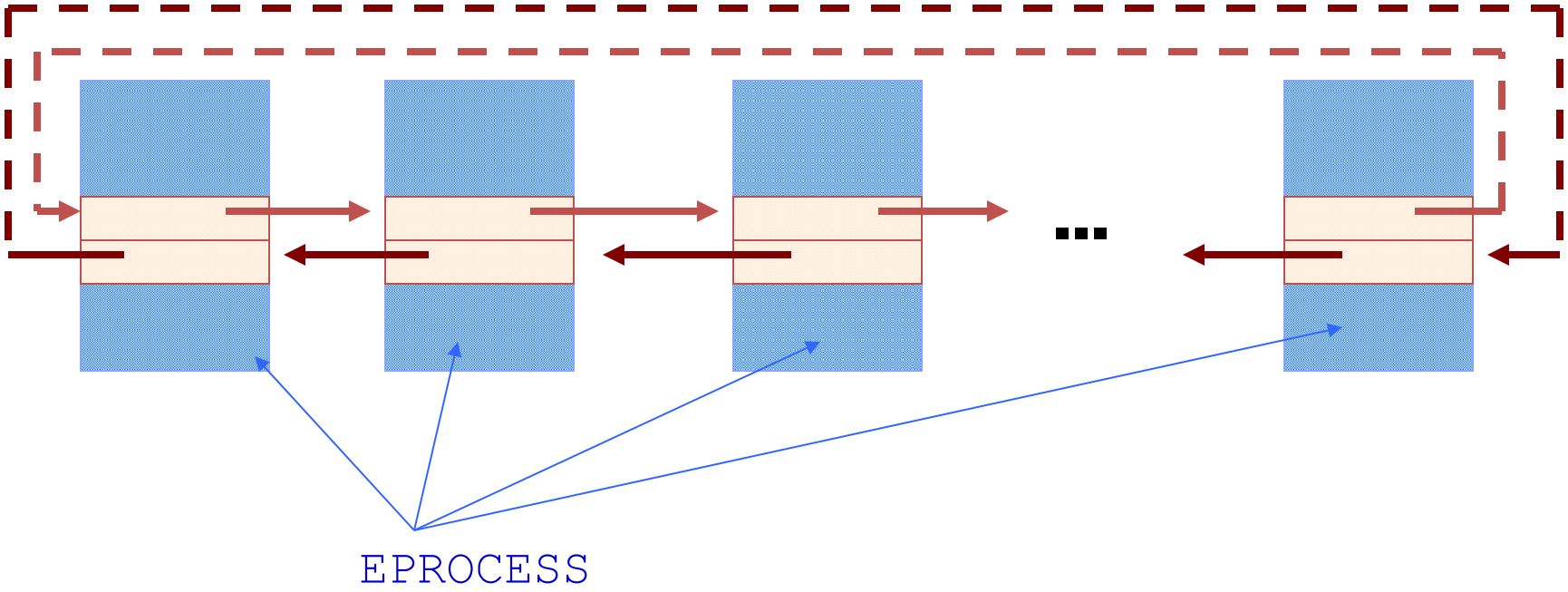
SSDT hooking

- System Service Descriptor Table
  - Kernel data structure that points to code which implements system calls in Win32, POSIX, and OS/2 subsystems
  - Indexed by system call number
- System Service Parameter Table
  - Specifies the number of bytes for the parameters of each call
- Hooking SSDT
  - Load rootkit as device driver
  - Replace SSDT entry to point to it instead of Ntoskrnl.exe or Win32k.sys
  - Later versions of Windows XP make memory that stores SSDT read-only (BSOD if you try to write)
    - Change CR0 to disable memory protection in kernel
    - Use Memory Descriptor Lists to change flags
  - HOOK_SYSCALL, UNHOOK_SYSCALL macros

Using SSDT hooks

- Hiding processes
  - Replace NTQuerySystemInformation function in SSDT
  - Hook calls original function and filters results to remove rootkit entries from SystemInformationClass buffer that is returned
  - Must update execution time statistics across all processes in list
    - If CPU does add up to 100%, someone will be suspicious

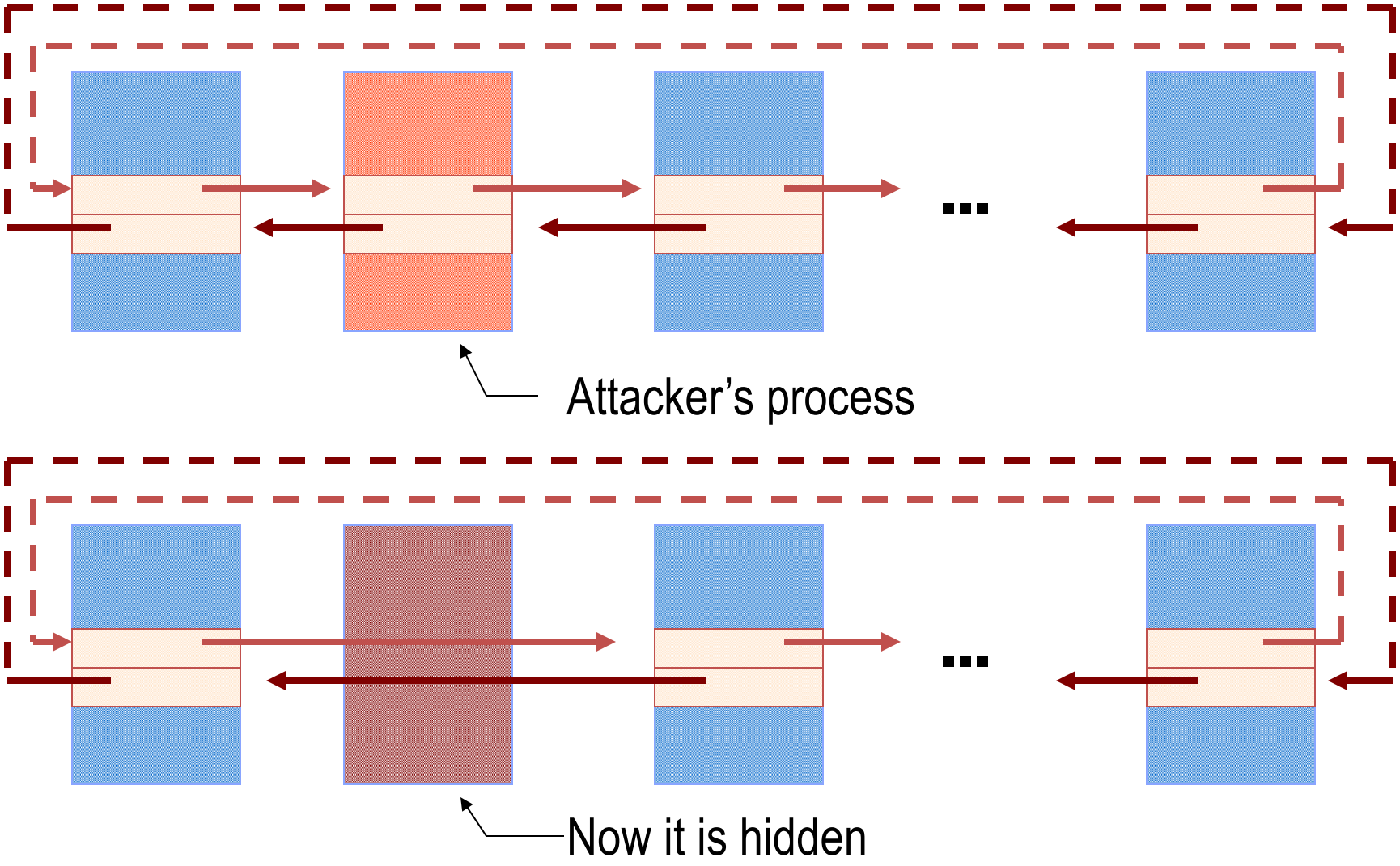
ActiveProcessLinks



All active processes in the system are kept on the single list. This list is implemented by pair of pointers in each EPROCESS block:

EPROCESS.ActiveProcessLinks

Fu rootkit



Botnets

- A botnet is a network of software robots (bots) run on zombie machines which run are controlled by command and control networks
  - IRCbots - command and control over IRC
  - Bot herder - owner/controller of network
  - "scrumping" - stealing resources from a computer
- Surprising Factoid: the IRC server is exposed.

What are botnets being used for?

piracy

mining

attacks

hosting

Activities we have seen

Stealing CD Keys:

```
ying!ying@ying.2.tha.yang PRIVMSG #atta :BGR|0981901486 $getcdkeys
BGR|0981901486!nmavmkmyam@212.91.170.57 PRIVMSG #atta :Microsoft Windows
Product ID CD Key: (55274-648-5295662-23992).
BGR|0981901486!nmavmkmyam@212.91.170.57 PRIVMSG #atta :[CDKEYS]: Search
completed.
```

Reading a user's clipboard:

```
B][!Guardian@globalop.xxx.xxx PRIVMSG ##chem## :~getclip
Ch3m|784318!~zbhibvn@xxx-7CCCB7AA.click-network.com PRIVMSG ##chem## :-
[Clipboard Data]- Ch3m|784318!~zbhibvn@xxx-7CCCB7AA.click-network.com PRIVMSG
##chem## :If You think the refs screwed the seahawks over put your name down!!!
```

DDoS someone:

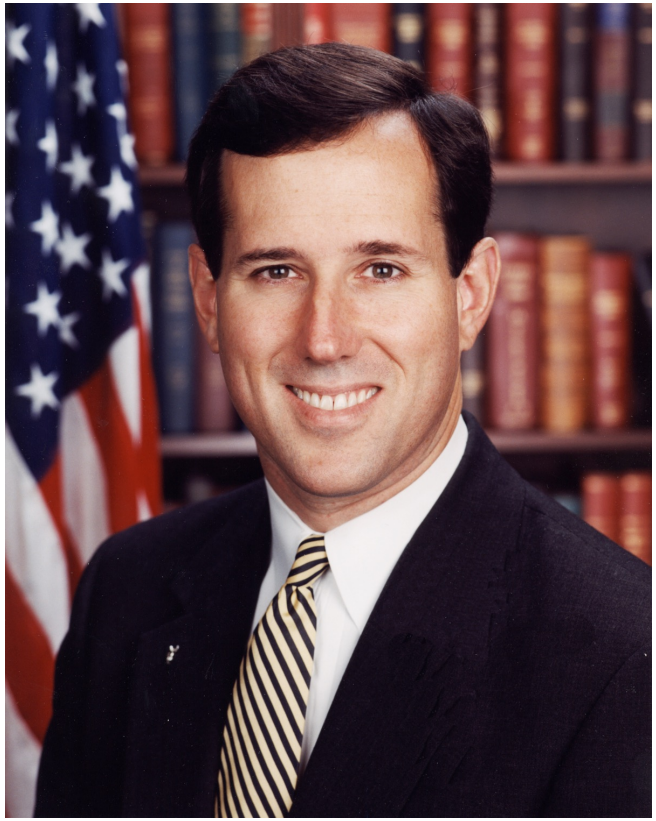
```
devil!evil@admin.of.hell.network.us PRIVMSG #t3rr0r0Fc1a :!pflood 82.147.217.39
443 1500 s7n|2K503827!s7s@221.216.120.120 PRIVMSG #t3rr0r0Fc1a :\002Packets\002
\002D\002one \002;\002>\n s7n|2K503827!s7s@221.216.120.120 PRIVMSG #t3rr0r0Fc1a
flooding....\n
```

Set up a web-server (presumably for phishing):

```
[DeXTeR]!alexo@l85-130-136-193.broadband.actcom.net.il PRIVMSG [Del]29466
:.http 7564 c:\\ [Del]38628!zaazbob@born113.athome233.wau.nl PRIVMSG _[DeXTeR]
:[HTTPD]: Server listening on IP: 10.0.2.100:7564, Directory: c:\\.
```

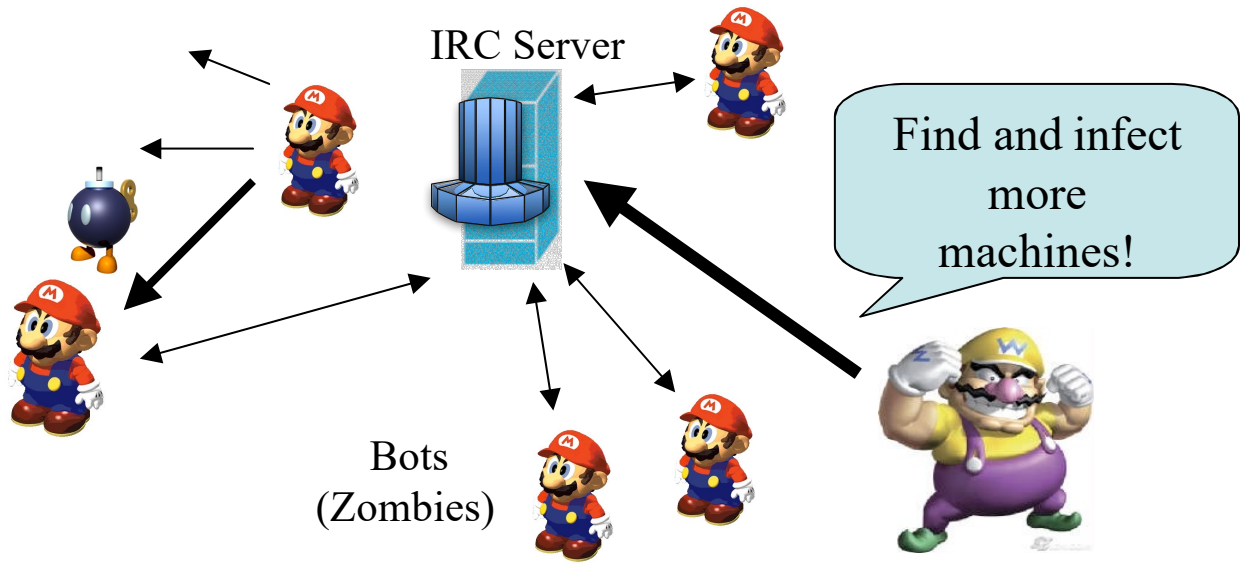
Other goals of a botnet ...

- SPAM relays
- Click fraud
- Spamdexing
- Adware



IRC botnets

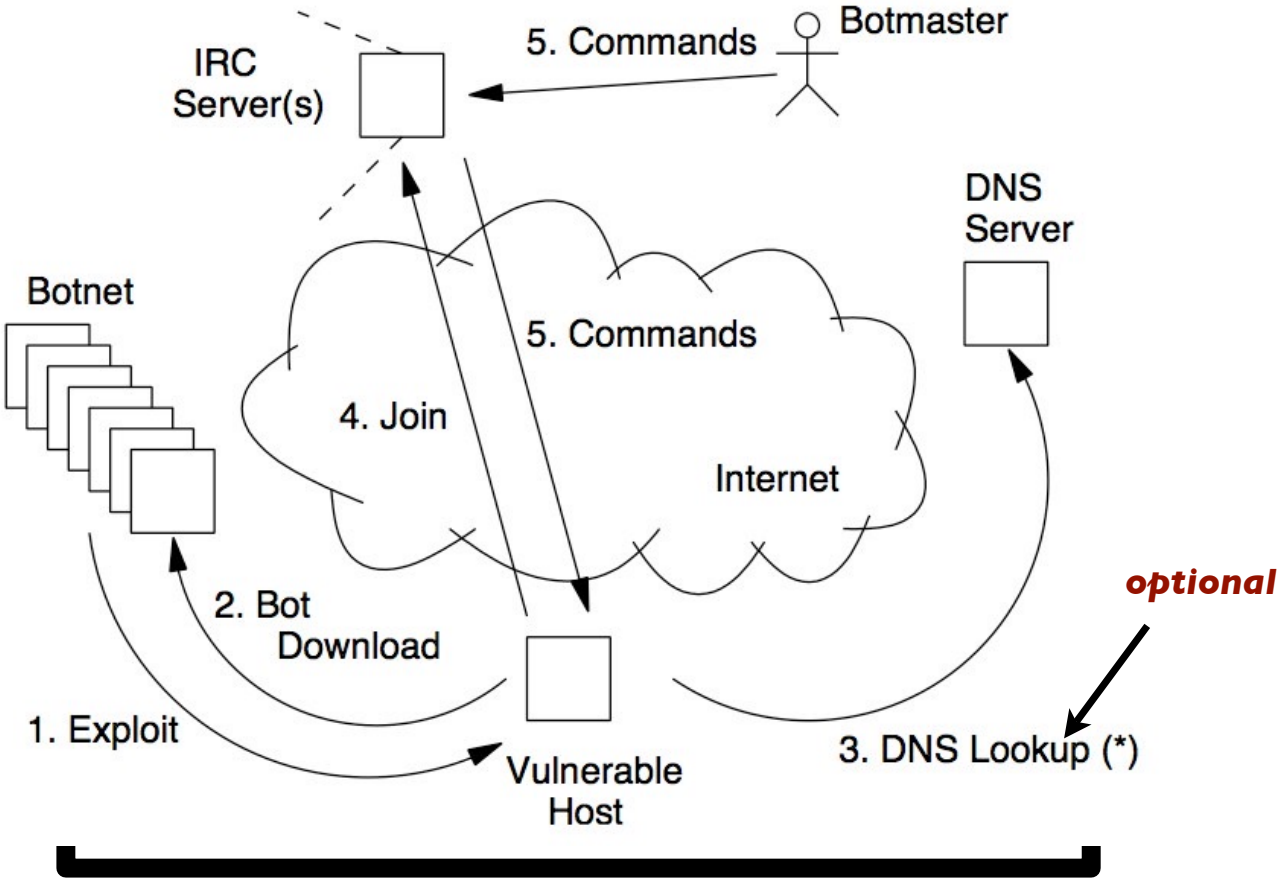
- An army of compromised hosts ("bots") coordinated via a command and control center (C&C).The perpetrator is usually called a "botmaster".



"A botnet is comparable to compulsory military service for windows boxes"

-- Bjorn Stromberg

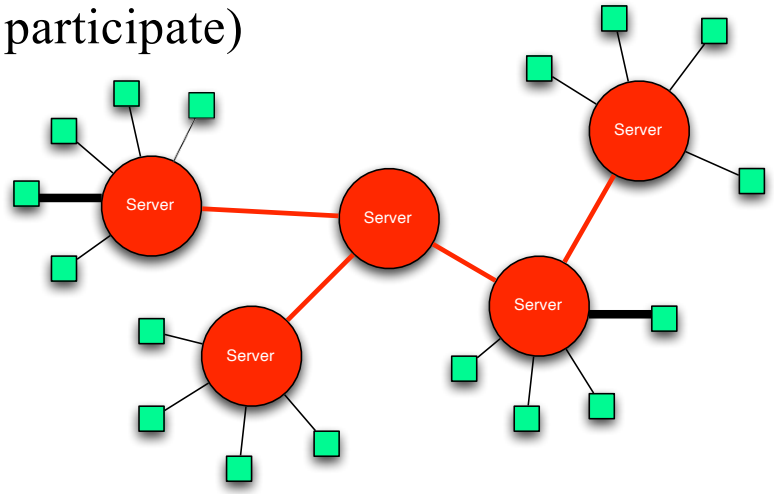
Typical (IRC) infection cycle



Bots usually require some form of authentication from their botmaster

IRC

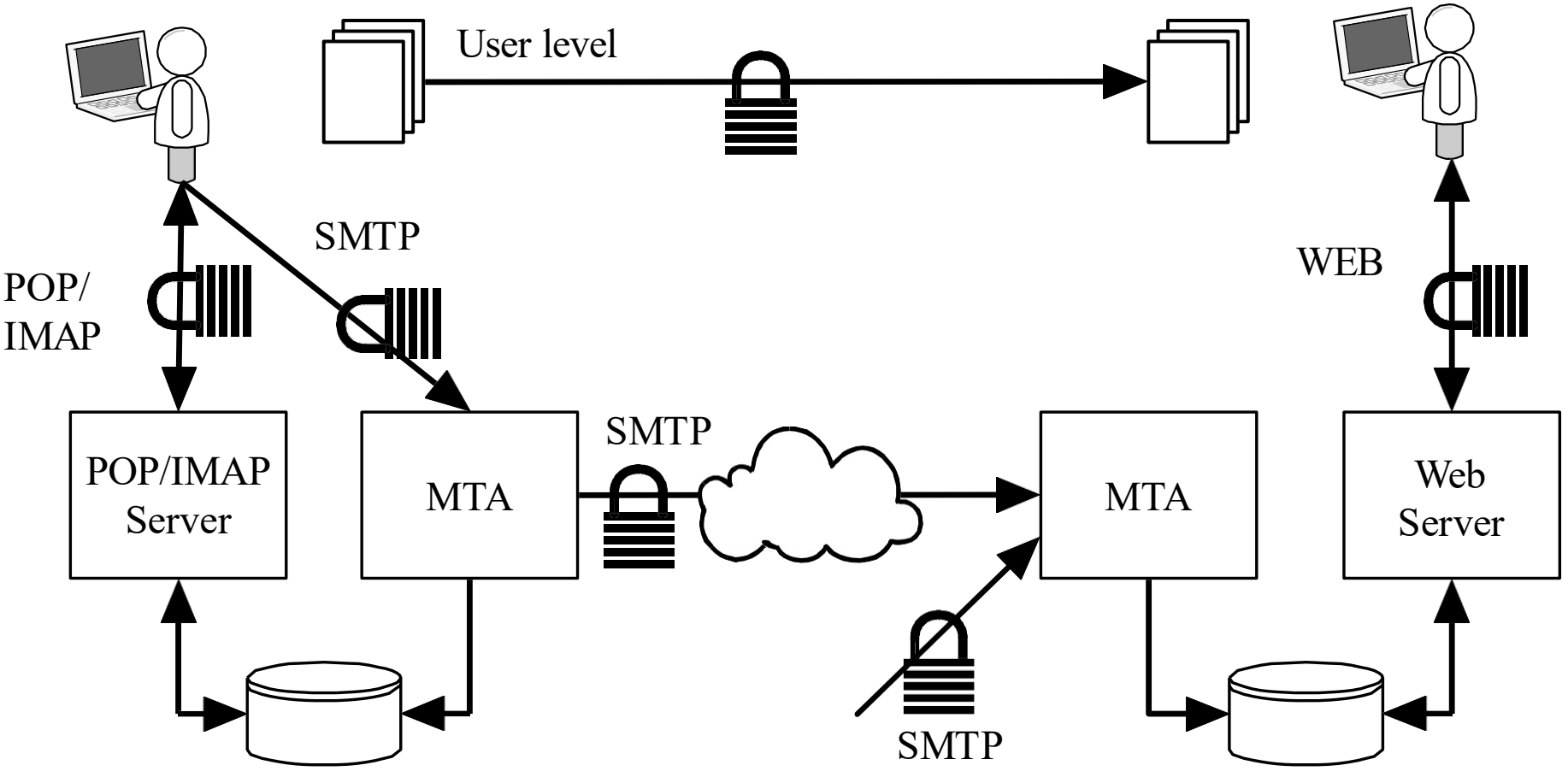
- 1988 - one-to-many or many-to-many chat (for BBS)
- Client/server -- TCP Port 6667
- Used to report on 1991 Soviet coup attempt
- Channels (sometimes password protected) are used to communicate between parties.
  - Invisible mode (no list, not known)
  - Invite only (must be invited to participate)

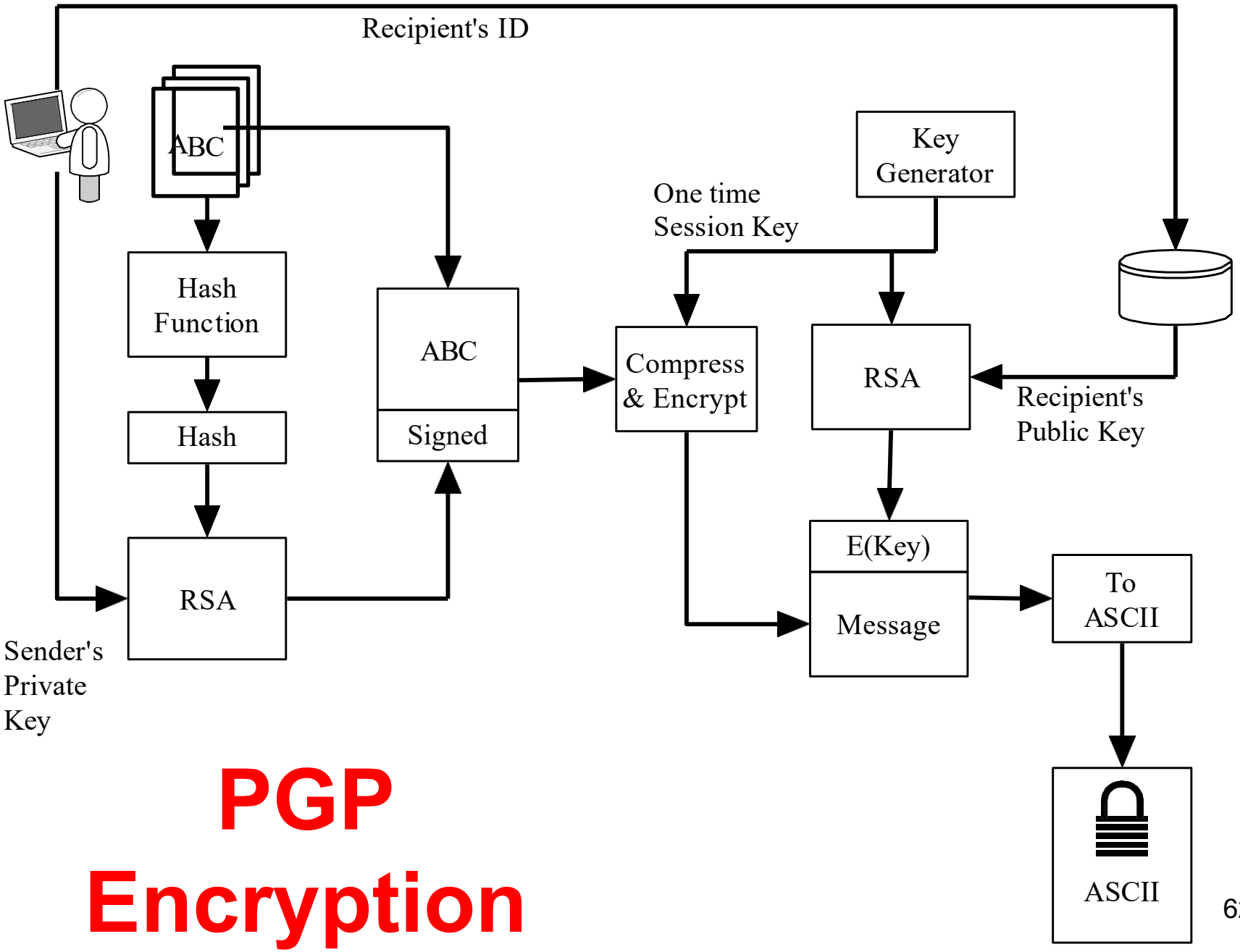


General Email Countermeasures

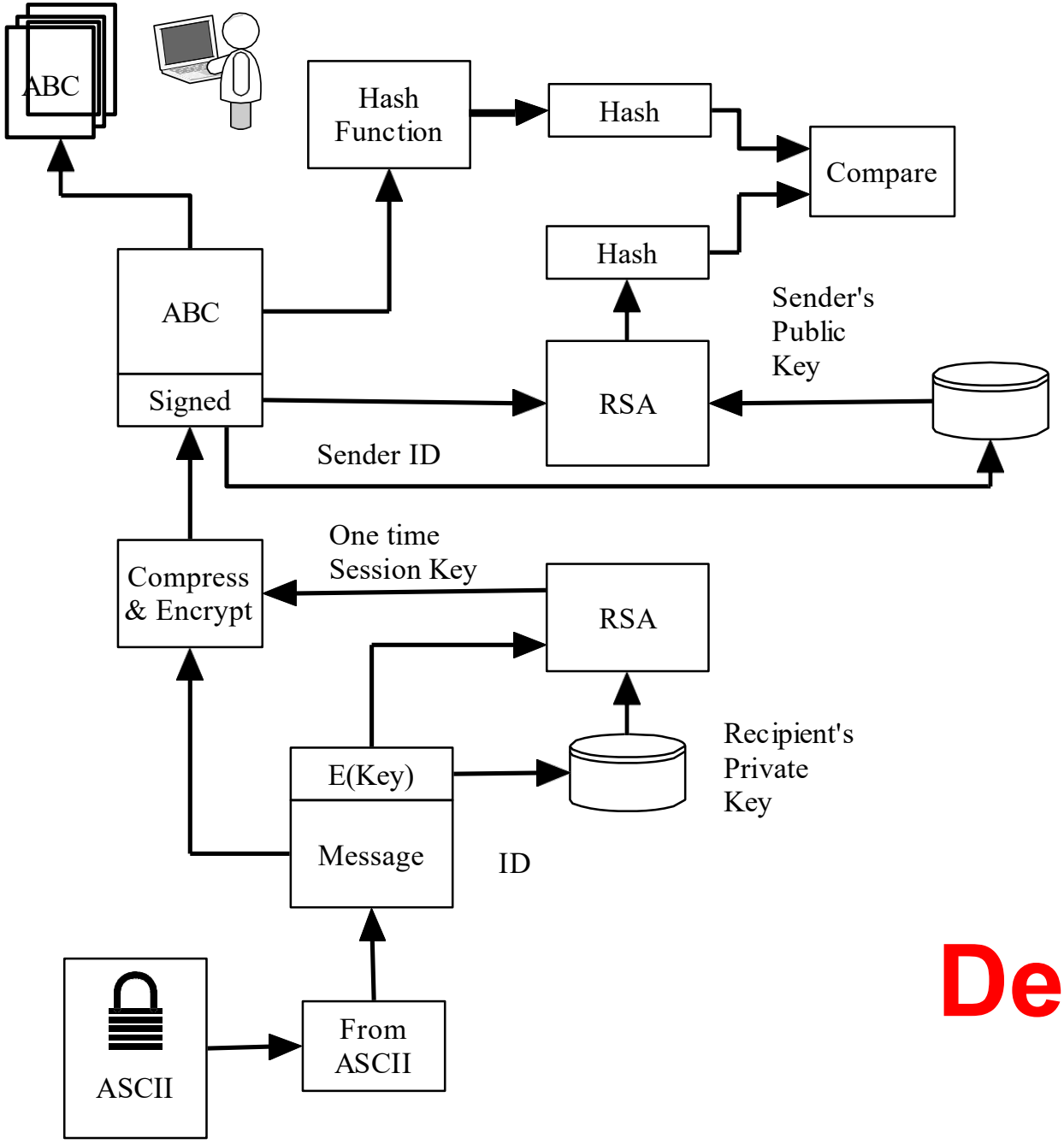
- Encryption & authentication
- Email filtering
- Content Filtering
- Email Forensics

Encryption & Authentication





PGP Encryption

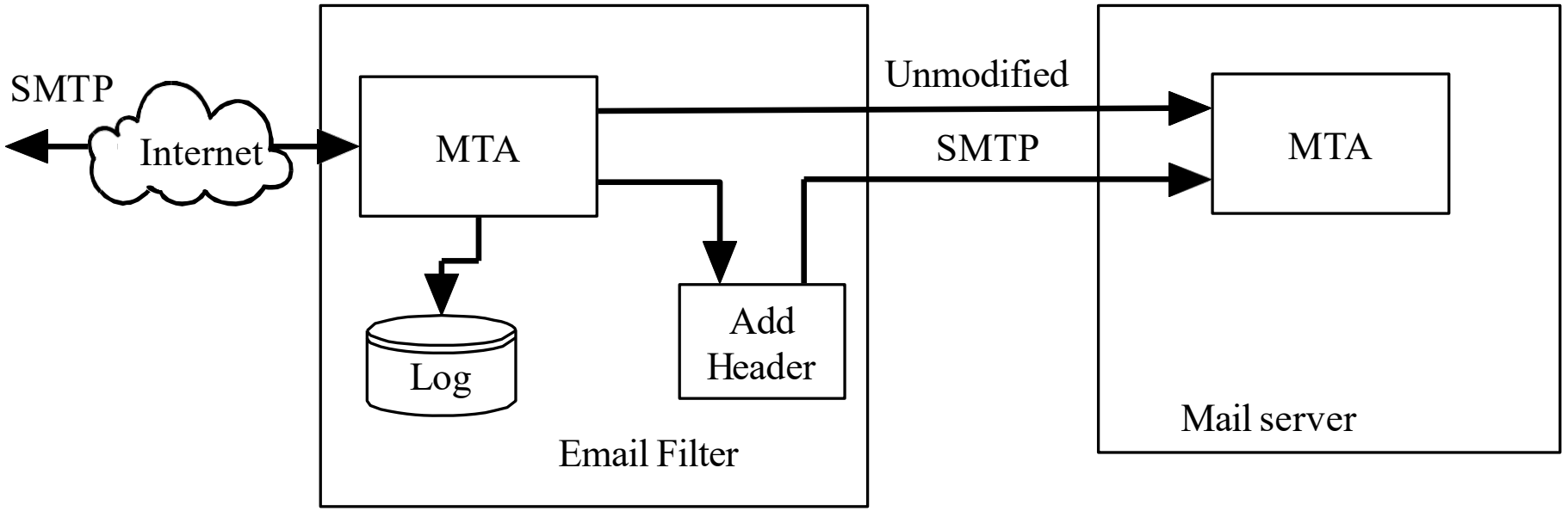


PGP Decryption

Email Filtering

- Check email
  - Based on email addresses
  - Based on domain address
  - Based on malicious payload
- Either Block, pass, or modify the email

Email Filtering



Spam Filter

- Uses learning to decide what content is spam.
- System is “trained” to know is spam
- Spam filter will mark the message as spam.
- Some User agents support spam detection and will move spam email into a spam folder

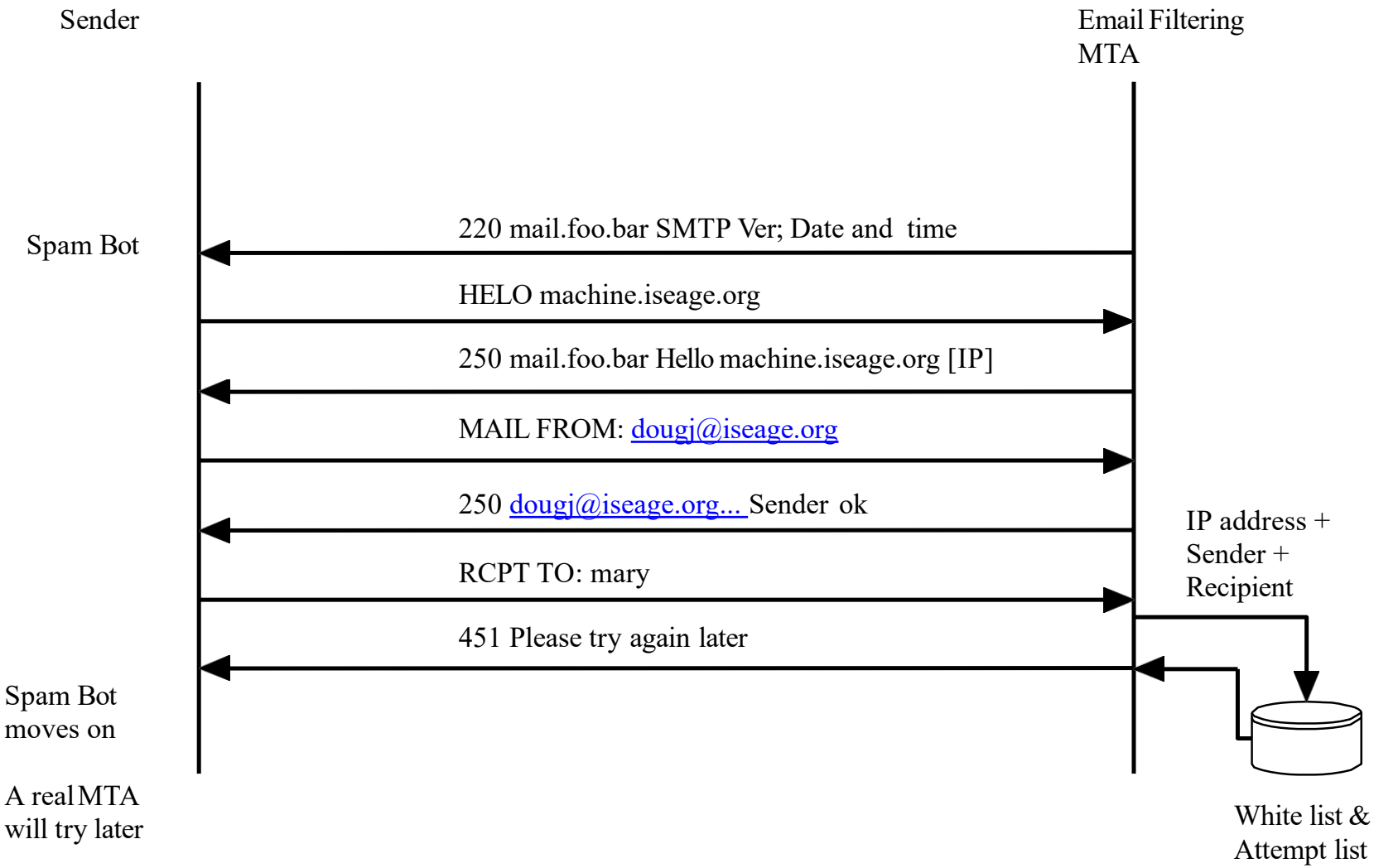
Bypassing a Spam Filter

- Misspelled keywords
- Picture only
- Picture with background words

Filtering list

- Blacklist
  - A list of bad users & domains
  - Spammers just change domains
- Whitelist
  - A list of good users and domains
  - Very restrictive

Greylist


Sender
Email Filtering MTA
Spam Bot
220 mail.foo.bar SMTP Ver; Date and time
HELO machine.iseage.org
250 mail.foo.bar Hello machine.iseage.org [IP]
MAIL FROM: dougj@iseage.org
250 dougj@iseage.org... Sender ok
RCPT TO: mary
451 Please try again later
IP address + Sender + Recipient
Spam Bot moves on
A real MTA will try later
White list & Attempt list

Content filter

- Examine the payload for:
  - Viruses
  - Worms
  - Trojan horses
- Often based on a signature
- Requires constant update of signatures

Outbound content filtering

- Used to keep private information from leaving
  - SS Numbers
  - Account Numbers
  - Medical records
- Will either log, stop, or encrypt violating emails

Bypassing a content filter

- Encryption
  - There are encrypted viruses
- Compression

Email Forensics

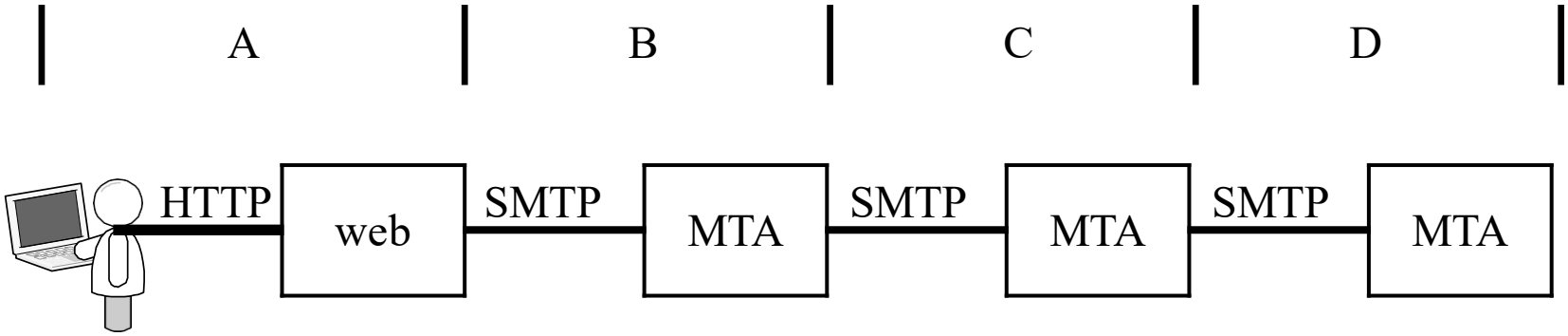
D
Received: from nf-out-0910.email.mta (nf-out-0910.email.mta [192.168.182.188])
by vulcan.ece.mail.spam (8.12.8/8.9.3) with ESMTP id k85FaxBT1486661
for <john@ee.mail.spam>; Tue, 5 Sep 2006 10:36:59 -0500 (CDT)

C
Received: by nf-out-0910.email.mta with SMTP id p77so1381355nfc
for <john@ee.mail.spam>; Tue, 05 Sep 2006 08:34:32 -0700 (PDT)
DomainKey-Signature: a=rsa-sha1; q=dns; c=nofws;
s=beta; d=spammer.fake;
h=received:message-id:date:from:to:subject:mime-version:content-type;
b=BD9tHbNaozYZj9gNQqXmkrnHNA3N8+3W4NApcFJkKsKyX8DdOTS7Dp1VNunGx66SLcU5r
YiDxCnY6SuVCktWq73DDH7MYEfWgaOtYdl/hILBIRVNcbLxGtyCoIT7I8use4F4RgCzZWc3
Oc6fjqNzgGLe5s3RFQ9eVPhS+HxW+DA=

B
Received: by 10.65.237.1 with SMTP id o1mr4809264qbr;
Tue, 05 Sep 2006 08:34:31 -0700 (PDT)

A
Received: by 10.70.63.2 with **HTTP**; Tue, 5 Sep 2006 08:34:31 -0700 (PDT)
Message-ID:
<ab156e9f0609050834v528b5b2eld9204458fe6409a1@mail.spammer.fake>
Date: Tue, 5 Sep 2006 10:34:31 -0500
From: "Harry Mudd" <Harry6502@spammer.fake>
To: john@ee.mail.spam
Subject: mail trace 2
MIME-Version: 1.0

Email Forensics



10.70.63.2
9/5/2006
8:34:31 PDT

10.65.237.1
9/5/2006
8:34:31 PDT

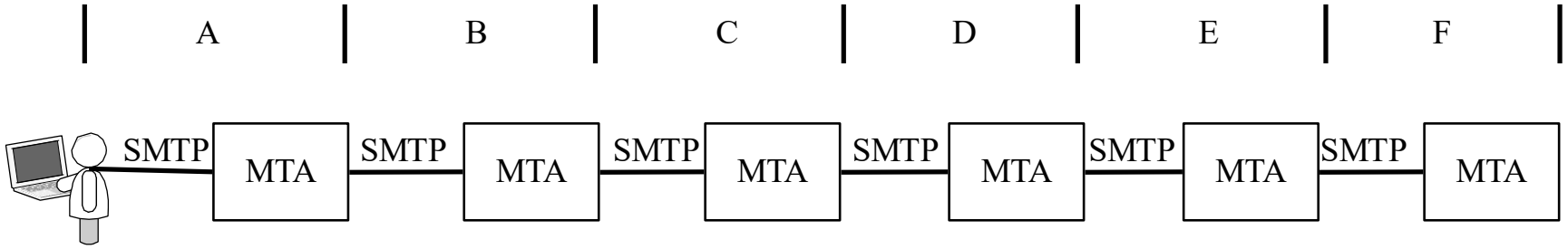
nf-out-0910.email.mta
9/5/2006
8:34:32 PDT

vulcan.ece.mail.spam
9/5/2006
10:36:59 CDT
From: 192.168.182.188

Email Forensics

F
```
Received: from pop-5.mail.spam (pop-5.mail.spam [172.16.7.12])
        by vulcan.ece.mail.spam (8.12.8/8.9.3) with ESMTP id
k85FjSBT1508024
        for <john@EE.MAIL.SPAM>; Tue, 5 Sep 2006 10:45:28   -0500 (CDT)
```
E
```
Received: from devirus-2.mail.spam (devirus-2.mail.spam [172.16.7.10])
        by pop-5.mail.spam (8.12.11.20060614/8.12.11) with SMTP id
k85Fgt28016542
        for <john@mail.spam>; Tue, 5 Sep 2006 10:42:55   -0500
```
D
```
Received: from (despam-3.mail.spam [172.16.7.5]) by devirus-2.mail.spam
with smtp
          id 0df9_ae8af2c2_3cca_11db_969a_001372537fef;
        Tue, 05 Sep 2006 10:38:34 +0000
```
C
```
Received: from magellan.sender.mta (magellan.sender.mta
[192.168.16.211])
        by despam-3.mail.spam (8.12.11.20060614/8.12.4) with ESMTP id
k85FgttT020053
        for <john@mail.spam>; Tue, 5 Sep 2006 10:42:55   -0500
```
B
```
Received: from vulcan.ece.mail.spam (vulcan.ece.mail.spam [ 172.20.5.6])
        by magellan.sender.mta (8.13.6/8.13.6) with ESMTP id
k85Fgemo030599
        for <dwj@sender.mta>; Tue, 5 Sep 2006 10:42:40   -0500 (CDT)
        (envelope-from john@mail.spam)
```
A
```
Received: from [172.21.4.7] (babylon4.ece.mail.spam [172.21.4.7])
        by vulcan.ece.mail.spam (8.12.8/8.9.3) with ESMTP id
k85Fj6BT1501144
        for <dwj@sender.mta>; Tue, 5 Sep 2006 10:45:06   -0500 (CDT)
Message-ID: <44FD9AEC.4040103@mail.spam>
Date: Tue, 05 Sep 2006 10:42:36  -0500
From: Harry Mudd <Harry@mail.spam>
Organization: ISU Information Assurance Center
User-Agent: Mozilla Thunderbird 1.0.7 (Windows/20050923)
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: Dave Johnson <dwj@sender.mta>
Subject: test 4
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit
```
Spam Filters
```
X-Filter-MailScanner-Information: Please contact the ISP for more
information
X-Filter-MailScanner: Found to be clean
X-Filter-MailScanner-SpamCheck: not spam, SpamAssassin (score=-2.6,
        required 6, autolearn=not spam, BAYES_00   -2.60, SPF_PASS  -0.00)
X-Filter-MailScanner-From: john@mail.spam
X-PMX-Version: 5.2.0.264296, Antispam-Engine: 2.4.0.264935, Antispam-
Data: 2006.9.5.82442
X-Perlmx-Spam: Gauge=IIIIIII, Probability=7%, Report='__C230066_ P5 0,
__CP_URI_IN_BODY 0, __CT 0, __CTE 0, __CT_TEXT_PLAIN 0, __HAS_MSGID 0,
__MIME_TEXT_ONLY 0, __MIME_VERSION 0, __SANE_MSGID 0, __USER_AGENT 0'
```

Email Forensics



vulcan.ece.mail.spam
9/5/2006
10:45:06 CDT
from 172.21.4.7

despam-3.mail.spam
9/5/2006
10:42:55 CDT
from: 192.168.16.211

pop-5.mail.spam
9/5/2006
10:42:55 CDT
from: 172.16.7.10

babylon4.ece.mail.spam

magellan.sender.mta
9/5/2006
10:42:40 CDT

devirus-2.mail.spam
9/5/2006
10:38:34 CDT
from: 172.16.7.5

vulcan.ece.mail.spam
9/5/2006
10:45:28 CDT
From: 192.168.182.188

Email Forensics

(Removed local headers)
Received: from ns09.egujarat.net (202-149-46-162.static.exatt.net [202.149.46.162] (may be forged))
D by despam-2.iastate.edu (8.12.11.20060614/8.12.4) with ESMTP id k89KIRCr017274
for <dougj@iastate.edu>; Sat, 9 Sep 2006 15:18:28 -0500
Received: from ns09.egujarat.net (localhost.localdomain [127.0.0.1])
C by ns09.egujarat.net (8.13.5/8.13.5) with ESMTP id k89H5sYI007263
for <dougj@iastate.edu>; Sat, 9 Sep 2006 22:37:19 +0530
Received: (from administrator@localhost)
B by ns09.egujarat.net (8.13.5/8.13.5/Submit) id k89Gxf4q006335;
Sat, 9 Sep 2006 22:29:41 +0530
Date: Sat, 9 Sep 2006 22:29:41 +0530
A Message-Id: <200609091659.k89Gxf4q006335@ns09.egujarat.net>
To: dougj@iastate.edu
Subject: Password change required!
From: "eBay Inc." <admin@eBay.com>
Content-Type: text/html
Spam Filter 2 X-egujarat-MailScanner-Information: Please contact the ISP for more information
X-egujarat-MailScanner: Found to be clean
X-MailScanner-From: administrator@ns09.egujarat.net
Spam Filter 1 X-PMX-Version: 5.2.0.264296, Antispam-Engine: 2.4.0.264935, Antispam-Data: 2006.9.9.124943
X-Perlmx-Spam: Gauge=XXXXXXXXXIIIIIIIII, Probability=99%,

<img src="http://pics.ebaystatic.com/aw/pics/navbar/eBayLogoTM.gif "
width="150" height="70">

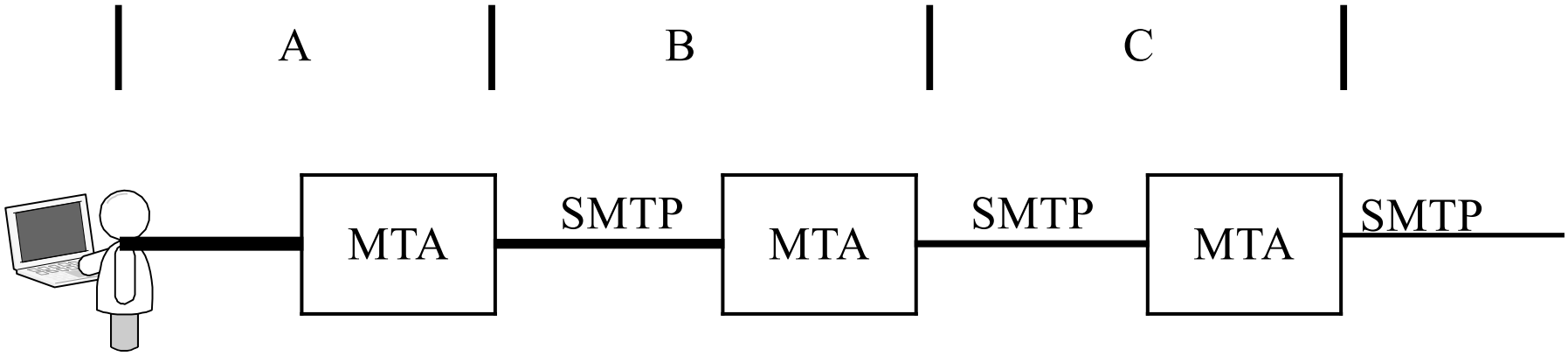
Logo Dear sir,

We recently have determined that different computers have logged onto your eBay account, and multiple password failures were present before the logons. We strongly advice CHANGE YOUR PASSWORD.

If this is not completed by <STRONG>September 15, 2006</STRONG>, we will be forced to suspend your account indefinitely, as it may have been used for fraudulent purposes. Thank you for your cooperation.

Phishing Site <A
href="http://linux.netzero.idv.tw/~ming/.change/index.php?MfcISAPICommand=ChangeFPP"
target=_blank>Click here to Change Your Password</A></TD>

Email Forensics



Logged into MTA	ns09.egujarat.net 9/9/2006 22:29:41	ns09.egujarat.net 9/9/2006 22:29:41 from 127.0.0.1	despam-2.mail.spam 9/9/2006 15:18:28 CDT from: 202.149.46.162