

Malware and botnets

.

Malware

- **malicious** and **software**
- software designed to infiltrate or damage a computer system without the owner's informed consent
 - Computer virus
 - Worm
 - Trojan horse
 - Rootkit, backdoor
 - Spyware
 - Keylogger
 - Adware
 - Zombie,bot
 - etc.

Virus – why the name?

A virus is:

- Not a full program, “cannot live alone”
- It reproduces itself, spreads. (“Infection”)
- Some transfer media, user interaction might be needed
- Makes nasty things (or not)

Most so-called viruses is not a virus by this definition.

Worm: Can reproduce automatically (no user needed)

Trojan Horse: The user thinks that the code is o.k., but it isn't.

Rootkit, backdoor: remains on the computer and hard to find

Bot: participates in a distributed network for malicious activity.

Virus classification

- boot sector
- file infector
- macro virus
- encrypted virus
- stealth virus
- polymorphic virus (modifications to avoid identification: encryption, inserting dummy code)
- metamorphic virus (the same, but not inserting dummy code, instead, code-that-does-the-same)

Old-school file virus

- Modifies executable files, appends own code into them (how? - .com:simple .exe:a bit more difficult)
- Whenever the executable is loaded, the virus is started
- Might instantly check for one-more-all other executables (only .com/only .exe/all) and infect them
- Or just load into the memory, stay resident (TSR), and infect whenever we execute any program
- Might modify or “encrypt” itself at every infection – some “decryption” part is still the same (opportunity to recognize the virus)
 - 20 byte is enough for a basic encryption scheme
 - Tricky modifications can be done by the virus (e.g. `xor ax,ax ; mov ax,0 ; sub ax,ax` are the same)

Other viruses („virii”)

- Boot virus: infects boot sector (if you leave a floppy in the drive, it loads the code, and then,...)
- Macro virus: Word/Excel macros affected
- Hardware level destroying virus: E.g. CIH bios clearing, or cd-rom firmware bugs
- BIOS virus: Code is stored in the BIOS – disinfection might be hard (one of the latest tricks, difficult)
- Encrypting (ransom) malware: Encrypts all the files, decryption only when you pay.

Goals of the virus-writers

- Old times: just to show it is possible to write such code (first virus ~1982)
- Be famous (or to collect- „vixers”)
- For fun
- Do harmful activities
- To write a better virus: harder to identify, harder to disinfect, faster spreading,
- Earn money (→ spam, fake virus scanners, phishing, password and credit card no. collection, ransom (by encryption), fake/rogue security software, etc.

Potyogós virus - cascade

- Back from 1987 – the starting time of the new era
- 1071 byte
- First virus that caused mass infection in Hungary
- Encrypts itself in some form (no, not AES, nor RSA)
- Nasty code: after some time, characters started to fall off the screen
- TSR code
- <http://www.youtube.com/watch?v=UWLg6tTeQRg>
- Also check: <http://kannan.jumbledthoughts.com/index.php/21-virus-and-other-malware-payload-videos/>

Potyogós – in action

```

COUNTRY.S S      COUNTRY.TXT      DEBUG.EXE      EDIT.COM      EXPAND.
FDISK.EXEY      FORMAT.OM      KEYB.COM      KEYBOARD.SYS  MEM.EXEEXE
NETWORKS.X      NLSFUNCC XE   OS2.TXT      QBASIC.EXE    README.T
SCANDISK.X      SYS.COM.E     XCOPY.EXE    CHOICE.C M    DEFRAG.EXT
DEFRAG.H T      DELOLDOS.E E  DOSHELP.HLP  EGA.CPI O     EGA2.CPIXE
EGA3.CPI E T    EMM386.EXE    KEYBRD2.YS   MSCDEX.E E    SCANDISK.INI
ANSI.SYSLP E    APPEND.E E    CHKSTATESSYS DBLWIN.H      DELTREE.EXE
DISKCOMP.O      DISKCO        M    DISPLAY.Y     DOSKEY.X      DRUSPACE EX
DRUSPACE.CL     DRUSPAPYX F   DRUSPACE S   MSD.EXECLP    REPL CE..XEE
  STORE.H      HELP.HCE.C    DRIVER.SS S  EDIT.HLPOM    FAST ELPE X
  STOPENEXE    FC.EXELP X    FIND.EXE.SYS GRAPHICS COM   GR P I S
  LP.OM.EX     HIMEM.SY.ID   INTERLNKYE E I TER UR.XE   L . X
READF X C M     E MAKERS NE   MEMMAKER     M MMA ER N    M C M
FA OU B OM      E.COM.E      MOVE E H     OO L          P . X
HE C 3          DR UE.S S    SE E E      E            S E
LO I L 6P      R N.E E      M H
MON M X        O .C M      F X
QBASIC.        U B          O 6
SMARTDR. 1 ( M      X4,300      .            .            A H C .
TREE.CO. M M      Y9 0 4     TVER .      N S          ABEL E .
COMMANDH     ROR X       ARTMXEX     E K .        ODE.O E
C:\DOS>U 8    SAM I T O    INTD.N.     MST LS..     OWER E E
C:\DOS>M.P E  UMA TMAC.M  S NFIGO38 L SHAR .EXDE   IZER.EXEE
C:\DOS>.CEME  ANFORME3,01 Ubytes.UMBLP SORT.EXEEI   UBST.EXEPRO
C:\DOS>930fi e s)UTOEX30,84 , 2 Cbytes.freeP PRINT.EXEL F UNDELETE.EXE

```


Part of disassembled virus “polimer”

```
polimer                proc        far

start::
    jmp                loc_4
    db                 00h, 3Fh
    db                 7 dup (3Fh)
    db                 43h, 4Fh, 4Dh, 00h, 02h, 00h
    db                 40h, 00h, 8Dh, 36h, 80h, 00h
    db                 03h, 00h
    db                 14 dup (0)
data_59                db                 'A legjobb kazetta a POLIMER kaze'
                    db                 'tta ! Vegye ezt! ', 0Ah, 0Dh
                    db                 '$'
                    db                 'ERROR', 0Ah, 0Dh, '$'
data_60                dw                 5
data_61                dw                 147Dh
loc_1::
    mov                si,data_46e
    mov                di,data_49e
    mov                cx,30h
    cld
    rep                movsb                ; Clear direction
    jmp                $-0BAh                ; Rep when cx >0 Mov [si] to es:[di]

loc_2::
    jmp                loc_10

loc_3::
    jmp                loc_9

loc_4::
    mov                al,0
    mov                ah,0Eh
    int                21h                ; DOS Services ah=function 0Eh
    ; set default drive dl (0=a:)

    mov                dx,data_36e
    mov                ah,1Ah
    int                21h                ; DOS Services ah=function 1Ah
    ; set DTA(disk xfer area) ds:dx
```

Sample polymorphic code – the basis

Start:

GOTO Decryption_Code

Encrypted:

...

lots of encrypted code

...

Decryption_Code:

A = Encrypted

Loop:

B = *A

B = B XOR CryptoKey

*A = B

A = A + 1

GOTO Loop IF NOT A = Decryption_Code

GOTO Encrypted

CryptoKey:

some_random_number

From wikipedia

The polymorphic equivalent

Start:

GOTO Decryption_Code

Encrypted:

...

lots of encrypted code

...

Decryption_Code:

C = C + 1

A = Encrypted

Loop:

B = *A

C = 3214 * A

B = B XOR CryptoKey

*A = B

C = 1

C = A + B

A = A + 1

GOTO Loop IF NOT A = Decryption_Code

C = C^2

GOTO Encrypted

CryptoKey:

some_random_number

Macro virus

- became very common in mid-1990s since
 - platform independent
 - infect documents
 - easily spread
- exploit macro capability of office apps
 - executable program embedded in office doc
 - often a form of Basic
- more recent releases include protection
- recognized by many anti-virus programs

Rogue security software -wiki

Partial list of rogue security software

The following is a partial list of rogue security software, most of which can be grouped into families. These are functionally-identical versions of the same program repackaged as successive new products by the same vendor.^{[1][12]}

- Advanced Cleaner^[18]
- AlfaCleaner^[19]
- AntiSpy Check 2.1^[20]
- AntiSpy Storm^[21]
- AntiSpyware 2009^[22]
- AntiSpyware Expert^[23]
- AntiSpywareMaster^[24]
- AntiSpyware Suite^[25]
- AntiSpyware Shield^[26]
- Antivermins^[27]
- Antivirgear^[28]
- Antivirus 2008^[29]
- Antivirus 2009^[30]
- Antivirus 2010 (also known as Anti-virus-1)^{[31][32]}
- Antivirus 360^[33]
- Antivirus Pro 2009^[34]
- AntiVirus Gold^[35]
- Antivirus Master^[36]
- Antivirus XP 2008^[37]
- Avatod Antispyware 8.0^[38]
- Awola^[39]
- Brave Sentry^[40]
- BestsellerAntivirus^[41]
- Cleanator^[42]
- ContraVirus^[43]
- Doctor Antivirus^[44]
- Doctor Antivirus 2008^[45]
- Drive Cleaner^[46]
- Easy Spyware Cleaner^[47]
- Erorsafe^[48]
- GreenAV2009^[49]
- IE Antivirus (aka IE Antivirus 3.2)^[50]
- IEDefender^[51]
- Infe Stop^[52]
- Internet Antivirus (aka Internet Antivirus Pro, distributed by plus4scan.com)^[53]
- KVMSecure^[54]
- Mac Sweeper^[55]
- Malware Crush^[56]
- Malware Core^[57]
- Malware Alarm^[58]
- Malware Bell (a.k.a. Malware Bell 3.2)^[59]
- Malware Defender (not to be confused with the HIPS firewall of the same name)^[60]
- MS Antivirus^[61]
- MS AntiSpyware 2009^[62]
- MaxAntiSpy^[63]
- Netcom3 Cleaner^[64]
- PC Secure System^[65]
- PC Antispy^[66]
- PC Clean Pro^[67]
- PC Privacy Cleaner^[68]
- PC SpeedScan Pro (distributed by FinallyFast.com, Rogueness is questionable)
- PestTrap^[69]
- Perfect Cleaner^[70]
- Perfect Defender 2009^[71]
- PersonalAntiSpy Free^[72]
- PAL Spyware Remover^[73]
- PC Privacy Tools^[74]
- PC Antispyware^[75]
- PSGuard^[76]
- Rapid AntiVirus^[77]
- Real AntiVirus^[78]
- Registry Great^[79]
- Safety Alert 2008^[80]
- SaliarAR^[81]
- Secure PC Cleaner^[82]
- Security Toolbar 7.1^[83]
- Smart Antivirus 2009^[84]
- SpyAxe^[85]
- Spy Away^[86]
- Spy Crush^[87]
- Spydawn^[88]
- Spy Guarder^[89]
- Spy Heal (a.k.a. SpyHeals & \VirusHeal)^[90]
- SpyMarshal^[91]
- Spylocked^[92]
- Spy Sheriff^[93]
- Spy Spotter^[94]
- SpywareBot (Spybot - Search & Destroy knockoff)^[95]
- Spyware Cleaner^[96]
- Spyware Guard 2008^[97]
- Spyware Protect 2009^[98]
- Spyware Quake^[99]
- Spyware Sheriff (often confused with Spy Sheriff)^[100]
- Spyware Stormer^[101]
- Spyware Striker Pro (distributed by FinallyFast.com)^[102]
- Spyware Protect 2009^[103]
- Super Ad Blocker
- Spyware Strike^[104]
- SpyRid^[105]
- SpyWiper^[106]
- System Antivirus 2008^[107]
- System Live Protect^[108]
- System Doctor^[109]
- System Security^[110]
- Total Secure 2009^[111]
- TrustedAntivirus^[112]
- TheSpyBot (Spybot - Search & Destroy knockoff)^[113]
- Ultimate Cleaner^[114]
- \VirusHeal^[115]
- \Virus Isolator^[116]
- \Virus Locker^[117]
- \Virus Protect Pro^[118]
- \Virus Remover 2008^[119]
- \Virus Remover 2009^[120]
- \VirusMelt^[121]
- \VirusRanger^[122]
- \Virus Response Lab 2009^[123]
- \VirusTrigger^[124]
- \Vista Antivirus 2008^[125]
- \WinAntiVirus Pro 2008^[126]
- \WinDefender (not to be confused with the legitimate Windows Defender)^[128]
- \WinFixer^[129]
- \WinHound^[129]
- \WinSpywareProtect^[130]
- \WinWab Security 2008^[131]
- \WorldAntiSpy^[132]
- XP Antivirus^[133]
- XP AntiSpyware 2009^[134]
- \XP_Shield^[135]

I guess You expected a shorter list,...

Limits of the malware

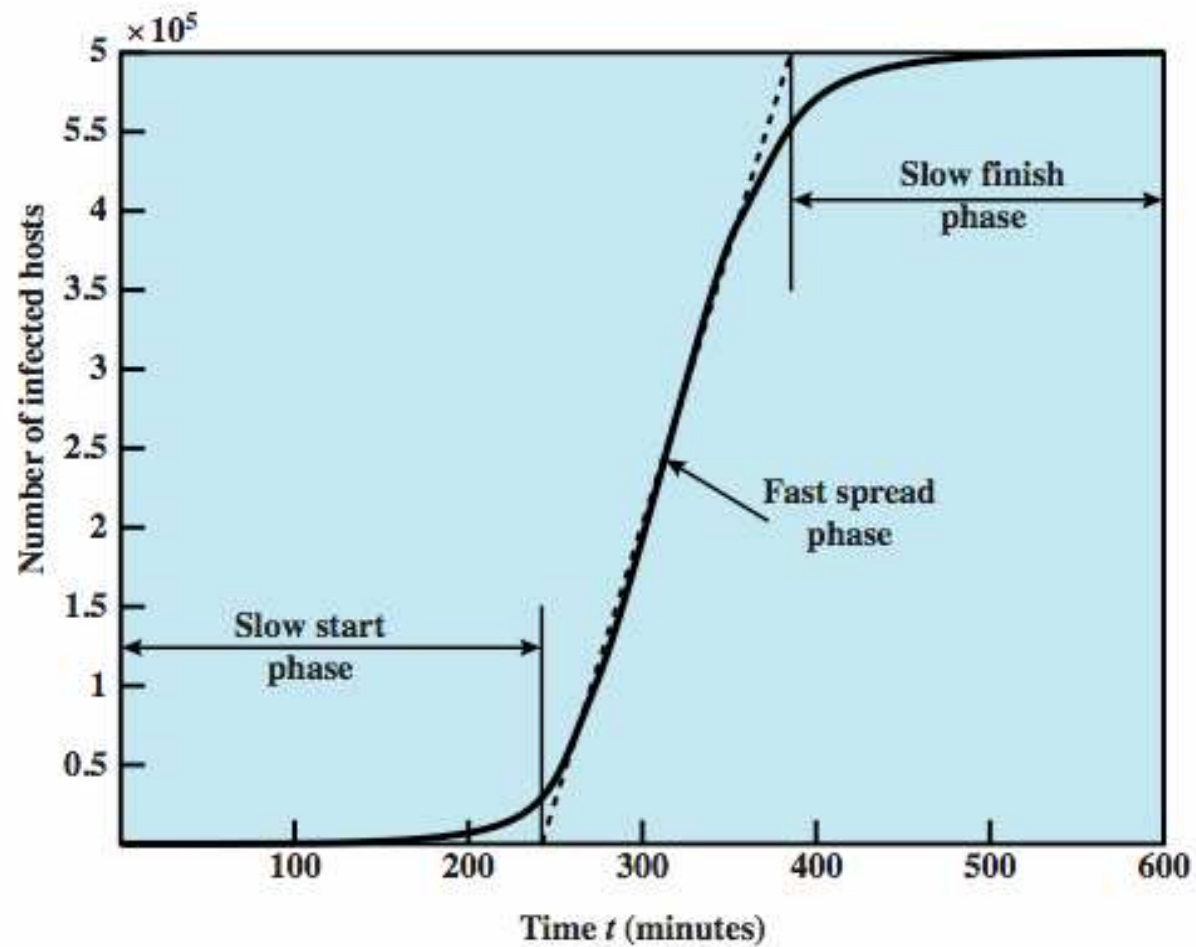
- A malware can fully control a computer
- Read memory, files
- Record keyboard, mouse, monitor activity
- Use webcam, microphone of the computer
- Find all archived information (emails, stored passwords, email, web history, stored files, etc.)
- A malware can hide itself very efficiently, currently it is almost always identifiable, but later...?

- Security schemes with additional hardware needed (smart card, token, OTP generator –with/without challenge) – remember: the computer is still controlled by the attacker
- No easy solution on untrusted terminal problem
- Therefore it is essential to avoid malware infections
- Of course, in practice, malwares are not perfect, but: expect the worst case scenario.

Worms

- replicating program that propagates over net
 - using email, remote exec, remote login
- has phases like a virus:
 - dormant, propagation, triggering, execution
 - propagation phase: searches for other systems, connects to it, copies self to it and runs
- may disguise itself as a system process
- concept seen in Brunner's "Shockwave Rider"
- implemented by Xerox Palo Alto labs in 1980's

Worm propagation model



Famous Worm Attacks

- Code Red
 - July 2001 exploiting MS IIS bug
 - probes random IP address, does DDoS attack
 - consumes significant net capacity when active
- Code Red II variant includes backdoor
- SQL Slammer
 - early 2003, attacks MS SQL Server
 - compact and very rapid spread
- Mydoom
 - mass-mailing e-mail worm that appeared in 2004
 - installed remote access backdoor in infected systems
- Nowtimes: One after the other, hard to keep-up with new worms/botnets

Identification of malware

- Based on signatures (hard to make for polymorphic or metamorphic code)
 - In files (virus)
 - In network traffic (worms, email viruses)
 - In memory (infected hosts, e.g. botnet)
 - Highly optimized (thousands of signatures should be detected)
- Based on behavior (anomaly detection, checking code (e.g. for unpacking), heuristic algorithms – scoring)

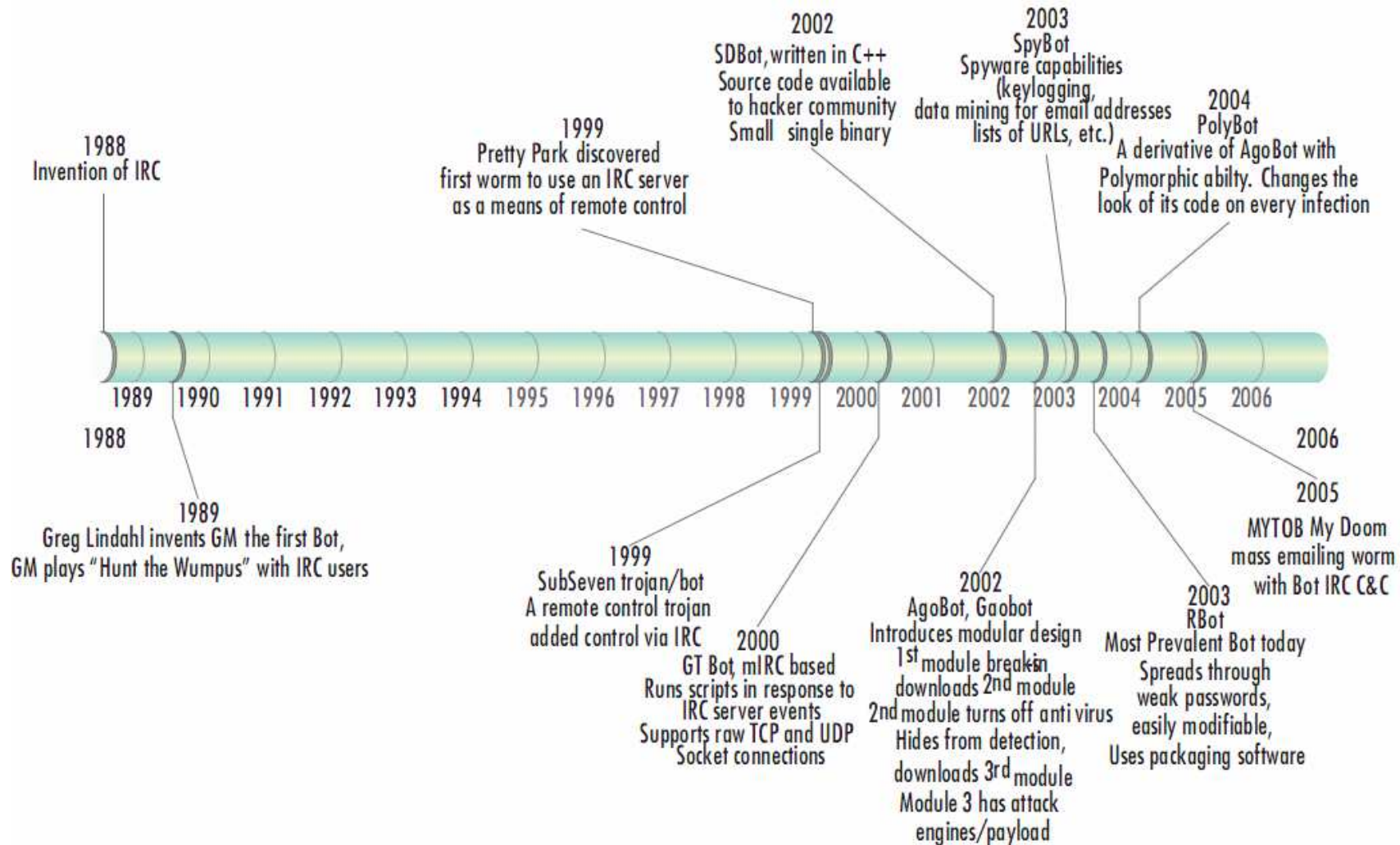
Removal of malware

- First step: terminate running malware
 - As the malware might stop the removal tool
 - The malware might detect our plans and do bad things (e.g. delete files) – unusual
 - Some malware run in multiple tasks to avoid stopping
 - Some malware are specially designed to download more malware – all should be removed
- The files of the malware should be identified
 - Based on signatures
 - Check auto-start applications
 - Can be deep in the OS (modified kernel, modified BIOS)
 - For traditional viruses: the code is injected into a binary executable
- Remove the malware
 - Nowtimes, generally a simple file deletion is enough
 - In traditional virus, the code should be extracted from the host software: hard task, virus „killers” exist, but not for all virus
 - Backdoors, or re-infection trick made by the malware should also be cleared (not very common)
 - The vulnerability should also be handled to avoid re-infection
 - Some “junk” might remain
 - including text files with collected passwords!
 - Most malware has a mechanism to avoid multiple infections – might be a trick to protect hosts

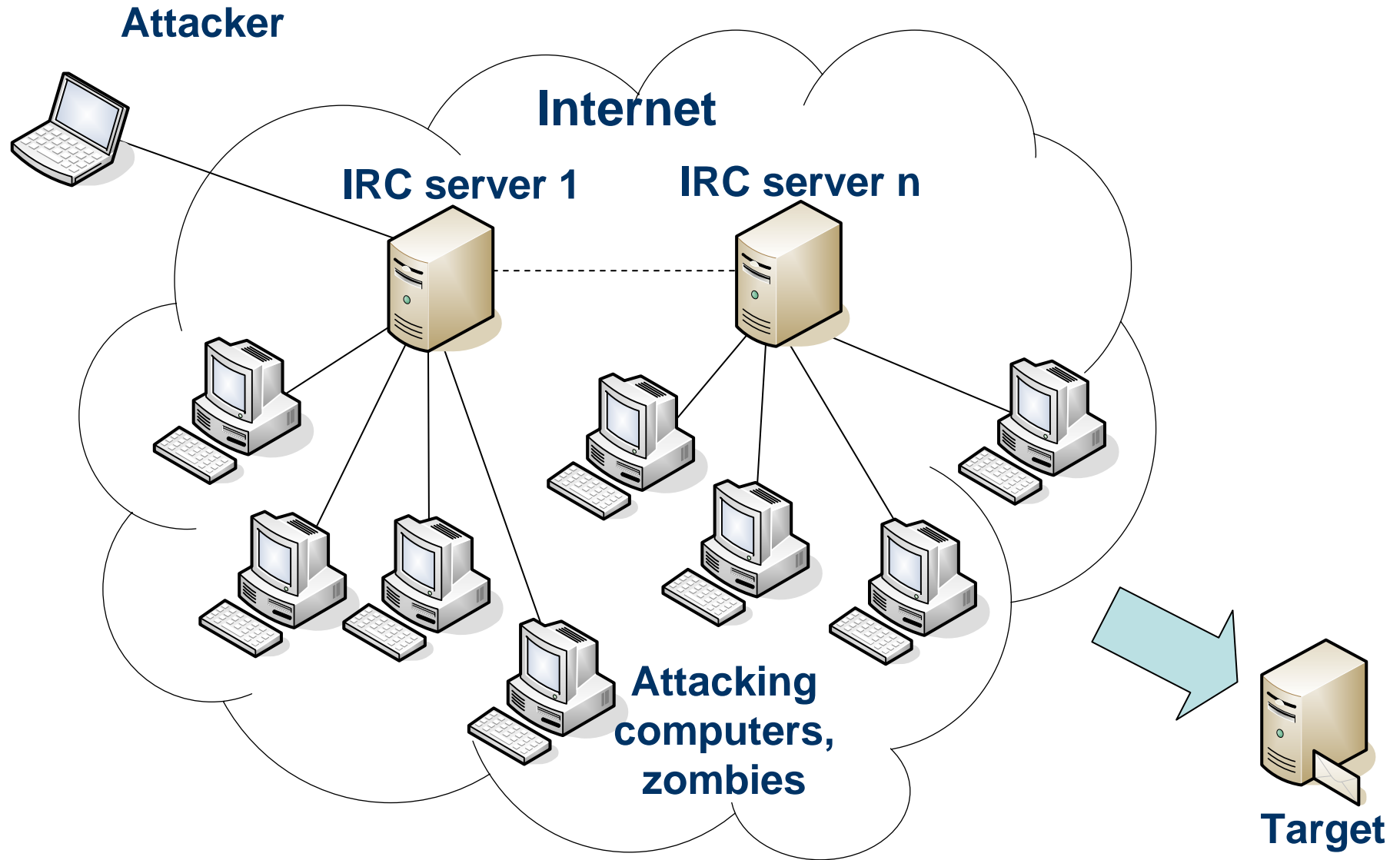
Botnet

- RoBOT NETwork
 - Gépek megfertőzése
 - Fertőzött gépekből hálózat kialakítása
 - Vezérlésre várakozás
 - Támadás, vezérlésre (DDoS, spam, etc.)
 - Frissítés
 - További fertőzések stb.
-
- Legnagyobb botnetek mérete milliós nagyságrendű

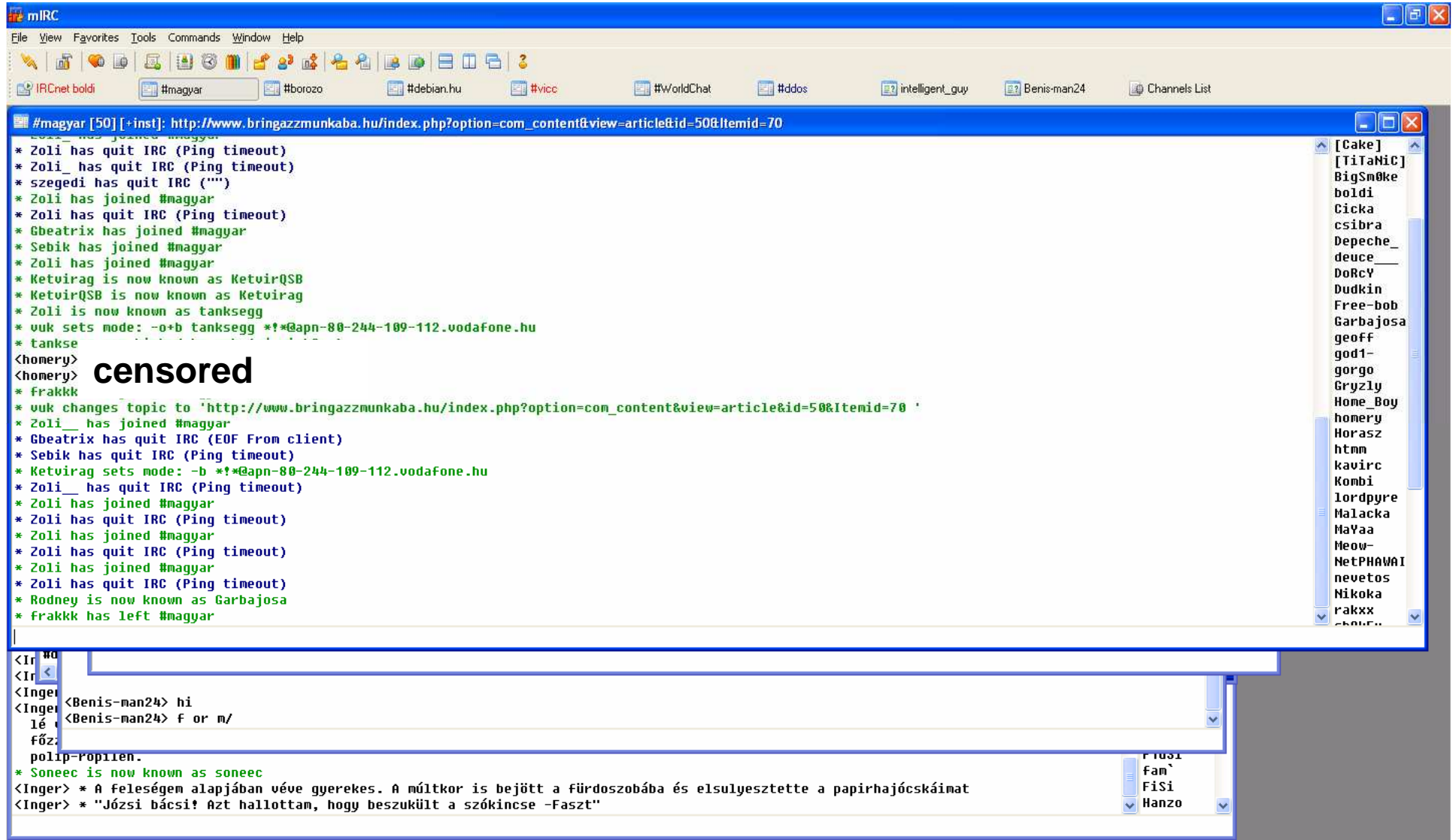
Botnet history



DDoS – Botnet with IRC



IRC-Internet Relay Chat



The screenshot shows the mIRC IRC client interface. The main window displays a chat log for the #magyar channel. The log contains several system messages (e.g., "Zoli has quit IRC", "Gbeatrrix has joined #magyar") and a message from user 'homery' that has been replaced with "censored". A URL is also visible in the log: "http://www.bringazzmunkaba.hu/index.php?option=com_content&view=article&id=50&Itemid=70".

On the right side, there is a channel list window showing a scrollable list of channels including [Cake], [TiTaNiC], BigSmoke, boldi, Cicka, csibra, Depeche_deuce, DoRcY, Dudkin, Free-bob, Garbajosa, geoff, god1-gorgo, Gryzly, Home_Boy, homery, Horasz, htmm, kavirc, Kombi, lordpyre, Malacka, MaYaa, Meow-NetPHAWAI, nevetos, Nikoka, rakxx, and soneec.

At the bottom, there is a text input area with a scrollable history of messages. The visible messages include:

- <Inger> <Benis-man24> hi
- <Inger> <Benis-man24> f or m/
- lé
- főz
- pollp-ropilen.
- * Soneec is now known as soneec
- <Inger> * A feleségem alapján véve gyerekes. A múltkor is bejött a fürdőszobába és elsulyesztette a papirhajócskáimat
- <Inger> * "Józsi bácsi! Azt hallottam, hogy beszukült a szókingse -Faszt"

IRC botnets

- IRC botnets: A controller can send messages to a channel
- The messages are received by the bots on the same channel
- (the servers relay the messages)
- The channel might be protected e.g. with password
- (of course, this can be recovered from active bots or by sniffing network activity: IRC is a cleartext protocol)
- The messages contain the commands of the controller/owner
- Bots can test the authenticity of the messages in some fashion

centralized vs. P2P botnet

- IRC-based and other centralized botnets have drawbacks
- A new trend for botnets is using P2P technologies
- DHT (distributed hash tables) based techniques are common
- However, e.g. delay might be higher for P2P botnets

	Planning	Botnet detection	Delay	Survivability	Identification of the controller/ owner
centralized	easy (1)	easy (1)	small (3)	bad (1)	easy (1)
P2P	hard (3)	hard (3)	medium (2)	good (3)	hard (3)

How to determine the size of the botnet?

- The size of the botnet is an important parameter. A large botnet can be more dangerous
- Counting individual IP addresses can give false results (e.g. bots behind NAT)
- The size of the botnet constantly changes – counting can also take time -> error
- IRC based botnets: activity of the bots might be visible, easy to count
- P2P botnets: e.g. doing queries in the DHT; sometimes the botnet uses IDs to identify individual bots – last ID might be queried

What to do against botnets

- Identification, size estimation
- Upgrade, patch against vulnerabilities (sometimes the patch gives hints to the attackers)
- Patch the vulnerable hosts remotely: illegal
- Find the owner of the botnet (hard task)
- Get control over the botnet (better botnets, harder to do)
- Support removal (by tools, knowledge): slow
- Eliminate upgrade possibilities (e.g domains, web pages) or control mechanism (disable communication, injecting code): harder and harder

Conficker botnet

- MS08-067 vulnerability is used
- A,B and C variants exist
- Conficker is a DLL
- Using the vulnerability it inserts itself into the system as a system service
- Also uses USB drives to infect – DLL + rundll32.exe (turn off auto-run for USB drives!)
- Update: Time-seeded random domain names are used to download encrypted binaries by HTTP.

- Source: Analysis of honeynet.org

Vulnerability used by Conficker

- Vulnerability: NetpwPathCanonicalize() in netapi32.dll. On an established SMB channel (port 445), a path string is canonicalized. E.g. aaa\bbb\..\ccc -> aaa\bbb
- With a specially crafted path string it is possible to move beyond the start of a stack buffer and overwrite return address (not a classical buffer overflow, but similar)
- PEB shellcode is used, “00” bytes are avoided with an xor encryption routine

- Conficker hooks some system calls
- E.g. DNS: to filter out for antivirus websites

DLL	Function
dnsapi.dll	DnsQuery_A DnsQuery_UTF8 DnsQuery_W Query_Main
netapi32.dll	NetpwPathCanonicalize
ntdll.dll	NtQueryInformationProcess
wininet.dll	InetnetGetConnectedState
ws2_32.dll	sendto

Table 1: *Functions hooked by Conficker.C*

NetpwCanonicalize hook

- First of all: no other botnets should be able to infect this computer
- Conficker: if “\.\” is found, then the “shellcode” is checked.
- Can decide if the exploit is coming from another conficker instance
- If a special “[http://..](#)” string is found in the data, conficker tries to use this to update itself.
- The behavior of the function is slightly modified ->ability to detect the bot
- Update checking: if RSA signature does not exist -> no update (SHA-1, 1024 bit RSA -> latest Conficker 4096 bit RSA + unknown hash)
- SHA-1 is from OpenSSL library

Upgrade mechanism

- Domain flux: For the update, conficker A/B generates 250-250 random domain names, daily.
- Antivirus companies tried to preregister them
- Conficker.C uses 50.000 domain names, daily
- The PRNG is seeded by the current time
- Time synchronization: downloads web pages (google, yahoo,...) and uses the time data (day, month, year) in the HTTP response

```
HTTP/1.1 200 OK
Date: Fri, 20 Mar 2009 17:01:13 GMTServer: BWS/1.0
Content-Length: 1809
Content-Type: text/html
Cache-Control: private
Expires: Fri, 20 Mar 2009 17:01:13 GMT
```

Conficker domain generation algorithm

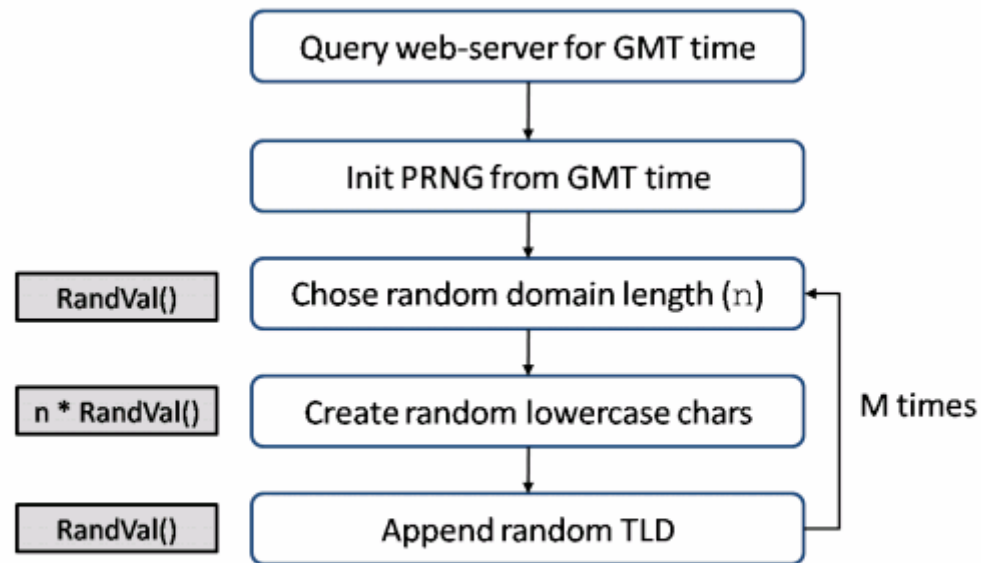


Figure 8: Domain name generation algorithm

	Conficker.A	Conficker.B	Conficker.C
Domains/ day	250	250	50.000
Domain name length	8-11	8-11	4-9
TLD suffixes	5	7	110

Table 3: Domain name generation facts

Conficker upgrade

- The generated domain name is checked for updates
- Updates are protected with RSA signatures
 - public key is in the bot itself
 - 1024 bit long in Conficker.A, 4096 bits for the other variants
 - The public key is a good signature to search for (bot identification)

Conficker blacklists

- Blacklist of network addresses is used by conficker
 - To avoid identification by these targets
 - To avoid scanning low-yield networks (expecting that most of the computers are patched here)
- E.g. IP addresses of the following companies are included:

Kaspersky

Trend Micro

Symantec

McAfee

F-Secure

Avira

Bitdefender

Microsoft Corp.

Microsoft Education

Microsoft License

Microsoft Visual Studios

Removal of Conficker

- Conficker detects removal tools and tries to avoid removal
- Conficker code is packed (polymorphic) on the network or in the file system
- However, on the target computer the code is unpacked while running
 - Easier to detect running processes
- The code is stored under random file names
 - not fully random (depends on the variant)
- Special flags and security settings on the file is used
- Every instance should be removed to avoid re-infection
- A trick: Conficker uses OS mutexes to avoid running multiple instances. The mutex generation is based on CRC. Might be used to avoid re-infections.

Hidden Conficker file

```
C:\Python25>python.exe
>>> f=file("c:/windows/system32/syysl.dll","r")

IOError: [Errno 13] Permission denied: 'c:/windows/system32/syysl.dll'

C:\Python25>dir c:\windows\system32\syyisl.dll
Directory of c:\windows\system32

File Not Found

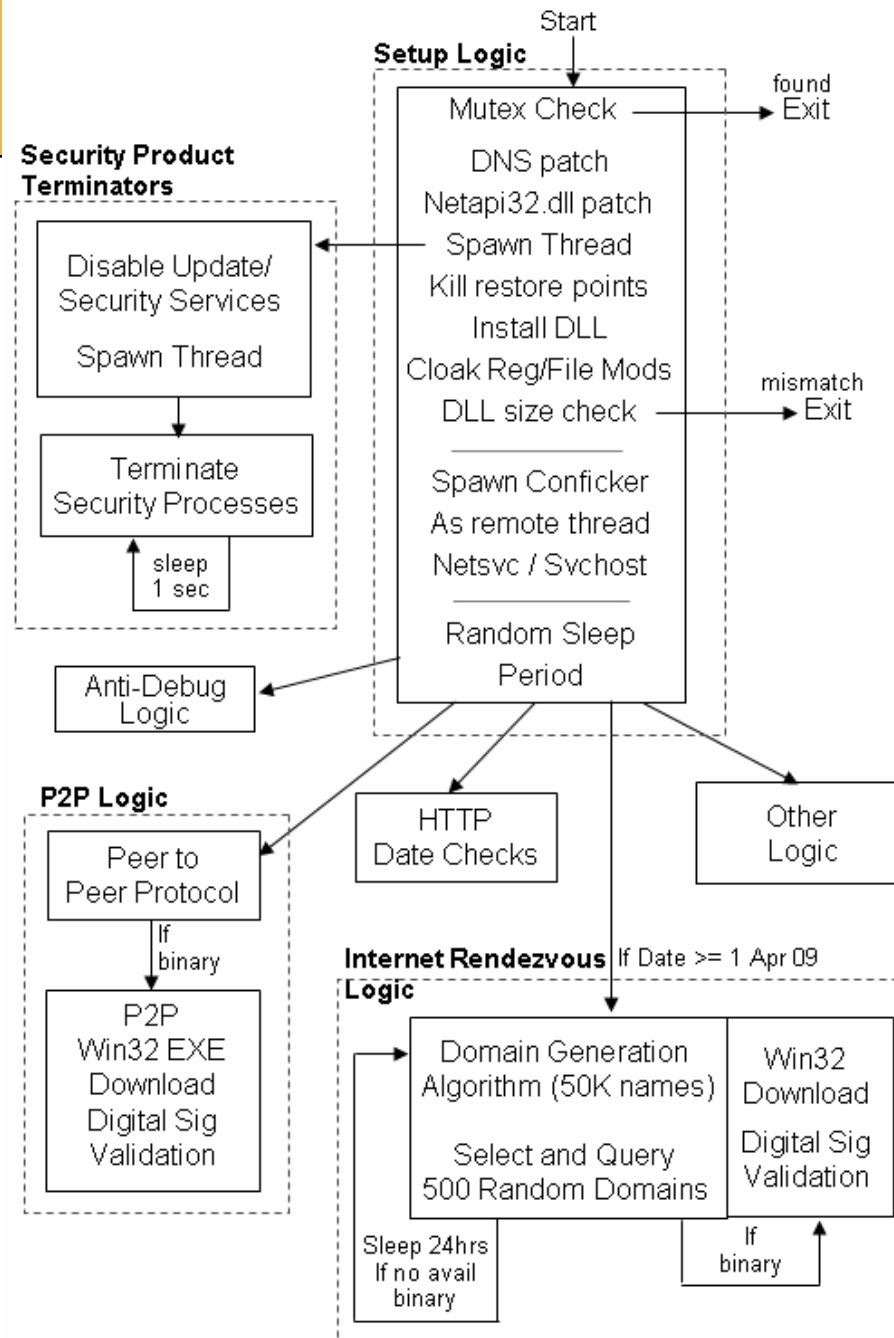
C:\Python25>dir /ah c:\windows\system32\syyisl.dll
Directory of C:\WINDOWS\system32

08/04/2004  01:00 PM           171,376 syysl.dll
             1 File(s)           171,376 bytes
```

How to identify bots in Conficker

- DNS sinkhole – antivirus countermeasure
- Update DNS names (getting queries from infected computers): although cannot inject any code as RSA signature might fail, the querying computer can be identified.
- Scanning on infected computers, removal tools – problematic
- Using the P2P approach of conficker

Conficker.C



Summary

- It was a long road from a single virus to the current sophisticated malware and botnet
- Every malware is different
- We must understand how they work to be able to protect against them
- Although lot of things has been shown, still only a small fraction of the knowledge about malware was included in the current slides
- No perfect protection exists currently
- Infrastructural changes might be needed in the future