Mid-term 29th November

2 groups – 2 exams: by LAST NAME

10:15-11:30 From Jeanmonod to Zrouga

11:45-13:00 From Abbey to Jalal

Inverse order than the first midterm!!!

Be seated by then!

1h15 for the exam

Multiple choice questions
True/False questions
2-line answers

Understanding concepts
~~**Memorizing concepts**~~

1st group: you **CANNOT** leave before 11:30 and you **CANNOT** use your phone/computer

Mid-term 29th November

- Pen blue or black (no others) - NO PENCIL
- Closed book, no calculator, no phone, no laptop
- CAMIPRO
- Pay attention to calligraphy, pretty please
- English (On essaie mais notre français n’est pas bon :/)

Will contain questions from the beginning!!!!
Until the last slide I explain in this class

Understanding concepts
~~Memorizing concepts~~

Mid-term 29th November

NEXT WEEK

Carmela’s Office hours

Tuesday 27th 1PM – 2PM

Last week – Trusted computing

TRUSTED HARDWARE

“A piece of hardware can be trusted if it always behaves in the expected manner for the intended purpose”

-Trusted Computing Group

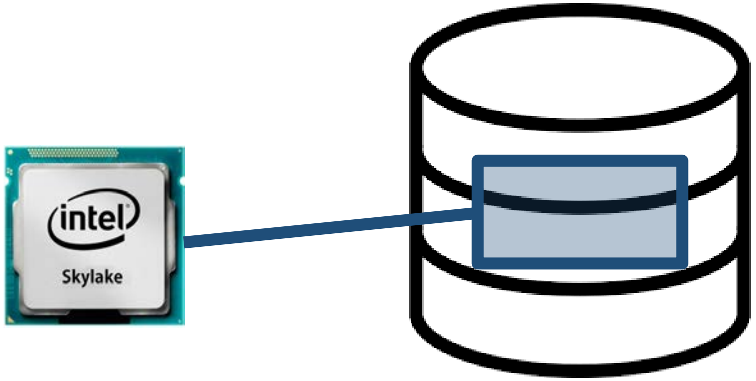
DEDICATED DEVICES

Operations **inside** the device

Optimized for cryptography

Strong adversary with physical access

SECURE ENCLAVES



The device supports the creation of a secure execution environment **in memory**

Defend from other software and firmware

Last week – 3 properties

ISOLATION

mechanism to constrain who and what has access to programs and data in the device

Tamper resistance: hard to open

Tamper evident: you can see if it has been opened

Tamper responsive: delete keys when attacked

Resistance to side-channel attacks and physical probing

Last week – 3 properties

ATTESTATION

mechanism that allows a hardware module to prove, to an authorized party, that it is in a specific state

Attest there is secure hardware:

the device has a key (endorsement key) to prove it is genuine

Attest the state of the OS:

after a series of instructions the state of registers is as expected

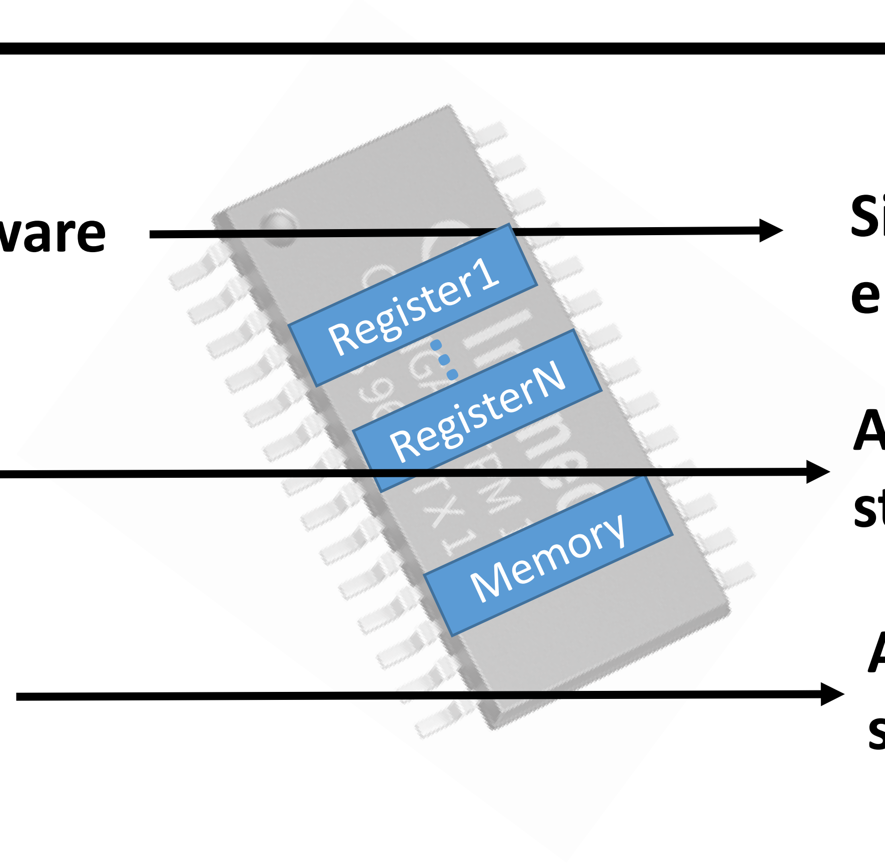
Attest the state of the code:

signature on the code (needs to correspond with the register values!)

Last week – 3 properties

ATTESTATION

mechanism that allows a hardware module to prove, to an authorized party, that it is in a specific state



Attest there is secure hardware → **Signature with manufacturer endorsement key**

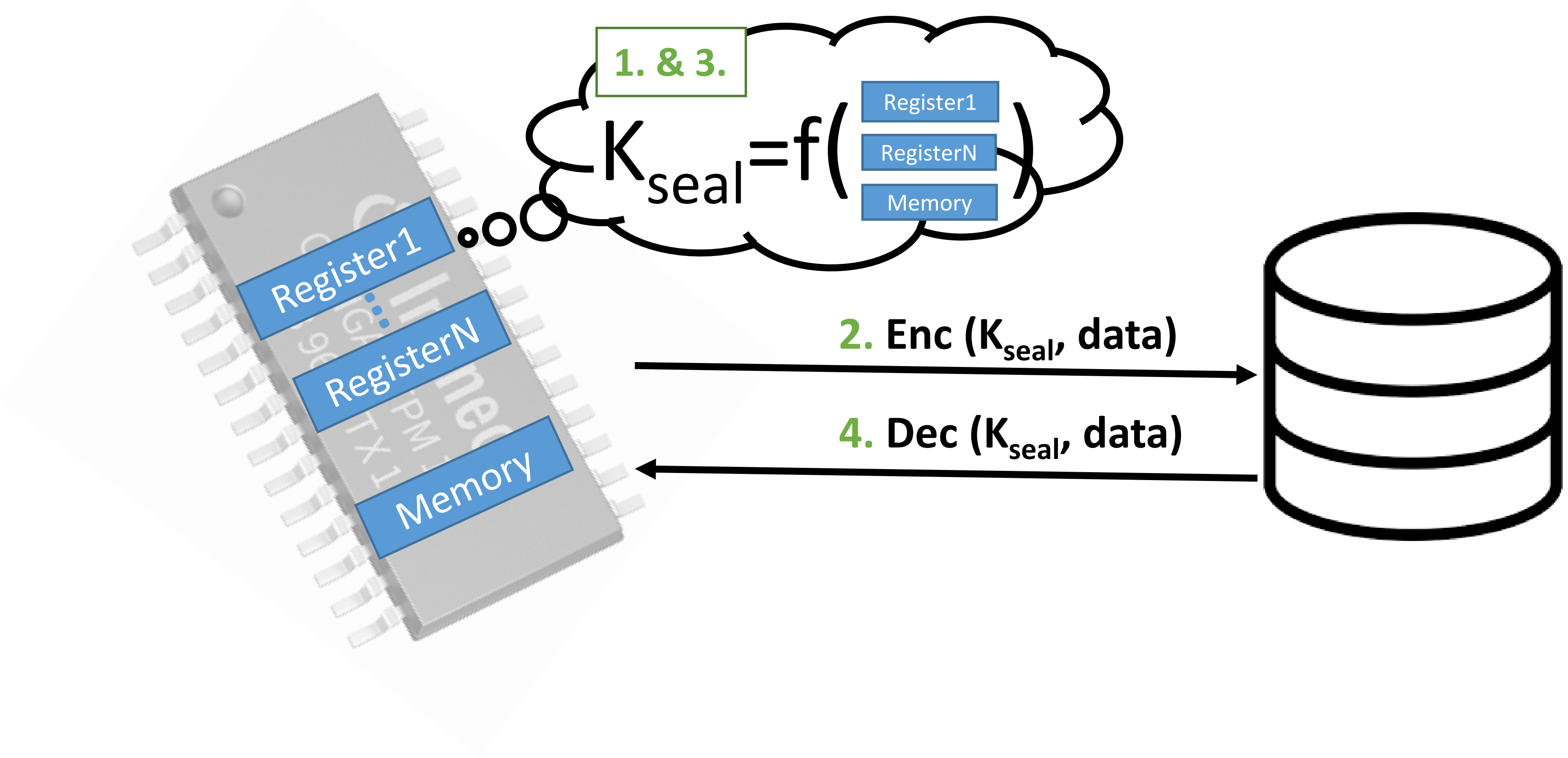
Attest the state of the OS → **Attest to register and memory state after booting: healthy boot**

Attest the state of the code → **Attest to register and memory state after code execution**

Last week – 3 properties

SEALING

a sealed storage protects private information by binding it to platform-configuration information including the software and hardware being used



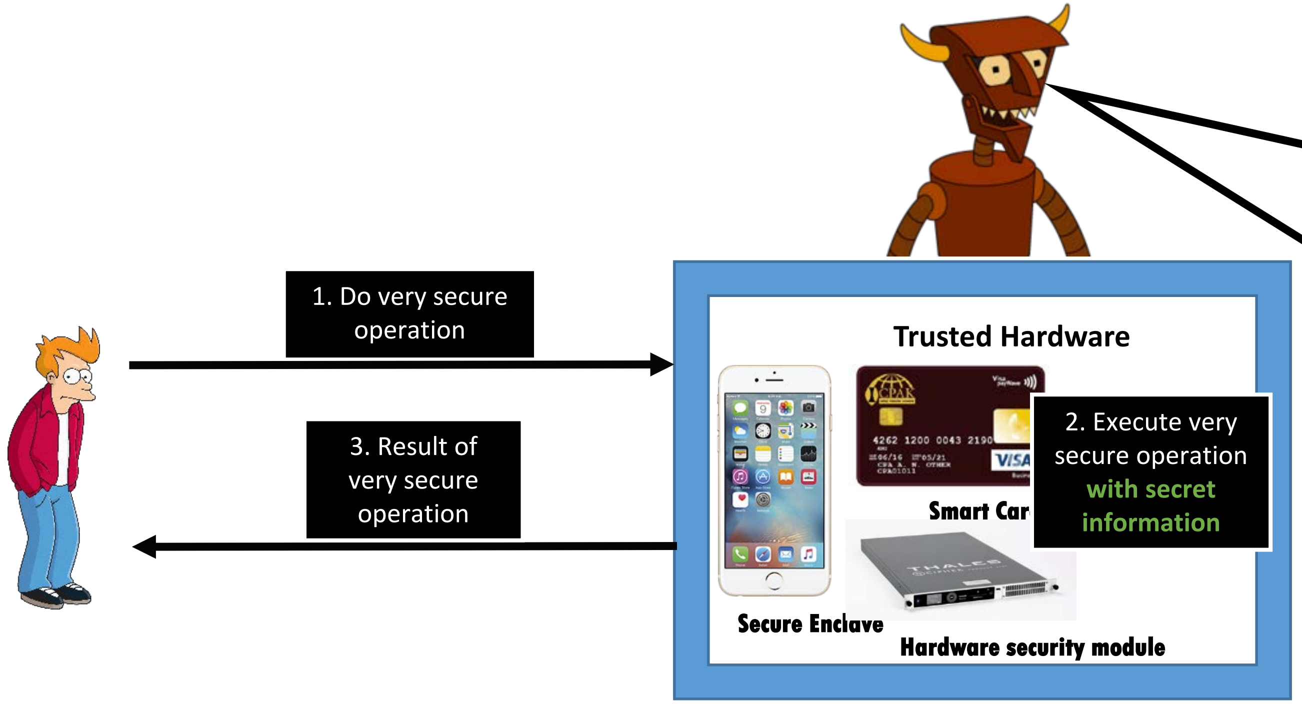
1. Before shutting down use internal state as seed for generating a key.

2. Securely store data (e.g., keys) in persistent memory.

3. When waking up use internal state as seed for generating key

4. Recover data from secure storage

Last week – Side channels



1. Do very secure operation

3. Result of very secure operation

Trusted Hardware

2. Execute very secure operation **with secret information**

Smart Car

Secure Enclave

Hardware security module

From **outside** learn secret information inside the trusted hardware by observing:

1. Time of execution
2. Power
3. Electromagnetic emanations

And try to find differences that depend on the secret

Side channel countermeasures

GOAL: Prevent secret inference from observable state

- **Hiding**: lowers signal to noise ratio
  - Noise generator, randomized execution order,…
- **Masking**: (secret sharing) splits state into shares; forces adversary to recombine leakage
  - Boolean or arithmetic masking, Higher-order masking
- **Leakage Resilience**: prevents leakage aggregation by updating secret every time the program is executed (very effective at a high cost)

Final remark:
Trusted hardware = no trust on anyone?

Backup data from Apple

- **HSM manufacturer.** It controls the production of the “black box”.
- **Apple.** To install the correct code the first time (attestation can help).

Private contact discovery for Signal

- **Intel.** It controls the production of the (black box) SGX system. It also controls the attestation keys built into every device.
- **Signal.** Not for running the correct set intersection code (the attestation allows to check that), but for not leaking any data from the smart phone application.

Trusted hardware - Lessons learned

It exists, and offers three properties

isolation: it is not possible to “peek” inside

attestation: it can prove that it does what you think it is doing

sealing: it can store secrets in memory that can only be recovered by itself

Building trusted hardware is difficult: side channel attacks

Trusted hardware means to trust the manufacturer!

Separation of privilege! Use several cards with advanced cryptography

Trusted hardware - Lessons learned

It exists, and offers three properties

isola

atte

seali ed by itself

Build

Trus

cryptography

MOST IMPORTANT LESSON

Having trusted hardware doesn’t mean you don’t need to think about protocols!!

New content!!

Computer Security (COM-301)
Malware (more on attacks)

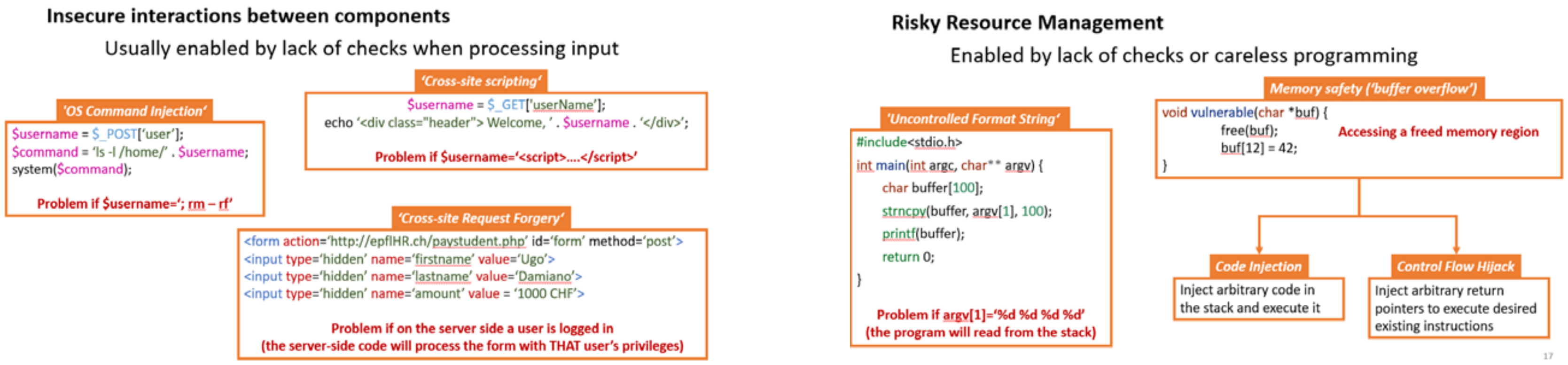
Carmela Troncoso

SPRING Lab

carmela.troncoso@epfl.ch

Some slides/ideas adapted from: Gianluca Stringhini / Emiliano de Cristofaro / George Danezis

Previous attacks: the adversary actively exploits model/ design/ implementation errors



Insecure interactions between components

Usually enabled by lack of checks when processing input

'OS Command Injection'

```
$username = $_POST['user'];
$command = 'ls -l /home/' . $username;
system($command);
```

Problem if $username='; rm – rf'

'Cross-site scripting'

```
$username = $_GET['userName'];
echo '<div class="header"> Welcome, ' . $username . '</div>';
```

Problem if $username='<script>....</script>'

'Cross-site Request Forgery'

```
<form action='http://epflHR.ch/paystudent.php' id='form' method='post'>
<input type='hidden' name='firstname' value='Ugo'>
<input type='hidden' name='lastname' value='Damiano'>
<input type='hidden' name='amount' value = '1000 CHF'>
```

Problem if on the server side a user is logged in (the server-side code will process the form with THAT user's privileges)

Risky Resource Management

Enabled by lack of checks or careless programming

'Uncontrolled Format String'

```
#include<stdio.h>
int main(int argc, char** argv) {
    char buffer[100];
    strncpy(buffer, argv[1], 100);
    printf(buffer);
    return 0;
}
```

Problem if argv[1]='%d %d %d %d' (the program will read from the stack)

Memory safety ('buffer overflow')

```
void vulnerable(char *buf) {
        free(buf);
        buf[12] = 42;
}
```

Accessing a freed memory region

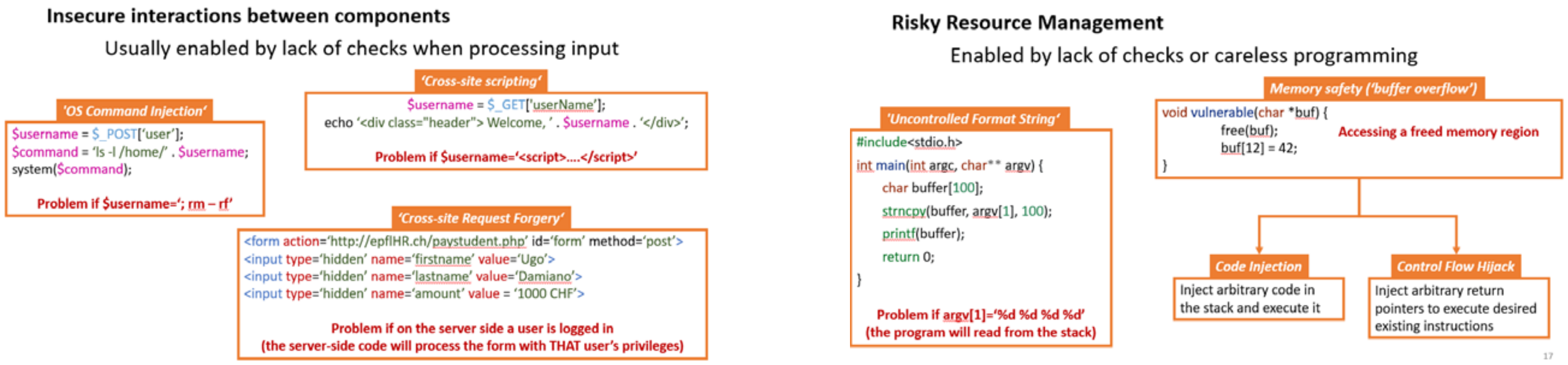
Code Injection

Inject arbitrary code in the stack and execute it

Control Flow Hijack

Inject arbitrary return pointers to execute desired existing instructions

Previous attacks: the adversary actively exploits model/ design/ implementation errors



Insecure interactions between components

Usually enabled by lack of checks when processing input

'OS Command Injection'

```
$username = $_POST['user'];
$command = 'ls -l /home/' . $username;
system($command);
```

Problem if $username='; rm – rf'

'Cross-site scripting'

```
$username = $_GET['userName'];
echo '<div class="header"> Welcome, ' . $username . '</div>';
```

Problem if $username='<script>....</script>'

'Cross-site Request Forgery'

```
<form action='http://epflHR.ch/paystudent.php' id='form' method='post'>
<input type='hidden' name='firstname' value='Ugo'>
<input type='hidden' name='lastname' value='Damiano'>
<input type='hidden' name='amount' value = '1000 CHF'>
```

Problem if on the server side a user is logged in
(the server-side code will process the form with THAT user's privileges)

Risky Resource Management

Enabled by lack of checks or careless programming

'Uncontrolled Format String'

```
#include<stdio.h>
int main(int argc, char** argv) {
      char buffer[100];
      strncpy(buffer, argv[1], 100);
      printf(buffer);
      return 0;
}
```

Problem if argv[1]='%d %d %d %d'
(the program will read from the stack)

Memory safety ('buffer overflow')

```
void vulnerable(char *buf) {
            free(buf);
            buf[12] = 42;
}
```

Accessing a freed memory region

Code Injection

Inject arbitrary code in the stack and execute it

Control Flow Hijack

Inject arbitrary return pointers to execute desired existing instructions

17

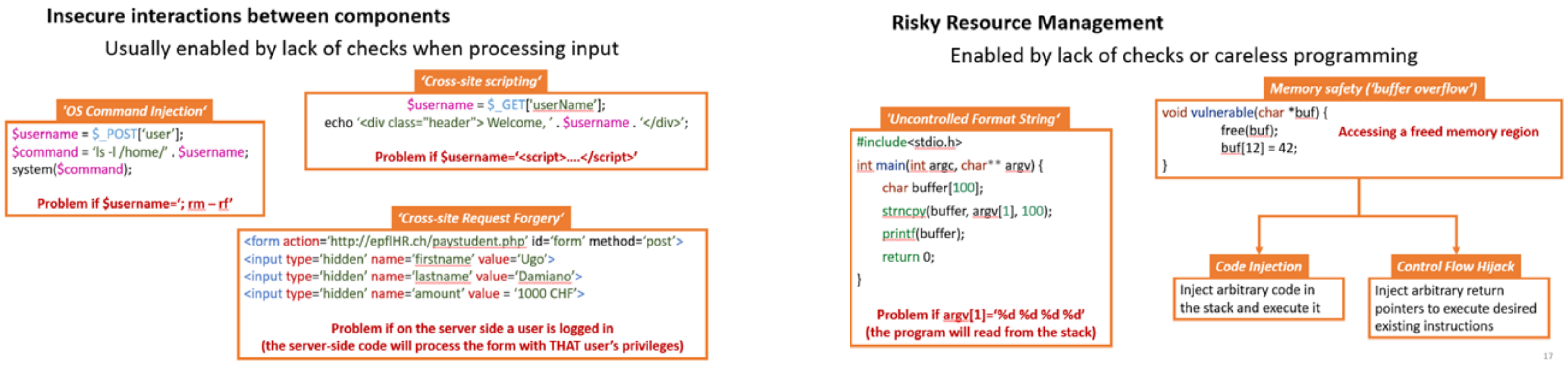
Expert adversary

requires deep understanding of computer systems and networks

"Manual" adversary

requires manual coding and testing to find the vulnerabilities and exploit them

Previous attacks: the adversary actively exploits model/ design/ implementation errors



Insecure interactions between components

Usually enabled by lack of checks when processing input

'OS Command Injection'

```
$username = $_POST['user'];
$command = 'ls -l /home/' . $username;
system($command);
```

Problem if $username='; rm – rf'

'Cross-site scripting'

```
$username = $_GET['userName'];
echo '<div class="header"> Welcome, ' . $username . '</div>';
```

Problem if $username='<script>....</script>'

'Cross-site Request Forgery'

```
<form action='http://epflHR.ch/paystudent.php' id='form' method='post'>
<input type='hidden' name='firstname' value='Ugo'>
<input type='hidden' name='lastname' value='Damiano'>
<input type='hidden' name='amount' value = '1000 CHF'>
```

Problem if on the server side a user is logged in
(the server-side code will process the form with THAT user's privileges)

Risky Resource Management

Enabled by lack of checks or careless programming

'Uncontrolled Format String'

```
#include<stdio.h>
int main(int argc, char** argv) {
    char buffer[100];
    strncpy(buffer, argv[1], 100);
    printf(buffer);
    return 0;
}
```

Problem if argv[1]='%d %d %d %d'
(the program will read from the stack)

Memory safety ('buffer overflow')

```
void vulnerable(char *buf) {
        free(buf);
        buf[12] = 42;
}
```

Accessing a freed memory region

Code Injection

Inject arbitrary code in the stack and execute it

Control Flow Hijack

Inject arbitrary return pointers to execute desired existing instructions

Expert adversary

requires deep understanding of computer systems and networks

"Manual" adversary

requires manual coding and testing to find the vulnerabilities and exploit them

Do all adversaries have these capabilities?
(expertise + time)

Malware

Shortening for Malicious Code

Software that **fulfills author's malicious intent**

Intentionally written to cause adverse effects

Many flavors with a common characteristic: **perform some unwanted activity**

Malware

Shortening for Malicious Code

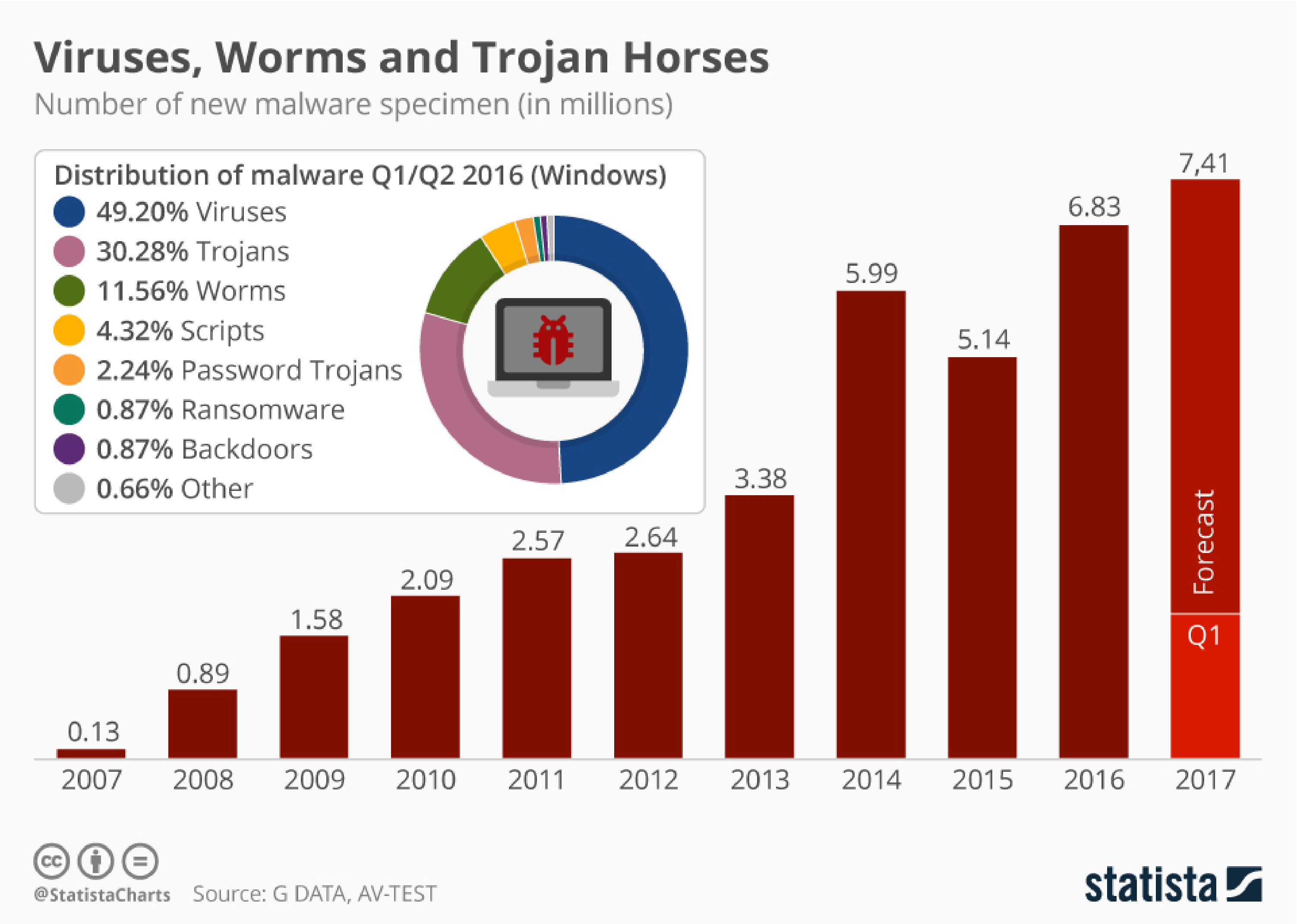
Software that **fulfills author's malicious intent**

Intentionally written to cause adverse effects

Many flavors with a common characteristic: **perform some unwanted activity**

Malware != virus

Virus is one type of Malware!



Viruses, Worms and Trojan Horses

Number of new malware specimen (in millions)

Source: G DATA, AV-TEST

https://www.statista.com/chart/10045/new-malware-specimen-and-share-of-windows-based-malware/

Malware – why the rise?

Homogeneous computing base

Windows/Android make very tempting targets

Unprecedented connectivity

Remote attacks or distributed attacks are increasingly easier

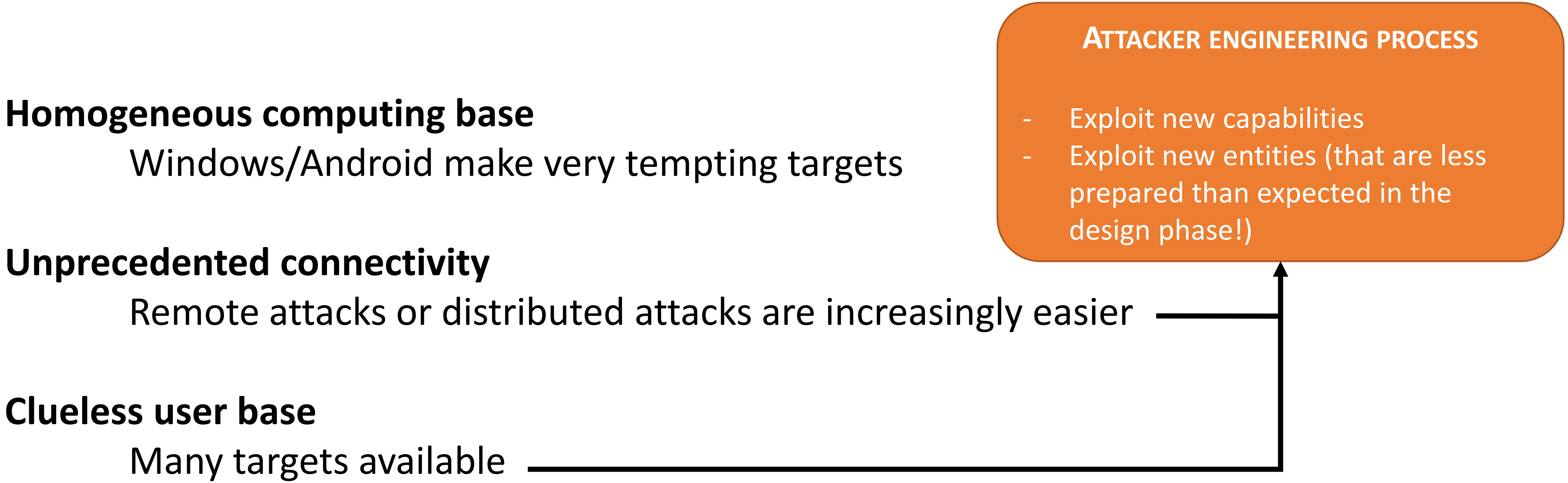
Clueless user base

Many targets available

Malicious code has become profitable!

Compromised computers can be sold and/or used to make money

Malware – why the rise?



Homogeneous computing base

Windows/Android make very tempting targets

Unprecedented connectivity

Remote attacks or distributed attacks are increasingly easier

Clueless user base

Many targets available

ATTACKER ENGINEERING PROCESS

- Exploit new capabilities
- Exploit new entities (that are less prepared than expected in the design phase!)

Malicious code has become profitable!

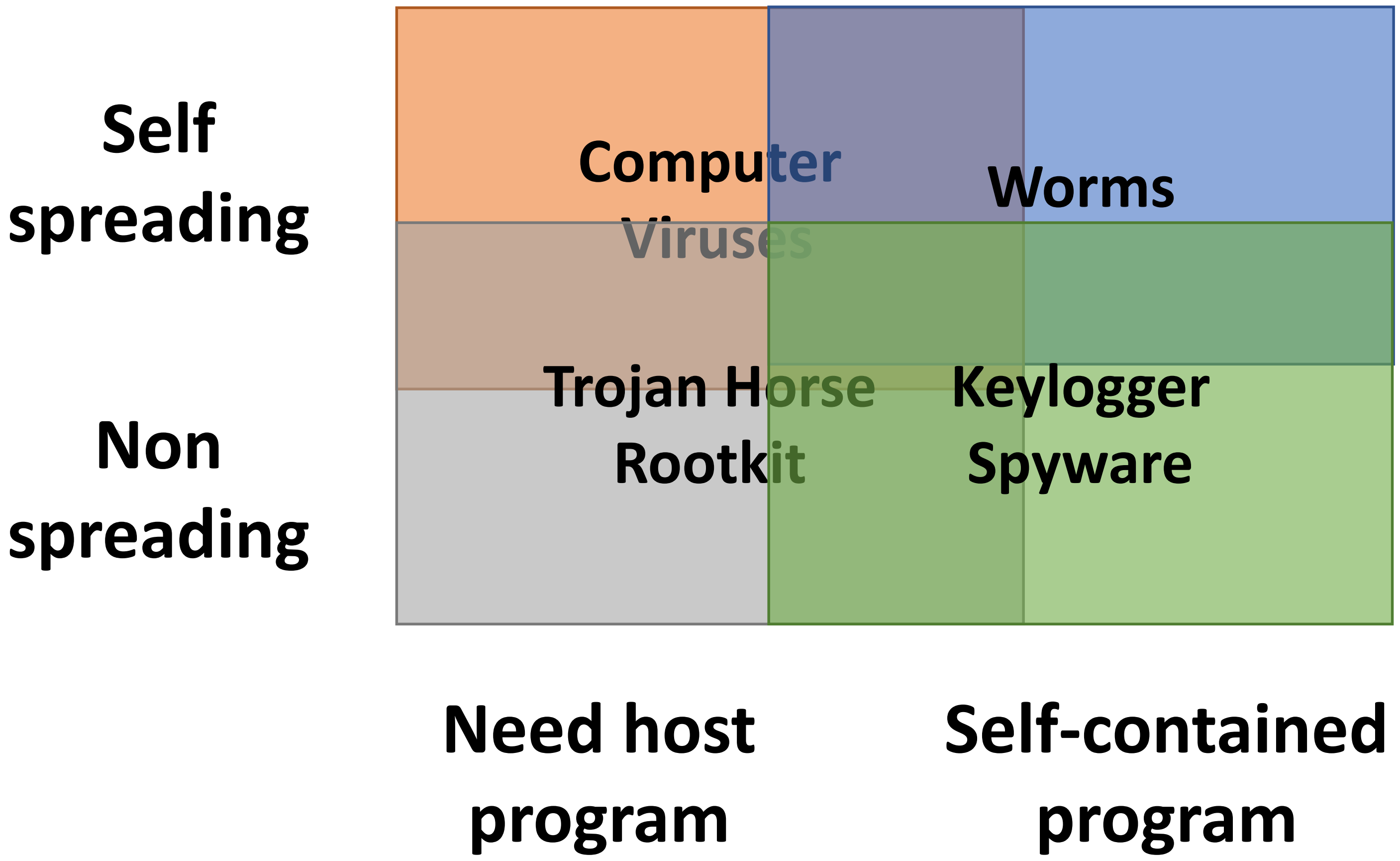
Compromised computers can be sold and/or used to make money

Malware – taxonomy

Self spreading	Computer Viruses	Worms
Non spreading	Trojan Horse Rootkit	Keylogger Spyware
	Need host program	Self-contained program

Malware – taxonomy

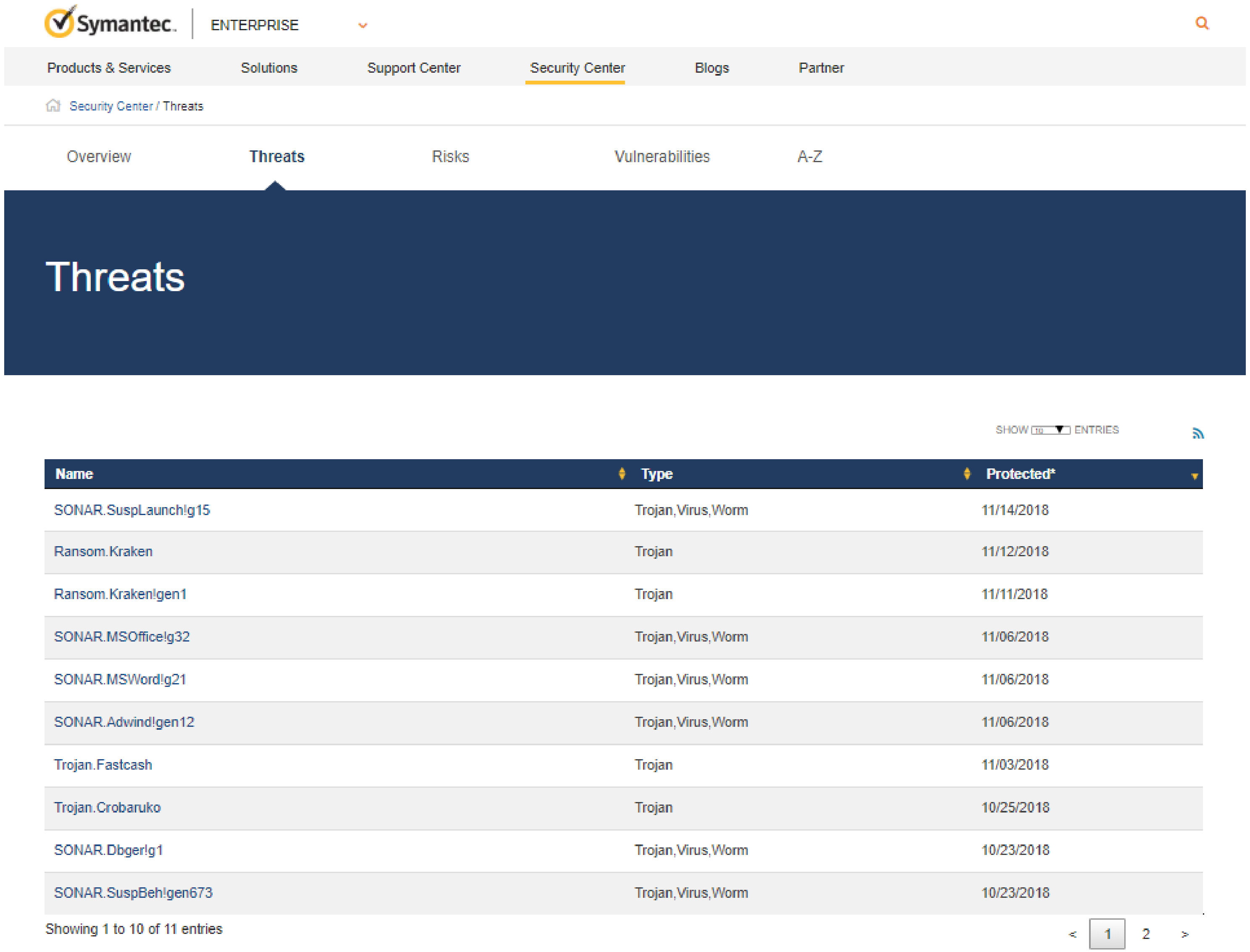
Modern malware tends to combine "the best" of the categories to achieve its purpose



	Need host program	Self-contained program
Self spreading	Computer Viruses	Worms
Non spreading	Trojan Horse Rootkit	Keylogger Spyware

Malware – taxonomy

Modern malware tends to combine "the best" of the categories to achieve its purpose



Name	Type	Protected*
SONAR.SuspLaunch!g15	Trojan,Virus,Worm	11/14/2018
Ransom.Kraken	Trojan	11/12/2018
Ransom.Kraken!gen1	Trojan	11/11/2018
SONAR.MSOffice!g32	Trojan,Virus,Worm	11/06/2018
SONAR.MSWord!g21	Trojan,Virus,Worm	11/06/2018
SONAR.Adwind!gen12	Trojan,Virus,Worm	11/06/2018
Trojan.Fastcash	Trojan	11/03/2018
Trojan.Crobaruko	Trojan	10/25/2018
SONAR.Dbger!g1	Trojan,Virus,Worm	10/23/2018
SONAR.SuspBeh!gen673	Trojan,Virus,Worm	10/23/2018

Showing 1 to 10 of 11 entries

https://www.symantec.com/security-center/

Lists of:

- **Threats**
- **Vulnerabilities**
- **Risks**

Virus (1)

Piece of software that **infects** programs to monitor / steal / destroy

modifies programs to include a (possibly modified) copy of the virus

cannot survive without the host

What are the virus permissions?

Virus (1)

RANSOMWARE: malware that threatens to destroy a system unless the owner pays money to receive the "antidote"

Piece of software that **infects** programs to monitor / steal / destroy

- modifies programs to include a (possibly modified) copy of the virus
- **cannot** survive without the host

What are the virus permissions?

Virus (1)

Piece of software that **infects** programs to monitor / steal / destroy

- modifies programs to include a (possibly modified) copy of the virus
- **cannot** survive without the host

A virus has the same permissions as the host

- can do anything that the host program is permitted to do
- **executes** secretly when the host program is run

Specific to operating system and hardware

- take advantage of their details and weaknesses

Virus (1)

Piece of software that **infects** programs to monitor / steal / destroy

- modifies programs to include a (possibly modified) copy of the virus
- **cannot** survive without the host

A virus has the same permissions as the host

- can do anything that the host program is permitted to do
- **executes** secretly when the host program is run

In this case, the host program would be acting as… ?

Specific to operating system and hardware

- take advantage of their details and weaknesses

Virus (1)

Yes! The confused deputy again!

Recurring problem in security!

Piece of software that **infects** programs to monitor / steal

modifies programs to include a (possibly modified) cop

cannot survive without the host

In this case, the host program would be acting as… ?

A virus has the same permissions as the host

can do anything that the host program is permitted to do

executes secretly when the host program is run

Specific to operating system and hardware

take advantage of their details and weaknesses

Virus (1)

Mitigation: follow the least privilege principle!

Yes! The confused deputy again!

Recurring problem in security!

In this case, the host program would be acting as… ?

Piece of software that **infects** programs to monitor / steal

- modifies programs to include a (possibly modified) cop
- **cannot** survive without the host

A virus has the same permissions as the host

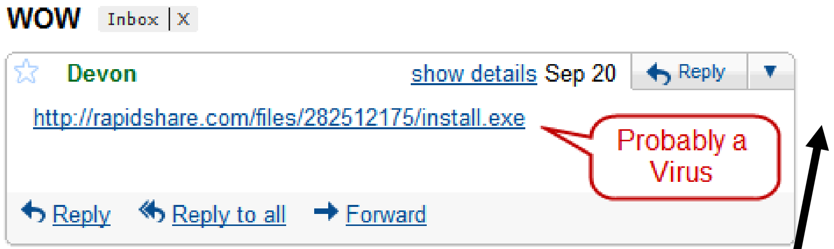
- can do anything that the host program is permitted to do
- **executes** secretly when the host program is run

Specific to operating system and hardware

- take advantage of their details and weaknesses

Virus (2)

Replicates to infect other content or machine (host spreads through network or hardware)



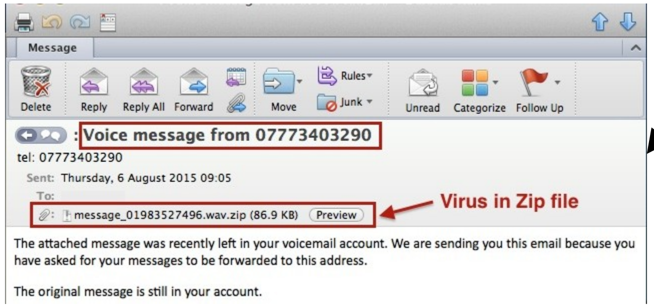
WOW

Devon Sep 20

http://rapidshare.com/files/282512175/install.exe

Probably a Virus

email



Voice message from 07773403290

tel: 07773403290

Sent: Thursday, 6 August 2015 09:05

To:

message_01983527496.wav.zip (86.9 KB)

Virus in Zip file

The attached message was recently left in your voicemail account. We are sending you this email because you have asked for your messages to be forwarded to this address.

The original message is still in your account.

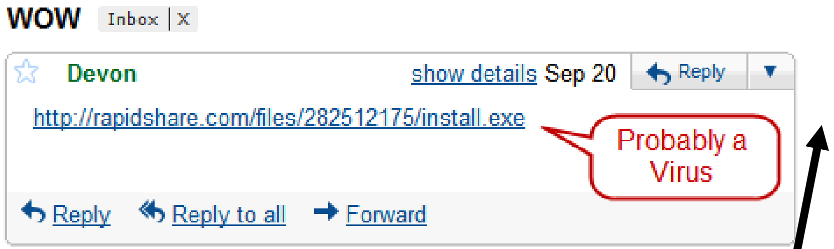
https://heimdalsecurity.com/blog/practical-online-protection-where-malware-hides/

Virus (2)

Replicates to infect other content or machine (host spreads through network or hardware)

web

email



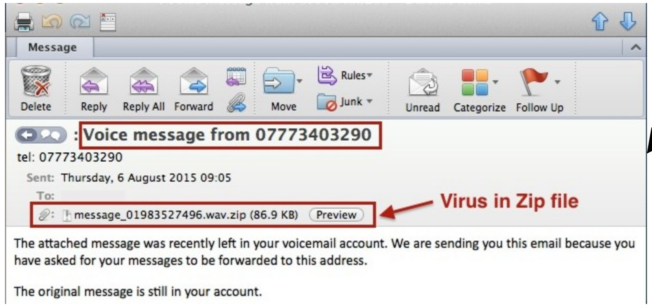
WOW Inbox | X

Devon show details Sep 20 Reply ▼

http://rapidshare.com/files/282512175/install.exe

Probably a Virus

Reply Reply to all Forward



: Voice message from 07773403290

tel: 07773403290

Sent: Thursday, 6 August 2015 09:05

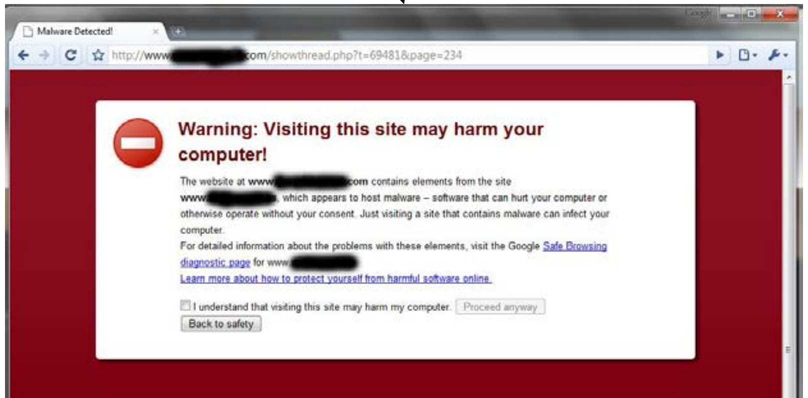
To:

message_01983527496.wav.zip (86.9 KB) Preview

Virus in Zip file

The attached message was recently left in your voicemail account. We are sending you this email because you have asked for your messages to be forwarded to this address.

The original message is still in your account.



Malware Detected!

http://www[redacted]com/showthread.php?t=69481&page=234

Warning: Visiting this site may harm your computer!

The website at www[redacted]com contains elements from the site www[redacted], which appears to host malware – software that can hurt your computer or otherwise operate without your consent. Just visiting a site that contains malware can infect your computer.

For detailed information about the problems with these elements, visit the Google Safe Browsing diagnostic page for www[redacted]

Learn more about how to protect yourself from harmful software online.

□ I understand that visiting this site may harm my computer. Proceed anyway

Back to safety

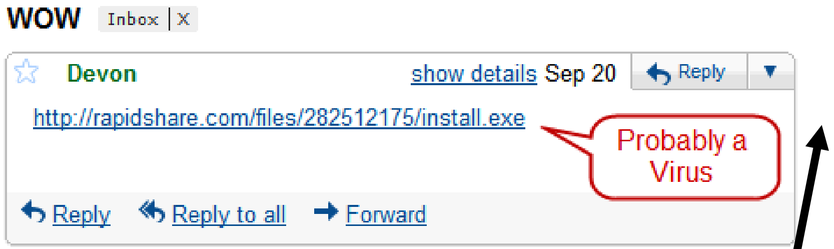
https://heimdalsecurity.com/blog/practical-online-protection-where-malware-hides/

Virus (2)

Replicates to infect other content or machine
(host spreads through network or hardware)

email

web



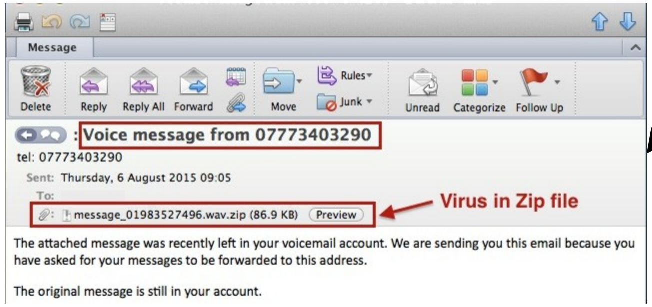
WOW Inbox | X

Devon show details Sep 20 Reply ▼

http://rapidshare.com/files/282512175/install.exe

Probably a Virus

Reply Reply to all Forward



: Voice message from 07773403290

tel: 07773403290

Sent: Thursday, 6 August 2015 09:05

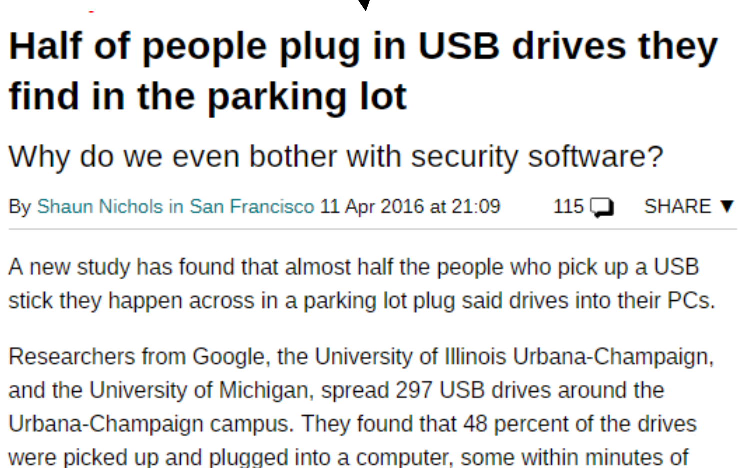
To:

message_01983527496.wav.zip (86.9 KB) Preview

Virus in Zip file

The attached message was recently left in your voicemail account. We are sending you this email because you have asked for your messages to be forwarded to this address.

The original message is still in your account.



Half of people plug in USB drives they find in the parking lot

Why do we even bother with security software?

By Shaun Nichols in San Francisco 11 Apr 2016 at 21:09 115 SHARE ▼

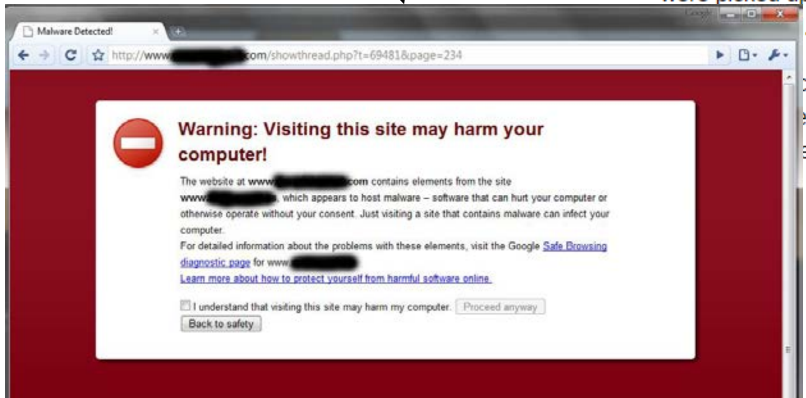
A new study has found that almost half the people who pick up a USB stick they happen across in a parking lot plug said drives into their PCs.

Researchers from Google, the University of Illinois Urbana-Champaign, and the University of Michigan, spread 297 USB drives around the Urbana-Champaign campus. They found that 48 percent of the drives were picked up and plugged into a computer, some within minutes of

.

community has long held the belief that users can be eered into picking up and plugging in seemingly lost USB ey find," the researchers reported this month.

whether driven by altruistic motives or human curiosity



Malware Detected!

http://www[redacted]com/showthread.php?t=69481&page=234

Warning: Visiting this site may harm your computer!

The website at www[redacted]com contains elements from the site www[redacted], which appears to host malware – software that can hurt your computer or otherwise operate without your consent. Just visiting a site that contains malware can infect your computer.

For detailed information about the problems with these elements, visit the Google Safe Browsing diagnostic page for www[redacted]

Learn more about how to protect yourself from harmful software online.

□ I understand that visiting this site may harm my computer. Proceed anyway

Back to safety

https://heimdalsecurity.com/blog/practical-online-protection-where-malware-hides/

Virus – where can they act

File infection

Overwrite (substitute the original program), Parasitic (append and modify)

Macro infection

Overwrite macro executed on program load (MS Excel, Word)

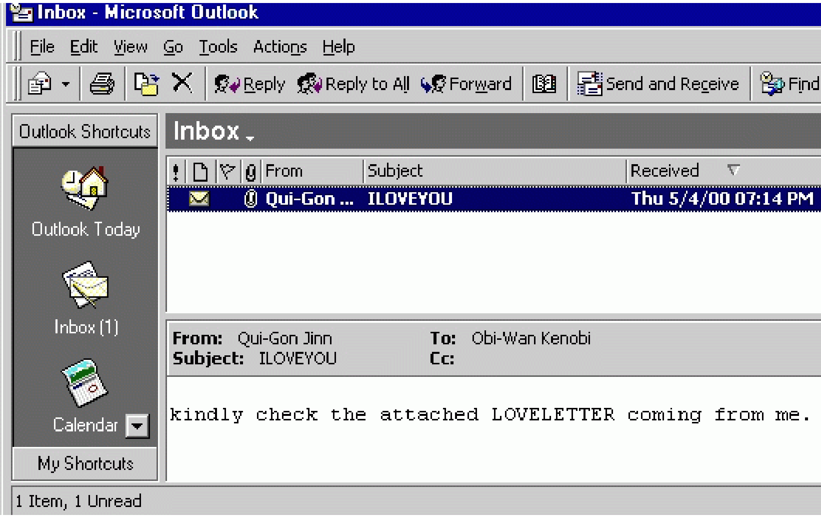
Need to find an exploit to insert the macro

Boot infection

Most difficult! …and most dangerous

Infect booting partition

Example Virus – ILoveYou (2000)



From	Subject	Received
Qui-Gon ...	ILOVEYOU	Thu 5/4/00 07:14 PM

From: Qui-Gon Jinn **To:** Obi-Wan Kenobi
Subject: ILOVEYOU **Cc:**

kindly check the attached LOVELETTER coming from me.

Target: Windows 9X/2000

`"LOVE-LETTER-FOR-YOU.txt.vbs“` sent as email attachment

Operation:

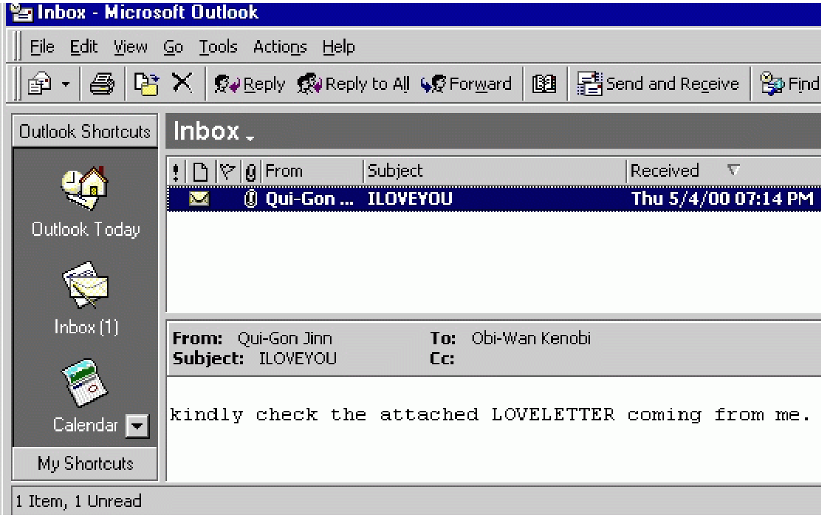
Replaced files with extensions JPG, JPEG, VBS, JS, DOC, … → unbootable!

Sent itself to each entry Outlook address book, sometimes changing subject

Downloaded the Barok Trojan: "WIN-BUGSFIX.EXE“ (steal passwords)

Damage: $10 billion in damages

Example Virus – ILoveYou (2000)



!	From	Subject	Received
	Qui-Gon ...	ILOVEYOU	Thu 5/4/00 07:14 PM

From: Qui-Gon Jinn **To:** Obi-Wan Kenobi
Subject: ILOVEYOU **Cc:**

```
kindly check the attached LOVELETTER coming from me.
```

1 Item, 1 Unread

Target: Windows 9X/2000

`"LOVE-LETTER-FOR-YOU.txt.vbs“` sent as email attachment

Known extension to Windows, hidden from users!
Users think they open a text file, not an executable

Operation:

Replaced files with extensions JPG, JPEG, VBS, JS, DOC, … → unbootable!
Sent itself to each entry Outlook address book, sometimes changing subject
Downloaded the Barok Trojan: "WIN-BUGSFIX.EXE“ (steal passwords)

Damage: $10 billion in damages

Virus – defenses

Antivirus Software

sequence of bytes/instructions that are known to be part of the virus

Signature-based detection

Database of byte-level or instruction-level signatures that match virus

Wildcards and regular expression can be used

Hash of known malicious programs

Heuristics (check for signs of infection – anomaly detection) and

incorrect header sizes, suspicious code section name

Behavioral signatures – detect series of changes done by a virus

Sandboxing

Run untrusted applications in restricted environment (e.g., use a VM)

Worm

Self-replicating computer program that uses a network to send copies of itself to other nodes

Does not need a host to survive

Autonomous spread over the network

Email harvesting (address book, inbox, browser cache)

Network share enumeration

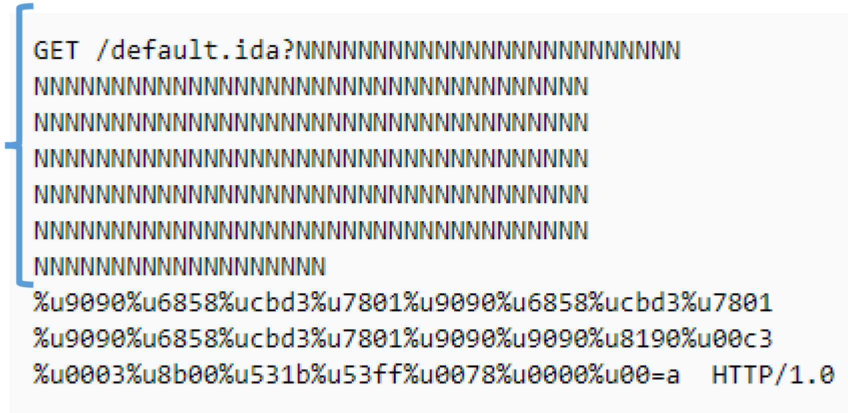
Scanning (at random or targeted)

Email: requires human interaction (fake `from`, hidden attachments)

Network: automated!

Worm example 1: Code Red (2001)

Exploits known buffer overflow in Microsoft IIS (web server)

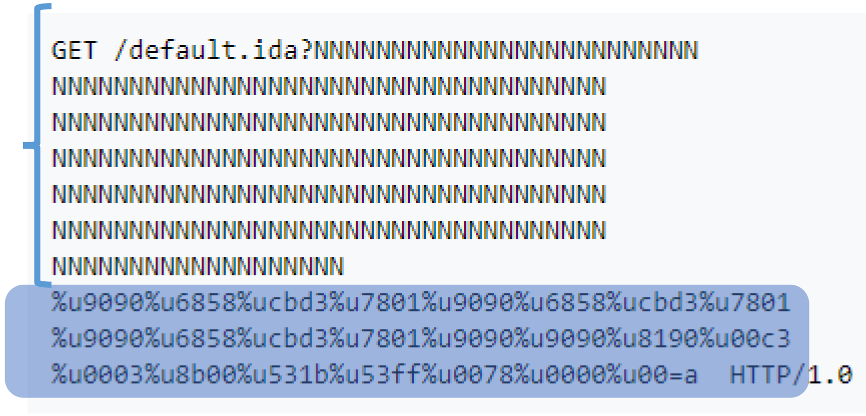


Achieve the overflow

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a  HTTP/1.0
```

Worm example 1: Code Red (2001)

Exploits known buffer overflow in Microsoft IIS (web server)



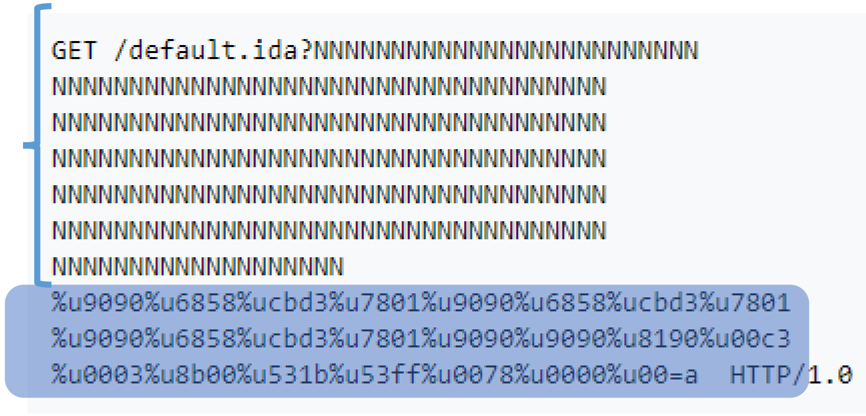
```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a  HTTP/1.0
```

Achieve the overflow

Payload: injected executable code

Worm example 1: Code Red (2001)

Exploits known buffer overflow in Microsoft IIS (web server)



Achieve the overflow

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a  HTTP/1.0
```

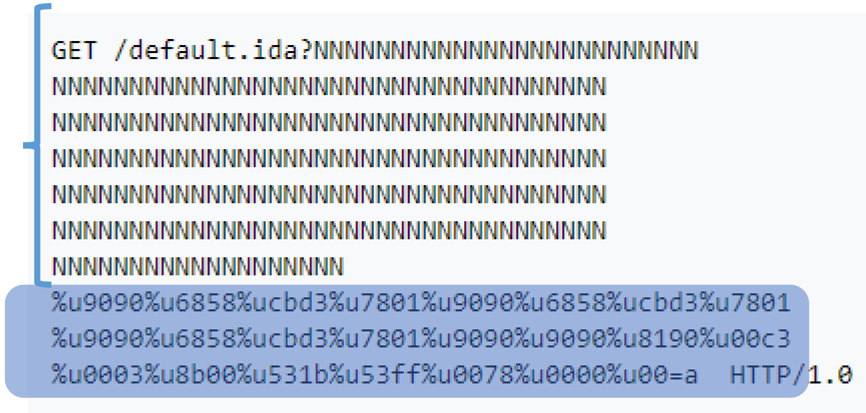
Payload: injected executable code

Lack of Sanitization!!

Worm example 1: Code Red (2001)

Exploits known buffer overflow in Microsoft IIS (web server)

Propagates through an HTTP request (TCP)



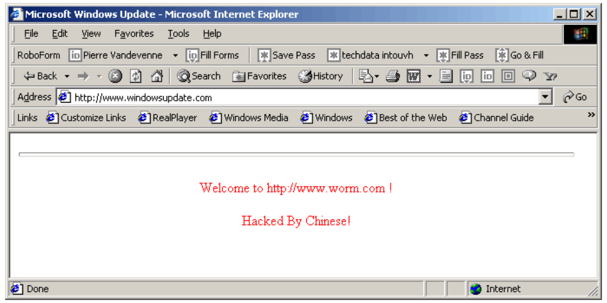
Achieve the overflow

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a  HTTP/1.0
```

Payload: injected executable code

Operation

1st <span>Defacing</span>:



Welcome to http://www.worm.com !

Hacked By Chinese!

2nd Date dependent:

Day 1-19: continue spreading (random IPs)

Day 20-27: Launch Denial of Service (among them White House)

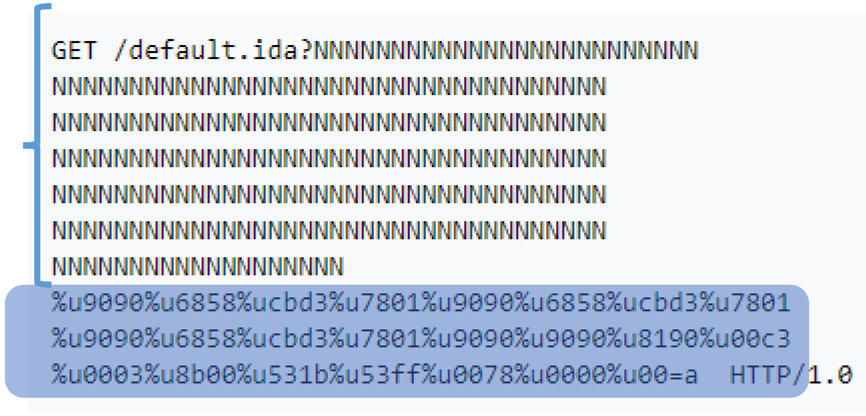
Day 28-end: Sleep

Worm example 1: Code Red (2001)

Exploits known buffer overflow in Microsoft IIS (web server)

Propagates through an HTTP request (TCP)

Achieve the overflow

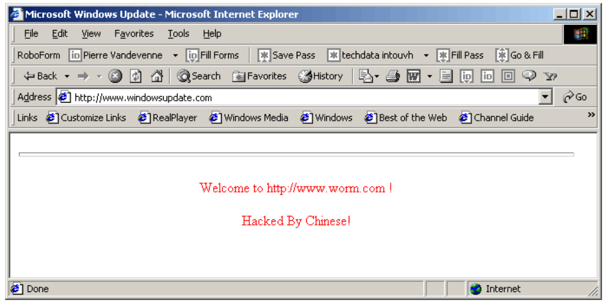


```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a  HTTP/1.0
```

Payload: injected executable code

Operation

1st Defacing:



http://www.windowsupdate.com

Welcome to http://www.worm.com !

Hacked By Chinese!

2nd Date dependent:

Day 1-19: continue spreading (random IPs)

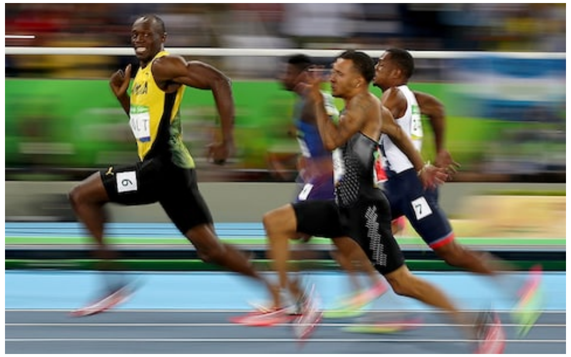
Day 20-27: Launch Denial of Service (among them White House)

Day 28-end: Sleep

Vulnerable population (360,000 servers) infected in 14 hours

Worm example 2: Slammer (2003)

Fastest worm ever

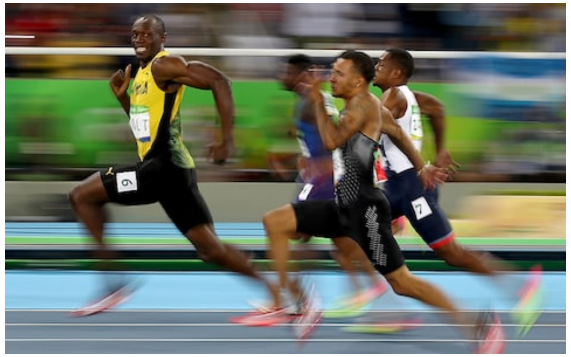


Exploits a vulnerability (buffer overflow) in Microsoft SQL Server

Creates random IPs and sends itself (small 380 byte UDP packet)

Worm example 2: Slammer (2003)

Fastest worm ever



Exploits a vulnerability (buffer overflow) in Microsoft SQL Server

Creates random IPs and sends itself (small 380 byte UDP packet)

75,000 victims infected within ten minutes

Doubled infections every 8.5 seconds

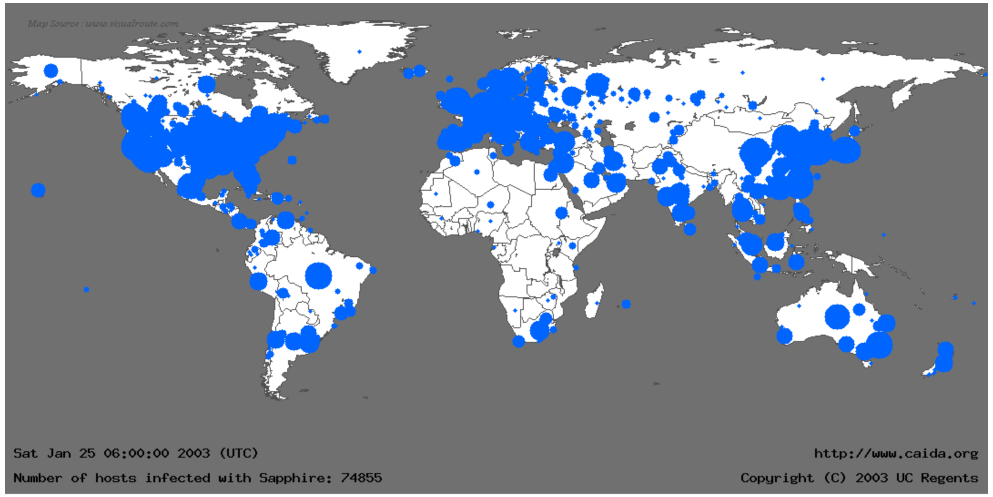
Consequences – Internet denied of service:

saturated many Internet links

routers collapsed

affected ATMs and emergency numbers

root DNS shut down


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

Worm example 2: Slammer (2003)

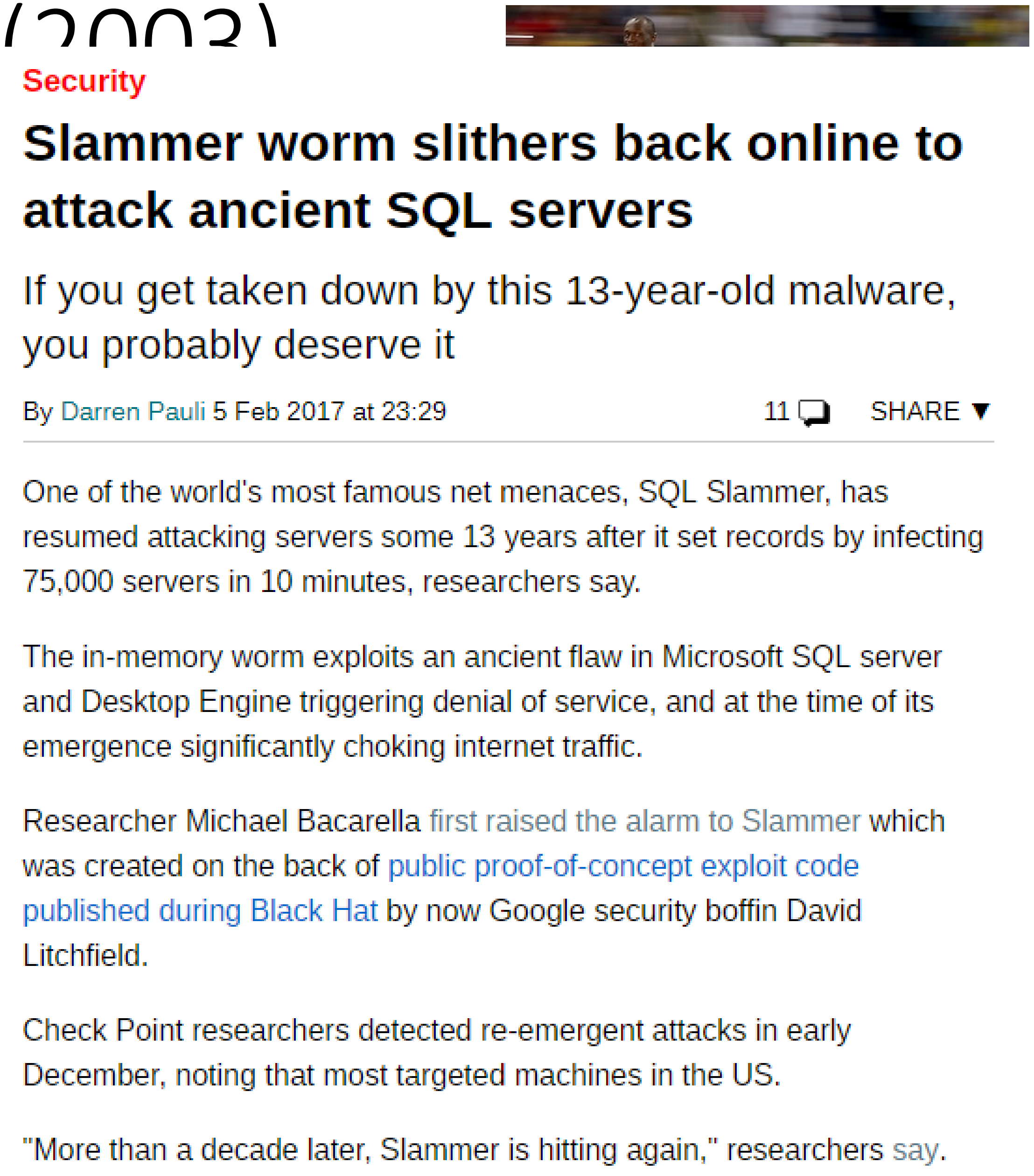
Fastest worm ever

Exploits a vulnerability (buffer overflow) in M

Creates random IPs and sends itself (small 38

75,000 victims infected within ten minutes
- Doubled infections every 8.5 seconds

Consequences – Internet denied of service:
- saturated many Internet links
- routers collapsed
- affected ATMs and emergency numbe
- root DNS shut down



Security

Slammer worm slithers back online to attack ancient SQL servers

If you get taken down by this 13-year-old malware, you probably deserve it

By Darren Pauli 5 Feb 2017 at 23:29 11 SHARE ▼

One of the world's most famous net menaces, SQL Slammer, has resumed attacking servers some 13 years after it set records by infecting 75,000 servers in 10 minutes, researchers say.

The in-memory worm exploits an ancient flaw in Microsoft SQL server and Desktop Engine triggering denial of service, and at the time of its emergence significantly choking internet traffic.

Researcher Michael Bacarella first raised the alarm to Slammer which was created on the back of public proof-of-concept exploit code published during Black Hat by now Google security boffin David Litchfield.

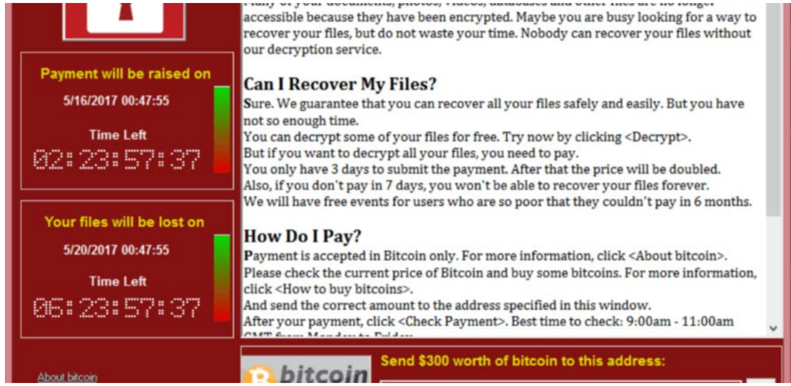
Check Point researchers detected re-emergent attacks in early December, noting that most targeted machines in the US.

"More than a decade later, Slammer is hitting again," researchers say.

Worm example 3: WannaCry (2017)

Most recent case of **Ransomware**

→ **Require money to recover system**



Payment will be raised on
5/16/2017 00:47:55
Time Left
02:23:57:37

Your files will be lost on
5/20/2017 00:47:55
Time Left
06:23:57:37

About bitcoin

Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am GMT from Monday to Friday.

bitcoin

Send $300 worth of bitcoin to this address:

Exploits a vulnerability revealed in a NSA hacking toolkit leak

- Mishandled packets for the Microsoft Server Message Block (protocol for shared access) enable arbitrary code execution
- The leak contained vulnerabilities in systems from e.g., Cisco Systems and Fortinet Inc

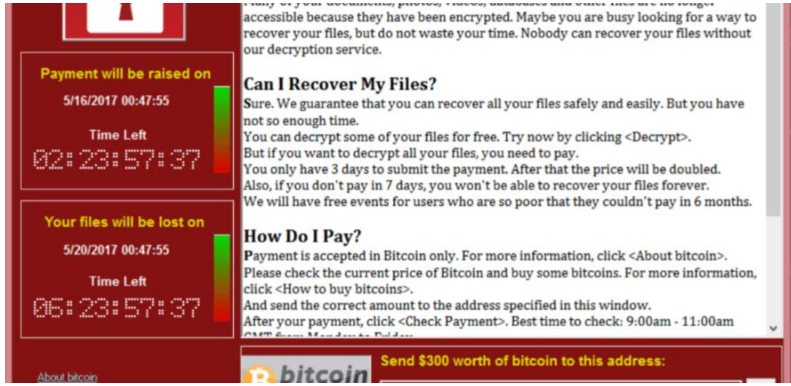
Encrypted data and asked for ransom in Bitcoins

- 300$ in 3 days or 600$ in 7 days or **DELETE**

Worm example 3: WannaCry (2017)

Most recent case of **Ransomware**

└──→ **Require money to recover system**



Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am GMT from Monday to Friday.

Payment will be raised on
5/16/2017 00:47:55
Time Left
02:23:57:37

Your files will be lost on
5/20/2017 00:47:55
Time Left
06:23:57:37

About bitcoin

bitcoin

Send $300 worth of bitcoin to this address:

Exploits a vulnerability revealed in a NSA hacking toolkit leak

- Mishandled packets for the Microsoft Server Message Block (protocol for shared access) enable arbitrary code execution
- The leak contained vulnerabilities in systems from e.g., Cisco Systems and Fortinet Inc

Encrypted data and asked for ransom in Bitcoins

- 300$ in 3 days or 600$ in 7 days or **DELETE**

WHAT IS **BITCOIN**?

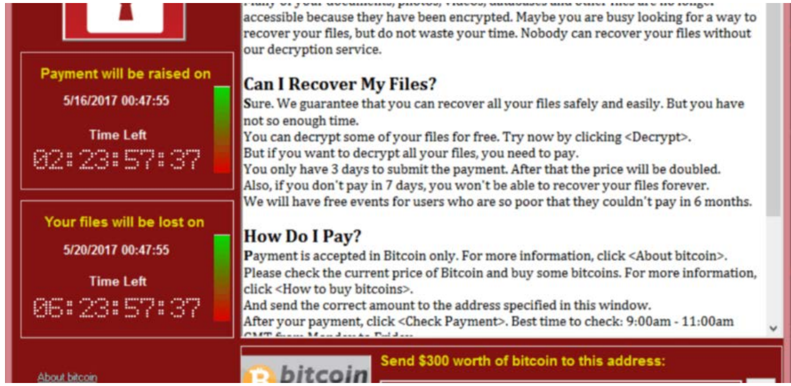
Decentralized currency
Anonymous payments

Works based on separation of privilege!!

Worm example 3: WannaCry (2017)

Most recent case of **Ransomware**

└──→ **Require money to recover system**



Payment will be raised on

5/16/2017 00:47:55

Time Left

02:23:57:37

Your files will be lost on

5/20/2017 00:47:55

Time Left

06:23:57:37

About bitcoin

accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

Send $300 worth of bitcoin to this address:

Exploits a vulnerability revealed in a NSA hacking toolkit leak

- Mishandled packets for the Microsoft Server Message Block (protocol for shared access) enable arbitrary code execution
- The leak contained vulnerabilities in systems from e.g., Cisco Systems and Fortinet Inc

Encrypted data and asked for ransom in Bitcoins

- 300$ in 3 days or 600$ in 7 days or **DELETE**

WHAT IS **BITCOIN**?

Decentralized currency

Anonymous payments

Works based on separation of privilege!!

>200,000 victims

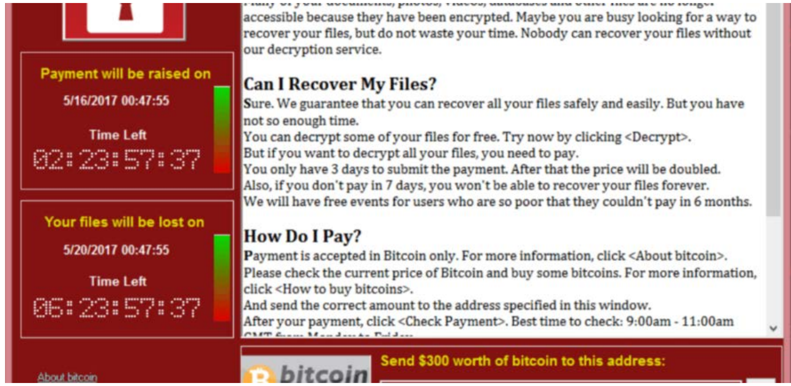
$130,634 obtained in ransom

billions of dollars in damage, UK Hospitals affected

Worm example 3: WannaCry (2017)

Most recent case of **Ransomware**

└→ **Require money to recover system**



Payment will be raised on

5/16/2017 00:47:55

Time Left

02:23:57:37

Your files will be lost on

5/20/2017 00:47:55

Time Left

06:23:57:37

About bitcoin

Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am GMT from Monday to Friday.

Send $300 worth of bitcoin to this address:

How did it end?

The worm "kill switch" was found

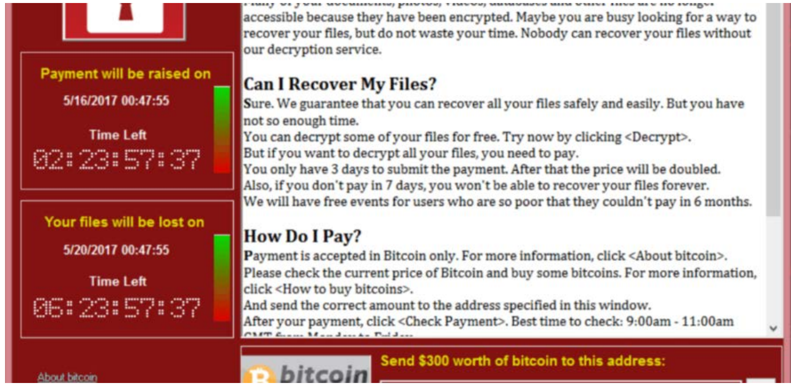
- Upon installation, the malware checked existence of a web. If yes, it stopped.
- A researcher registered the website and the worm stopped

Why have a kill switch?

Worm example 3: WannaCry (2017)

Most recent case of **Ransomware**

└→ **Require money to recover system**



Payment will be raised on

5/16/2017 00:47:55

Time Left

02:23:57:37

Your files will be lost on

5/20/2017 00:47:55

Time Left

06:23:57:37

About bitcoin

Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am GMT from Monday to Friday.

Send $300 worth of bitcoin to this address:

How did it end?

The worm "kill switch" was found

Upon installation, the malware checked existence of a web. If yes, it stopped.

A researcher registered the website and the worm stopped

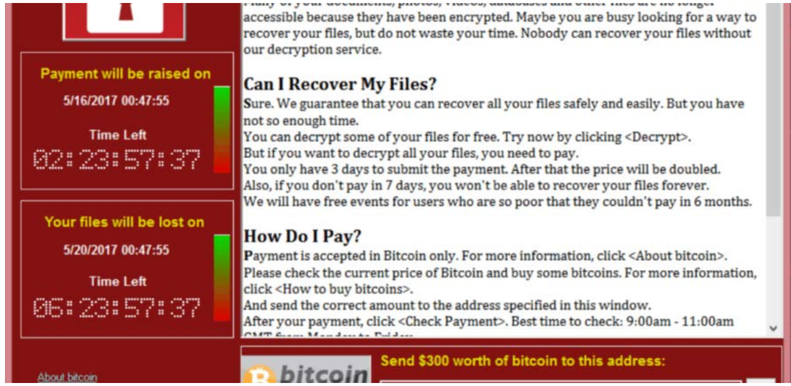
Why have a kill switch?

After the fact, hackers tried to use the **Mirai botnet** to DoS the domain and let the worm free again

Worm example 3: WannaCry (2017)

Most recent case of **Ransomware**

→ **Require money to recover system**



Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am GMT from Monday to Friday.

Send $300 worth of bitcoin to this address:

How did it end?

The worm "kill switch" was found

Upon installation, the malware checked existence of a web. If yes, it stopped.

A researcher registered the website and the worm stopped

Why have a kill switch?

Avoid worm study if hijacked, or if in sandbox

After the fact, hackers tried to use the **Mirai botnet** to DoS the domain and let the worm free again

Worm – defenses

Host-level

Protecting software from remote exploitation → **Attacks & Software security lecture!**

Stack protection techniques → **Software security lecture!**

Achieve diversity to increase protection → require more sophisticated worms

Antivirus (email-based Worms)

Network-level

Limit the number of outgoing connections: limit worm spreading

Personal firewall

e.g., block outgoing SMTP connections from unknown applications

Intrusion detection systems

Worm – defenses

Host-level

Protecting software from remote exploitation → **Attacks & Software security lecture!**

Stack protection techniques → **Software security lecture!**

Achieve diversity to increase protection → require more sophisticated worms

Antivirus (email-based Worms)

Heterogeneous systems
Different OS
Different programs
Different interfaces…

Network-level

Limit the number of outgoing connections: limit worm spreading

Personal firewall

- e.g., block outgoing SMTP connections from unknown applications

Intrusion detection systems

Worm – defenses

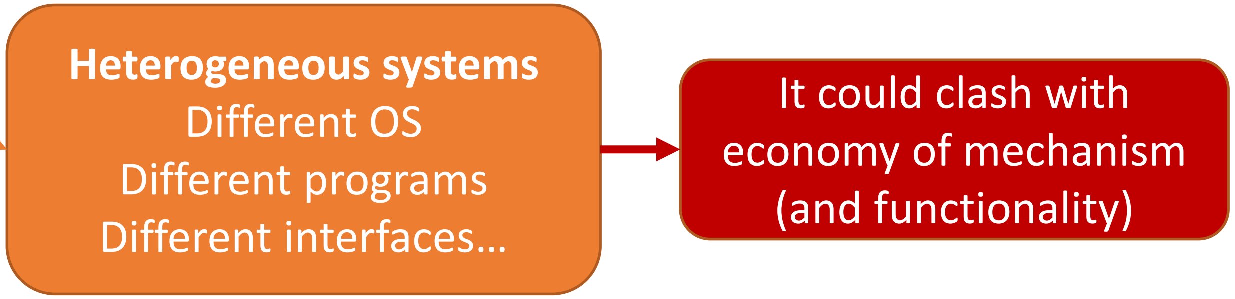
Host-level

Protecting software from remote exploitation → **Attacks & Software security lecture!**

Stack protection techniques → **Software security lecture!**

Achieve diversity to increase protection → require more sophisticated worms

Antivirus (email-based Worms)



Heterogeneous systems
Different OS
Different programs
Different interfaces…

It could clash with economy of mechanism (and functionality)

Network-level

Limit the number of outgoing connections: limit worm spreading

Personal firewall

- e.g., block outgoing SMTP connections from unknown applications

Intrusion detection systems

Intrusion detection systems – what they do

Host-based vs. Network-based

Host: process running on a host. Detects local malware

Network: network appliance monitoring all traffic

Signature based vs. Anomaly-based detection

Signature: identifies known patterns

+ low false alarms

- expensive (need up-to-date signatures), can't find new attacks

Anomaly: attempts to identify behavior different than legitimate

+ adapt to new attacks (legitimate does not change!)

- high false alarms

Trojan Horse

Malware that *appears to perform a desirable function* but it also performs **undisclosed malicious activities**

Requires users to **explicitly** run the program

Cannot replicate but can do any malicious activity

Spy on sensitive user data (spyware)

Allow remote access (backdoor)

Base for further attacks→ act as mail relay (for spammers)

Damage routines (corrupting files)

Trojan Horse

Malware that *appears to perform a desirable function* but it also performs **undisclosed malicious activities**

Requires users to **explicitly** run the program

Defense: Train users!

and follow least privilege principle!

Cannot replicate but can do any malicious activity

Spy on sensitive user data (spyware)

Allow remote access (backdoor)

Base for further attacks→ act as mail relay (for spammers)

Damage routines (corrupting files)

Trojan Horse examples:
Tiny Banker Trojan (2012)
Gameover Zeus (2013)

Goal: steal users sensitive data, such as account login information and banking codes.

Mode of Operation 1

1. Sniff packets to learn when a user visits **a banking website**
2. Steal credentials before they are sent → send to malware server

Mode of Operation 2

1. Sniff packets to learn when a user visits **a banking website**
2. Steal appearance from website
3. Ask questions to user on a pop-up → send answer to malware server

Trojan Horse examples:
Tiny Banker Trojan (2012)
Gameover Zeus (2013)

Goal: steal users sensitive data, such as account login information and banking codes.

Mode of Operation 1

1. Sniff packets to learn when a user visits **a banking website**
2. Steal credentials before they are sent → send to malware server

    Reads keystrokes before encryption!!

Mode of Operation 2

1. Sniff packets to learn when a user visits **a banking website**
2. Steal appearance from website
3. Ask questions to user on a pop-up → send answer to malware server

Rootkit

Adversary controlled code that takes residence **deep within the TCB** of a system

- Hides his presence by modifying the OS

Installed by an attacker **after** a system has been compromised

- Difficult to detect
  - Replace system programs with trojaned versions
  - Modify kernel data structures to hide processes, files, and network activities
- Allows the adversary to return on a later time

Rootkit

Adversary controlled code that takes residence **deep within the TCB** of a system

- Hides his presence by modifying the OS

Installed by an attacker **after** a system has been compromised

- Difficult to detect
  - Replace system programs with trojaned versions
  - Modify kernel data structures to hide processes, files, and network activities
- Allows the adversary to return on a later time

Defense (difficult!): Integrity checkers user/kernel level

Rootkit+Worm example: Stuxnet (2010)

Goal: Attack SCADA (Control systems) of Iran’s nuclear power plants

Three modules:

- Worm: spread Stuxnet’s payload
- Link file: executed malicious code
- Rootkit: hide the presence of malicious file to avoid detection

Rootkit+Worm example: Stuxnet (2010)

Goal: Attack SCADA (Control systems) of Iran’s nuclear power plants

Three modules:

Worm: spread Stuxnet’s payload

Link file: executed malicious code

Rootkit: hide the presence of malicious file to avoid detection

Stuxnet needs to be in the network, but the network is closed and disconnected

→ Entered via infected USB

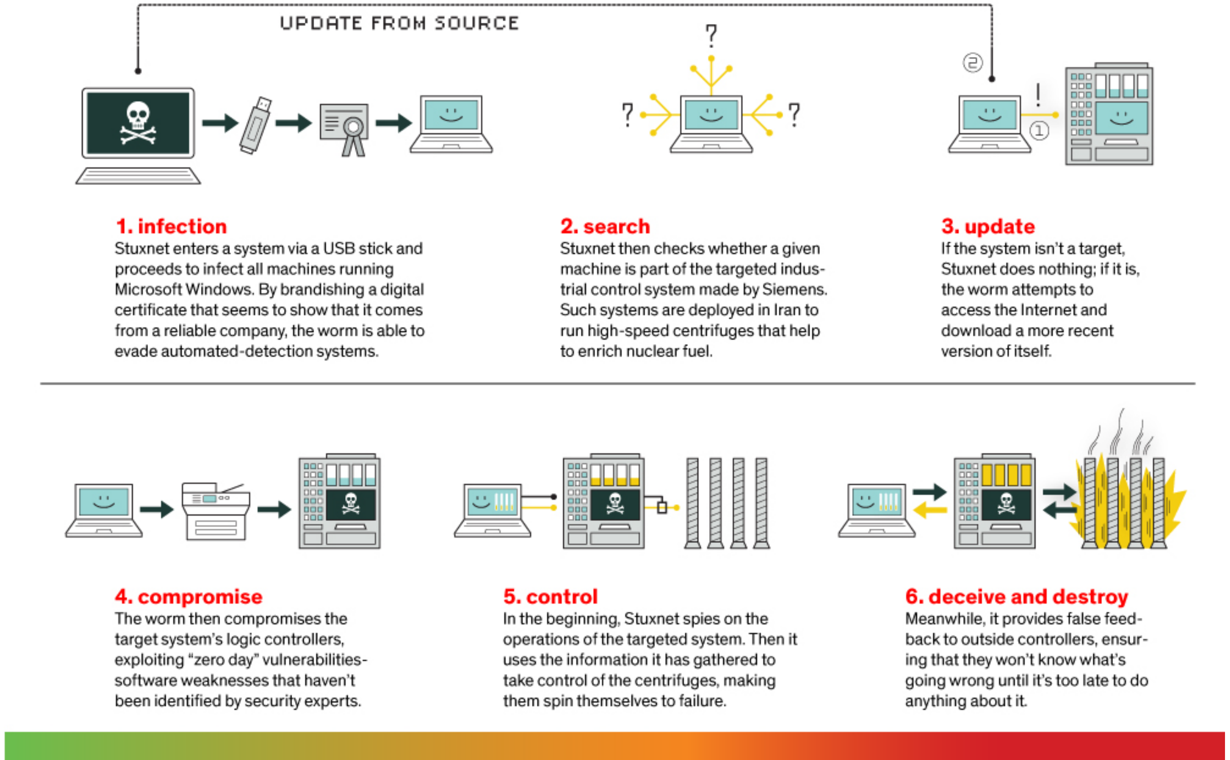
Rootkit+Worm example: Stuxnet (2010)

Goal: Attack SCADA (Control systems) of Iran’s nuclear power plants

Three modules:

- Worm: spread Stuxnet’s payload
- Link file: executed malicious code
- Rootkit: hide the presence of malicious file to avoid detection

Stuxnet needs to be in the network, but the network is closed and disconnected
→ Entered via infected USB



1. infection
Stuxnet enters a system via a USB stick and proceeds to infect all machines running Microsoft Windows. By brandishing a digital certificate that seems to show that it comes from a reliable company, the worm is able to evade automated-detection systems.

2. search
Stuxnet then checks whether a given machine is part of the targeted industrial control system made by Siemens. Such systems are deployed in Iran to run high-speed centrifuges that help to enrich nuclear fuel.

3. update
If the system isn’t a target, Stuxnet does nothing; if it is, the worm attempts to access the Internet and download a more recent version of itself.

4. compromise
The worm then compromises the target system’s logic controllers, exploiting “zero day” vulnerabilities-software weaknesses that haven’t been identified by security experts.

5. control
In the beginning, Stuxnet spies on the operations of the targeted system. Then it uses the information it has gathered to take control of the centrifuges, making them spin themselves to failure.

6. deceive and destroy
Meanwhile, it provides false feedback to outside controllers, ensuring that they won’t know what’s going wrong until it’s too late to do anything about it.

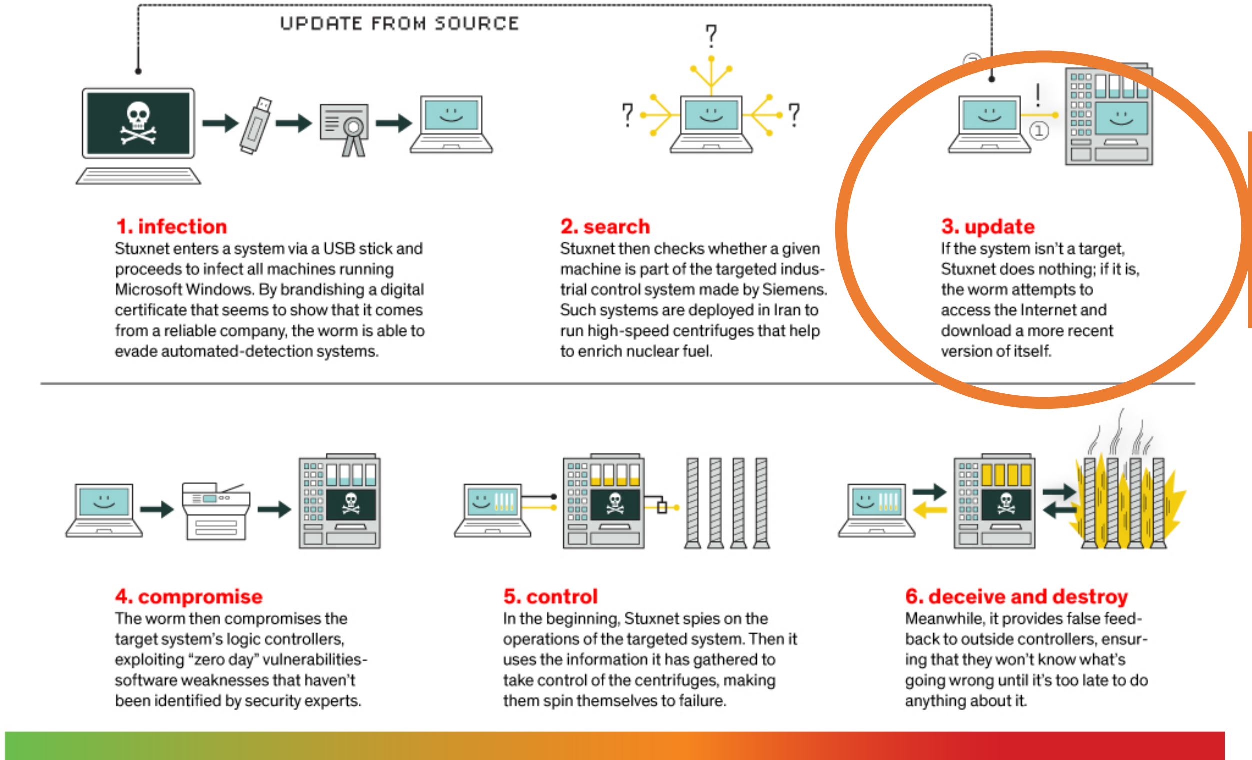
Rootkit+Worm example: Stuxnet (2010)

Goal: Attack SCADA (Control systems) of Iran’s nuclear power plants

Three modules:

- Worm: spread Stuxnet’s payload
- Link file: executed malicious code
- Rootkit: hide the presence of malicious file to avoid detection

Stuxnet needs to be in the network, but the network is closed and disconnected
→ Entered via infected USB



1. infection
Stuxnet enters a system via a USB stick and proceeds to infect all machines running Microsoft Windows. By brandishing a digital certificate that seems to show that it comes from a reliable company, the worm is able to evade automated-detection systems.

2. search
Stuxnet then checks whether a given machine is part of the targeted industrial control system made by Siemens. Such systems are deployed in Iran to run high-speed centrifuges that help to enrich nuclear fuel.

3. update
If the system isn’t a target, Stuxnet does nothing; if it is, the worm attempts to access the Internet and download a more recent version of itself.

4. compromise
The worm then compromises the target system’s logic controllers, exploiting “zero day” vulnerabilities-software weaknesses that haven’t been identified by security experts.

5. control
In the beginning, Stuxnet spies on the operations of the targeted system. Then it uses the information it has gathered to take control of the centrifuges, making them spin themselves to failure.

6. deceive and destroy
Meanwhile, it provides false feedback to outside controllers, ensuring that they won’t know what’s going wrong until it’s too late to do anything about it.

Very targeted attack. If the system infected is not a target, the malware stays dormant.

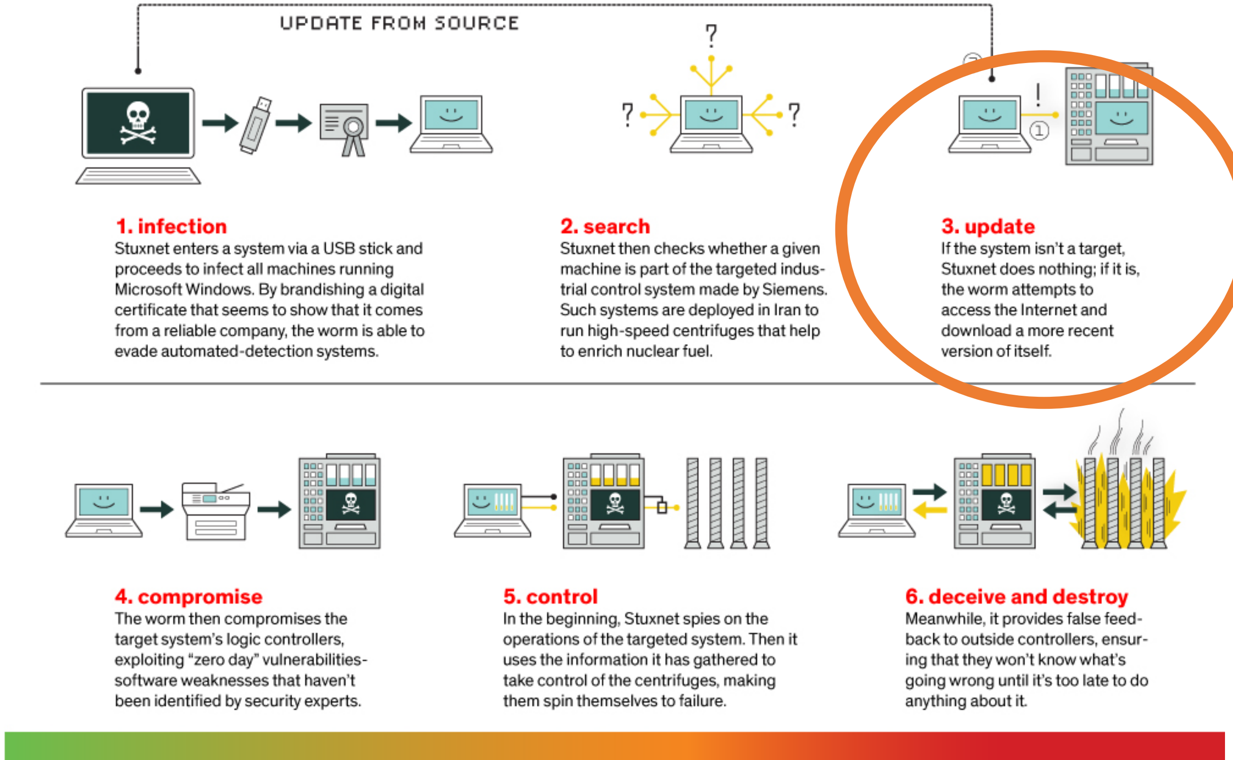
Rootkit+Worm example: Stuxnet (2010)

Goal: Attack SCADA (Control systems) of Iran’s nuclear power plants

Three modules:

- Worm: spread Stuxnet’s payload
- Link file: executed malicious code
- Rootkit: hide the presence of malicious file to avoid detection

Stuxnet needs to be in the network, but the network is closed and disconnected
→ Entered via infected USB



1. infection
Stuxnet enters a system via a USB stick and proceeds to infect all machines running Microsoft Windows. By brandishing a digital certificate that seems to show that it comes from a reliable company, the worm is able to evade automated-detection systems.

2. search
Stuxnet then checks whether a given machine is part of the targeted industrial control system made by Siemens. Such systems are deployed in Iran to run high-speed centrifuges that help to enrich nuclear fuel.

3. update
If the system isn’t a target, Stuxnet does nothing; if it is, the worm attempts to access the Internet and download a more recent version of itself.

4. compromise
The worm then compromises the target system’s logic controllers, exploiting “zero day” vulnerabilities-software weaknesses that haven’t been identified by security experts.

5. control
In the beginning, Stuxnet spies on the operations of the targeted system. Then it uses the information it has gathered to take control of the centrifuges, making them spin themselves to failure.

6. deceive and destroy
Meanwhile, it provides false feedback to outside controllers, ensuring that they won’t know what’s going wrong until it’s too late to do anything about it.

Very targeted attack. If the system infected is not a target, the malware stays dormant.

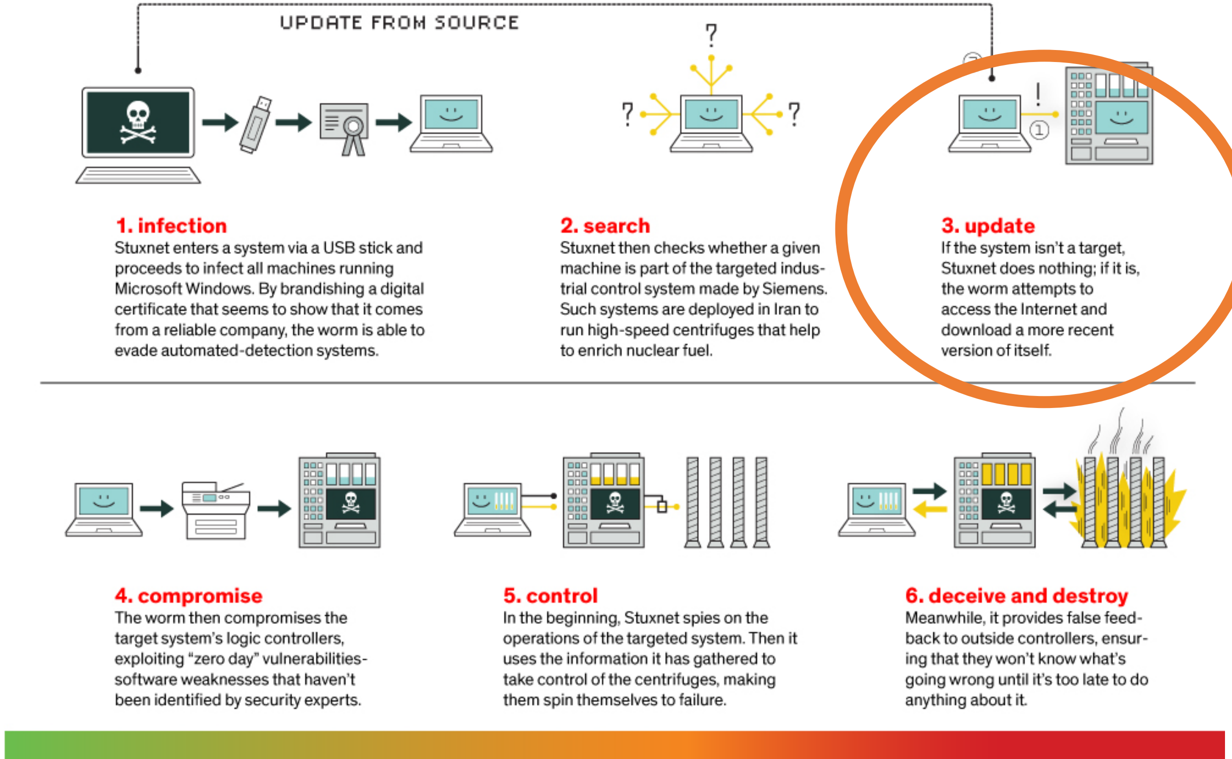
Rootkit+Worm example: Stuxnet (2010)

Goal: Attack SCADA (Control systems) of Iran’s nuclear power plants

Three modules:

- Worm: spread Stuxnet’s payload
- Link file: executed malicious code
- Rootkit: hide the presence of malicious file to avoid detection

Stuxnet needs to be in the network, but the network is closed and disconnected
→ Entered via infected USB



1. infection
Stuxnet enters a system via a USB stick and proceeds to infect all machines running Microsoft Windows. By brandishing a digital certificate that seems to show that it comes from a reliable company, the worm is able to evade automated-detection systems.

2. search
Stuxnet then checks whether a given machine is part of the targeted industrial control system made by Siemens. Such systems are deployed in Iran to run high-speed centrifuges that help to enrich nuclear fuel.

3. update
If the system isn’t a target, Stuxnet does nothing; if it is, the worm attempts to access the Internet and download a more recent version of itself.

4. compromise
The worm then compromises the target system’s logic controllers, exploiting “zero day” vulnerabilities-software weaknesses that haven’t been identified by security experts.

5. control
In the beginning, Stuxnet spies on the operations of the targeted system. Then it uses the information it has gathered to take control of the centrifuges, making them spin themselves to failure.

6. deceive and destroy
Meanwhile, it provides false feedback to outside controllers, ensuring that they won’t know what’s going wrong until it’s too late to do anything about it.

Very targeted attack. If the system infected is not a target, the malware stays dormant.

Authorship?

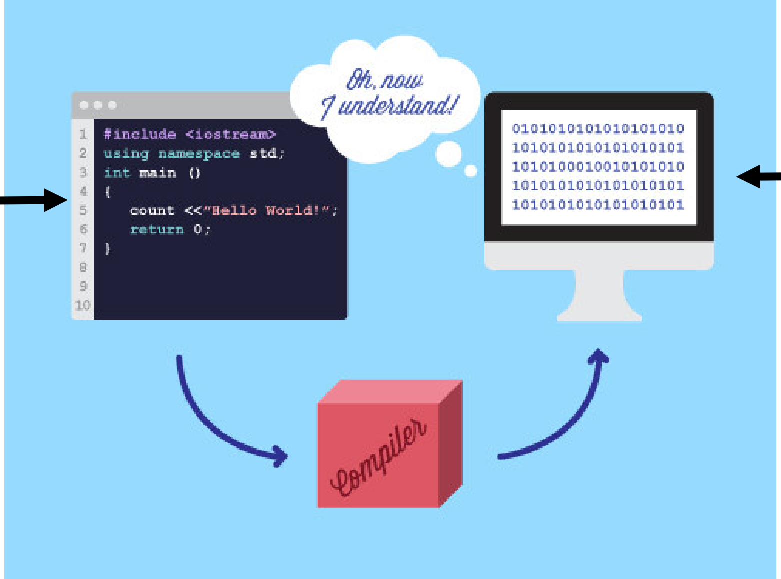
Alleged Israel/US Cyberweapon

https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet/

Cheatsheet: what is a compiler?

Cheatsheet: what is a compiler?

Computer software that transforms high-level code (basically any programming language) into low-level code that can be understood by the machine



```
#include <iostream>
using namespace std;
int main ()
{
    count <<"Hello World!";
    return 0;
}
```

```
0101010101010101010
1010101010101010101
1010100010010101010
1010101010101010101
1010101010101010101
```

Source code

Executable

Example compilers: gcc (C language), javac (Java language)

https://s3.amazonaws.com/kukuruku-co/uploads/images/59/03/5903d81e484b2.jpg

Backdoor

A **hidden** functionality that allows the adversary to bypass some security mechanism

Backdoor

A **hidden** functionality that allows the adversary to bypass some security mechanism

Why not “audit” the program?

- We can audit the program source

Backdoor

A **hidden** functionality that allows the adversary to bypass some security mechanism

Why not “audit” the program?

- We can audit the program source
  - what if the **compiler** is malicious and introduces backdoors?

Chain of reasoning leads us to suspect all programs down to the very first compiler!

Key paper: *Thompson, Ken. "Reflections on trusting trust." Communications of the ACM (1984)*

More readable summary: https://www.schneier.com/blog/archives/2006/01/countering_trus.html

Backdoor

How can we avoid blind trust on compilers? (How can we avoid trusting trust?)

Challenge: you have the executable of two compilers C1 (Exe_{C1}) and C2 (Exe_{C2}). You want to know if they are hiding a backdoor.

Backdoor

How can we avoid blind trust on compilers? (How can we avoid trusting trust?)

Challenge: you have the executable of two compilers C1 (Exe_{C1}) and C2 (Exe_{C2}). You want to know if they are hiding a backdoor.

1. Start with another compiler C_A source and compile it with the two compiler executables Exe_{C1} and Exe_{C2}

   Compile C_A with the first compiler: $Exe_{C1}(C_A) = Exe_{A,1}$

   Compile C_A with the second compiler: $Exe_{C2}(C_A) = Exe_{A,2}$

Backdoor

How can we avoid blind trust on compilers? (How can we avoid trusting trust?)

Challenge: you have the executable of two compilers C1 ($ExeC_1$) and C2 ($ExeC_2$). You want to know if they are hiding a backdoor.

2. Use the compiler executables $Exe_{A,1}$ and $Exe_{A,2}$ to compile C_A again

We expect that: $\mathbf{Exe_{A,1}(C_A)=Exe_{A,2}(C_A)}$ **Why?**

Backdoor

How can we avoid blind trust on compilers? (How can we avoid trusting trust?)

Challenge: you have the executable of two compilers C1 ($ExeC_1$) and C2 ($ExeC_2$). You want to know if they are hiding a backdoor.

2. Use the compiler executables $Exe_{A,1}$ and $Exe_{A,2}$ to compile C_A again

We expect that: $\mathbf{Exe_{A,1}(C_A)=Exe_{A,2}(C_A)}$ **Why?**

$Exe_{A,1}$ and $Exe_{A,2}$ are executables of the same compiler C_A**!**
Therefore given the same input they should output the same executable code

Backdoor

How can we avoid blind trust on compilers? (How can we avoid trusting trust?)

Challenge: you have the executable of two compilers C1 ($ExeC_1$) and C2 ($ExeC_2$). You want to know if they are hiding a backdoor.

2. Use the compiler executables $Exe_{A,1}$ and $Exe_{A,2}$ to compile C_A again

We expect that: $\mathbf{Exe_{A,1}(C_A)=Exe_{A,2}(C_A)}$ **Why?**

$Exe_{A,1}$ and $Exe_{A,2}$ are executables of the same compiler C_A**!**
Therefore given the same input they should output the same executable code

3. If not, one of them (or both) are introducing a backdoor!

https://dwheeler.com/trusting-trust/

Backdoor

What principle have we used here?

How can we avoid blind trust on compilers? (How can we avoid trusting trust?)

Challenge: you have the executable of two compilers C1 ($ExeC_1$) and C2 ($ExeC_2$). You want to know if they are hiding a backdoor.

1. Start with another compiler C_A source and compile it with the two compiler executables Exe_{C1} and Exe_{C2}

   $Exe_{C1}(C_A) = Exe_{A,1}$ $\quad\quad$ $Exe_{C2}(C_A) = Exe_{A,2}$

2. Use the compiler executables $Exe_{A,1}$ and $Exe_{A,2}$ to compile C_A again

   The two binaries should now be the same! **$Exec_{A,1}(C_A)$=$Exec_{A,2}(C_A)$?**

3. If not, one of them (or both) are introducing a backdoor!

https://dwheeler.com/trusting-trust/

Backdoor

What principle have we used here?

How can

Separation of privilege

Have two compilers → the adversary needs to subvert both in the same way!

rusting trust?)

Challeng and C2 (ExeC$_2$). You want to know if they are hiding a backdoor.

1. Start with another compiler C_A source and compile it with the two compiler executables Exe_{C1} and Exe_{C2}

   $Exe_{C1}(C_A) = Exe_{A,1}$ $Exe_{C2}(C_A) = Exe_{A,2}$

2. Use the compiler executables $Exe_{A,1}$ and $Exe_{A,2}$ to compile C_A again

   The two binaries should now be the same! **$Exec_{A,1}(C_A)=Exec_{A,2}(C_A)$?**

3. If not, one of them (or both) are introducing a backdoor!

Botnets

Attacks at scale!!

Multiple (millions) compromised **hosts** under the control of a single entity

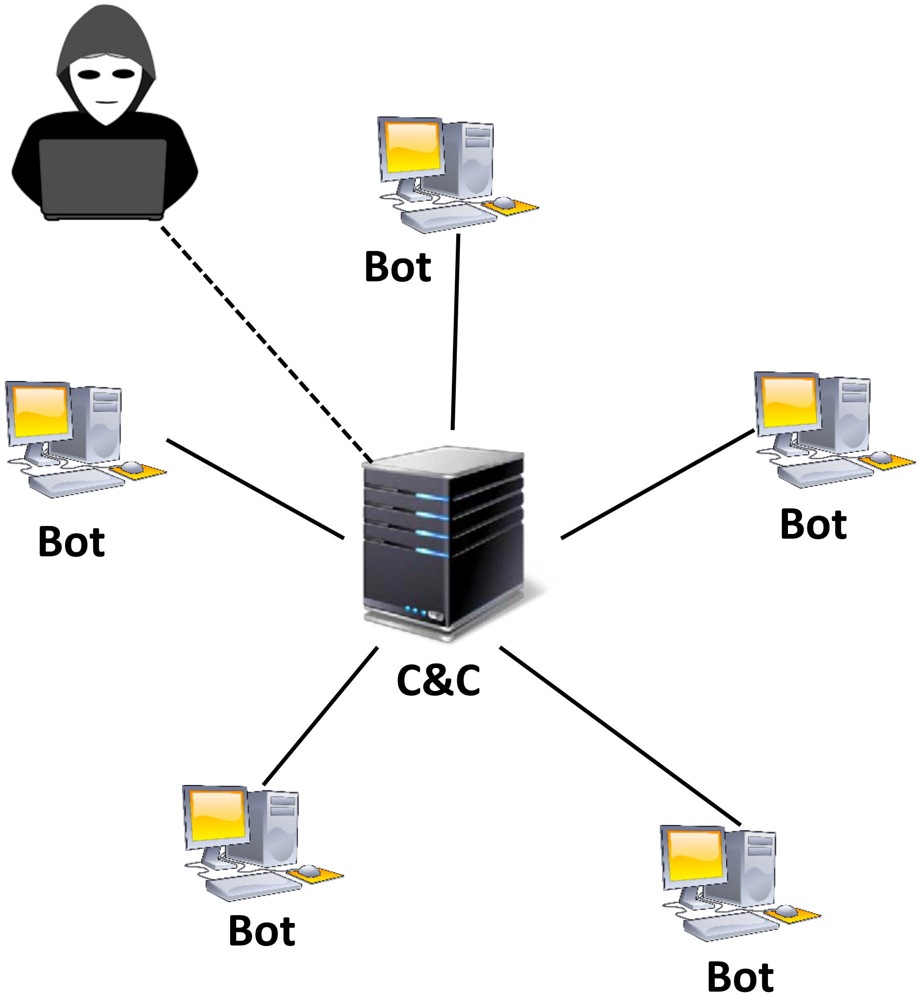
"zombies" or "bots"

uses

Bot-net command & control (C&C)

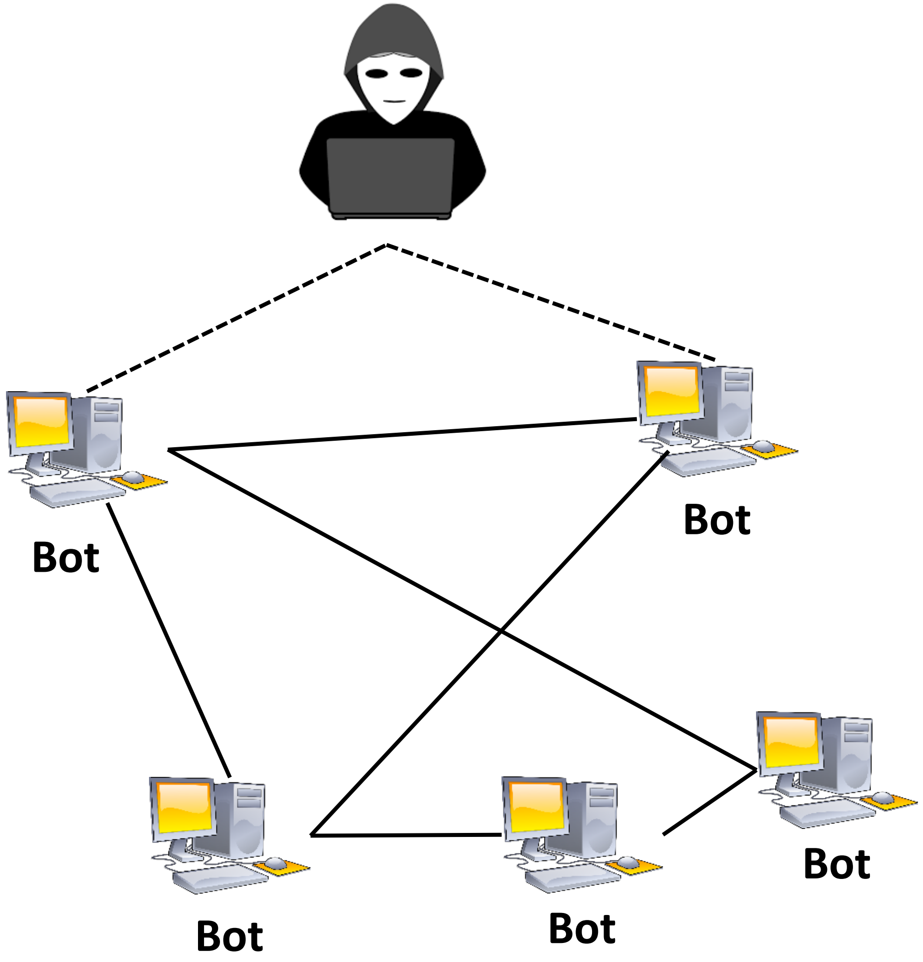
System to keep track of bots and send commands to them

Botnets - Star Topology



What is the problem here?

Botnets – P2P Topology

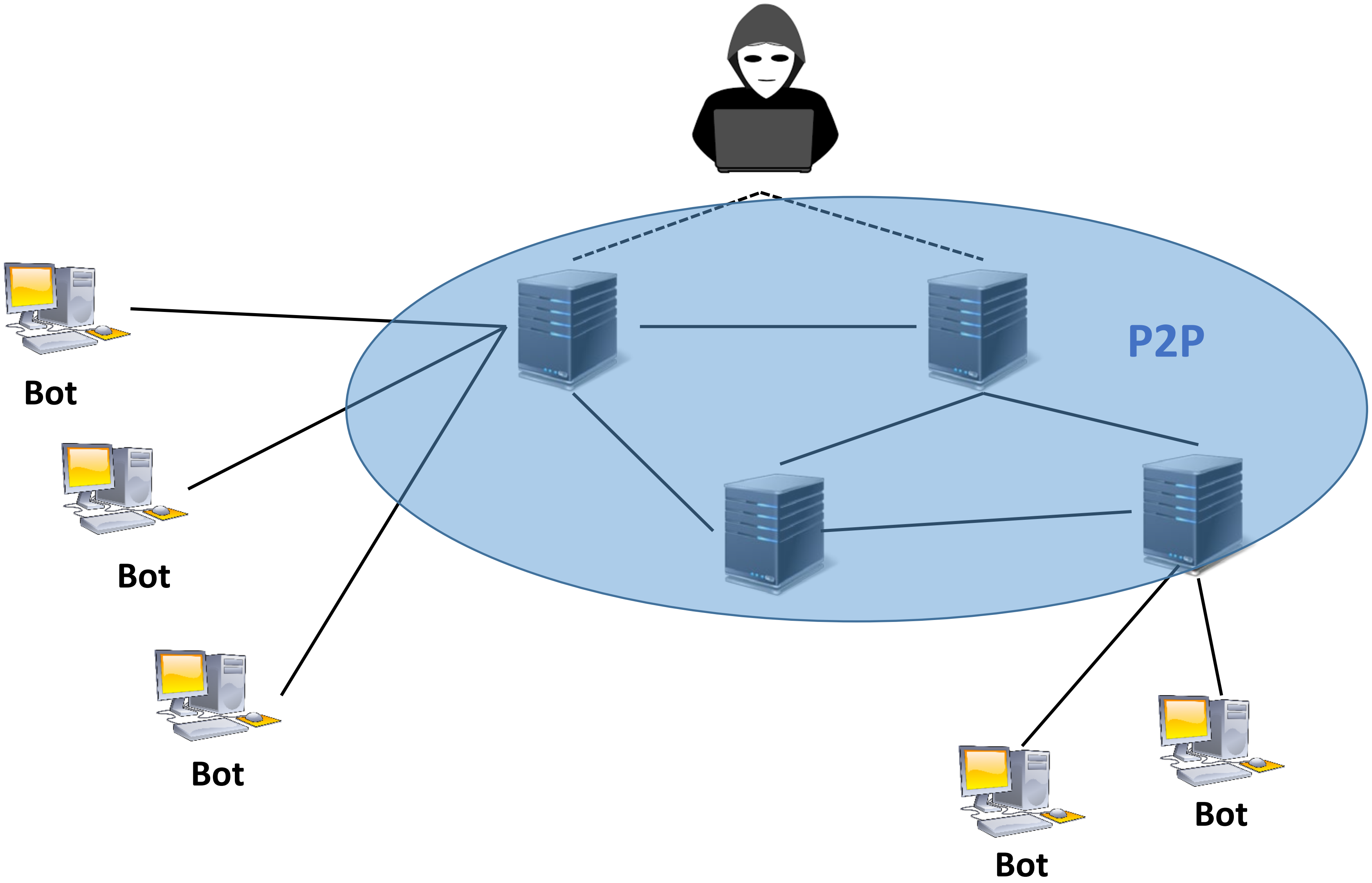


No Command and Control!!

Difficult management (join? leave?)

Vulnerable to attacks in which too many bots are taken over (these are called Sybil attacks)

Botnets – Hybrid



Monetizing Botnets

Rental – “Pay me money, and I’ll let you use my botnet…”

DDoS extortion – “Pay me or I take down you legitimate business”

Bulk traffic selling – “Pay me to boost visit counts on your website”

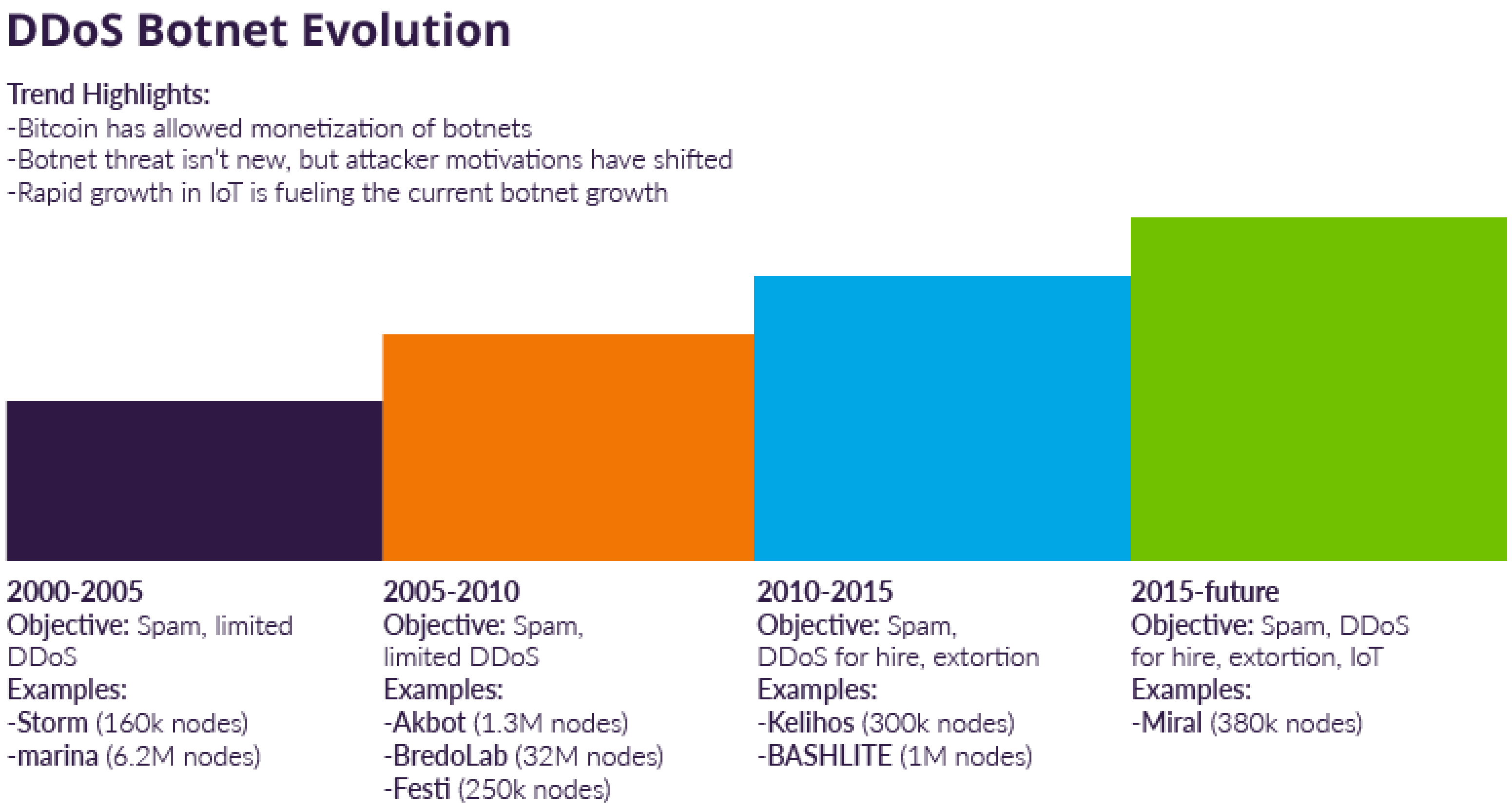
Click fraud – “Simulate clicks on advertised links to generate revenue”

Distribute Ransomware – “I’ve encrypted your hard drive, pay!”

Advertise products – “Pay me, I will leave comments all around the web”

Bitcoin mining!!

…



DDoS Botnet Evolution

Trend Highlights:
-Bitcoin has allowed monetization of botnets
-Botnet threat isn't new, but attacker motivations have shifted
-Rapid growth in IoT is fueling the current botnet growth

2000-2005
Objective: Spam, limited DDoS
Examples:
-**Storm** (160k nodes)
-**marina** (6.2M nodes)

2005-2010
Objective: Spam, limited DDoS
Examples:
-**Akbot** (1.3M nodes)
-**BredoLab** (32M nodes)
-**Festi** (250k nodes)

2010-2015
Objective: Spam, DDoS for hire, extortion
Examples:
-**Kelihos** (300k nodes)
-**BASHLITE** (1M nodes)

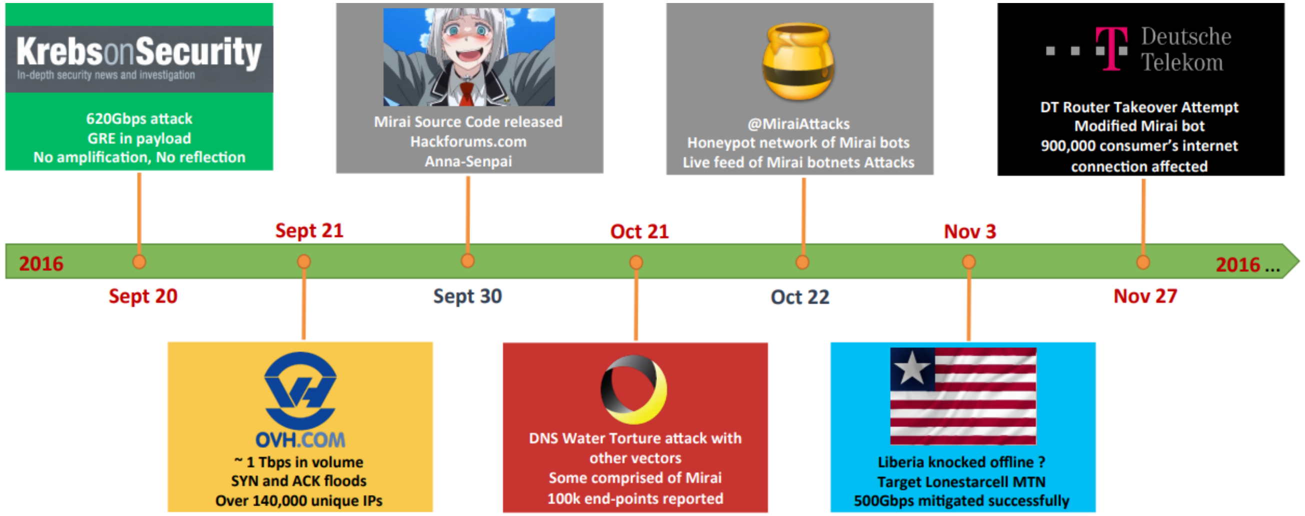
2015-future
Objective: Spam, DDoS for hire, extortion, IoT
Examples:
-**Miral** (380k nodes)

https://www.incapsula.com/blog/how-to-identify-a-mirai-style-ddos-attack.html

Example Botnet – Mirai (2016)

Target: IoT devices

scanning of Telnet ports, attempted to log in using 61 username/password combos



2016

Sept 20

KrebsonSecurity
In-depth security news and investigation
620Gbps attack
GRE in payload
No amplification, No reflection

Sept 21

OVH.COM
~ 1 Tbps in volume
SYN and ACK floods
Over 140,000 unique IPs

Sept 30

Mirai Source Code released
Hackforums.com
Anna-Senpai

Oct 21

DNS Water Torture attack with other vectors
Some comprised of Mirai
100k end-points reported

Oct 22

@MiraiAttacks
Honeypot network of Mirai bots
Live feed of Mirai botnets Attacks

Nov 3

Liberia knocked offline ?
Target Lonestarcell MTN
500Gbps mitigated successfully

Nov 27

Deutsche Telekom
DT Router Takeover Attempt
Modified Mirai bot
900,000 consumer's internet connection affected

2016 …

Open source code – variants appear all the time

Wicked (2018): scans ports 8080, 8443, 80, and 81 and attempts to locate vulnerable, unpatched IoT devices running on those ports.

https://www.nanog.org/sites/default/files/1_Winward_Mirai_The_Rise_v1.pdf
https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf

Botnets: defense

Attack C&C infrastructure

Take communication channel off-line

Hijack/poison DNS to route traffic to black hole

Honeypots

Vulnerable computer that serves no purpose other than to attract attackers and study their behavior in controlled environments

Study botnet behavior to find defense (or study ecosystem)

Other malware

Rabbit: code that replicates itself w/o limit to exhaust resources

Logic (time) bomb: code that triggers action when condition (time) occurs

Dropper: code that drops other malicious code

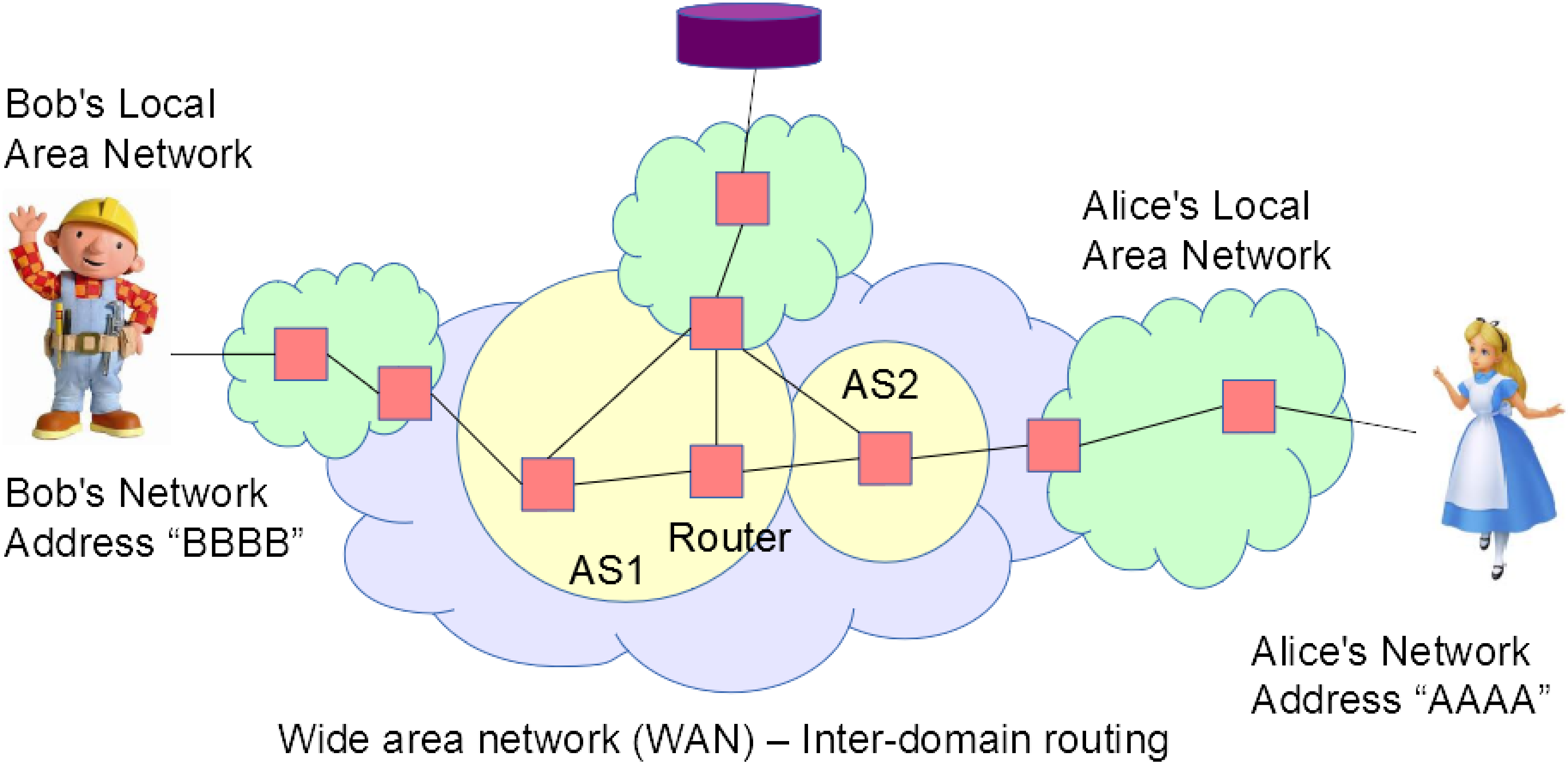
Tool/toolkit: program used to assemble malicious code (not malicious itself)

Scareware: false warning of malicious code attack

Up to here: attacks on hosts
What about the network?

Up to here: attacks on hosts
What about the network?

The network is not a tube!!!



Desired properties

Naming security

The association between lower level names (eg. network addresses) and higher level names (eg. Alice / Bob) must not be influenced by the adversary.

Routing security

The route over the network and the eventual delivery of messages must not be influenced by the adversary.

Session security

The association of messages within the same session, or their ordering, must be as intended by the communicating parties, and no more, less, or different messages should be associated with the session.

More on these after the midterm!!