

Computer Security (COM-301)

Malware

Introduction

Carmela Troncoso

SPRING Lab

carmela.troncoso@epfl.ch

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

Network attacks

Lack of security mechanisms in network protocols

'ARP/DNS/BGP Hijacking'

Change association of high-level and low-level addresses, names, routes

'Denial of Service'

Capability of adversaries to exhaust the resources of a victim by using flaws at the network or application layers

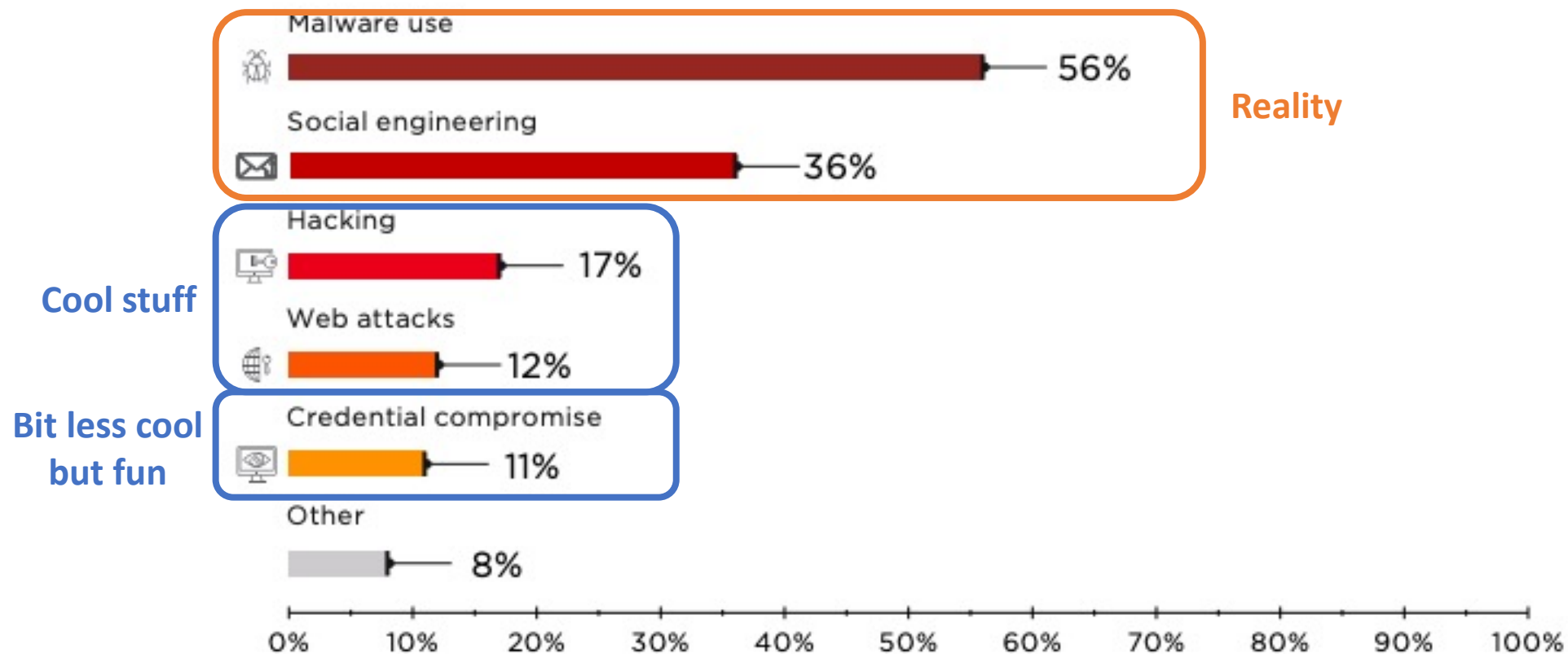
Expert adversary

requires deep understanding of computer systems and networks

"Manual" adversary

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





Malware

Short for “Malicious Software”

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

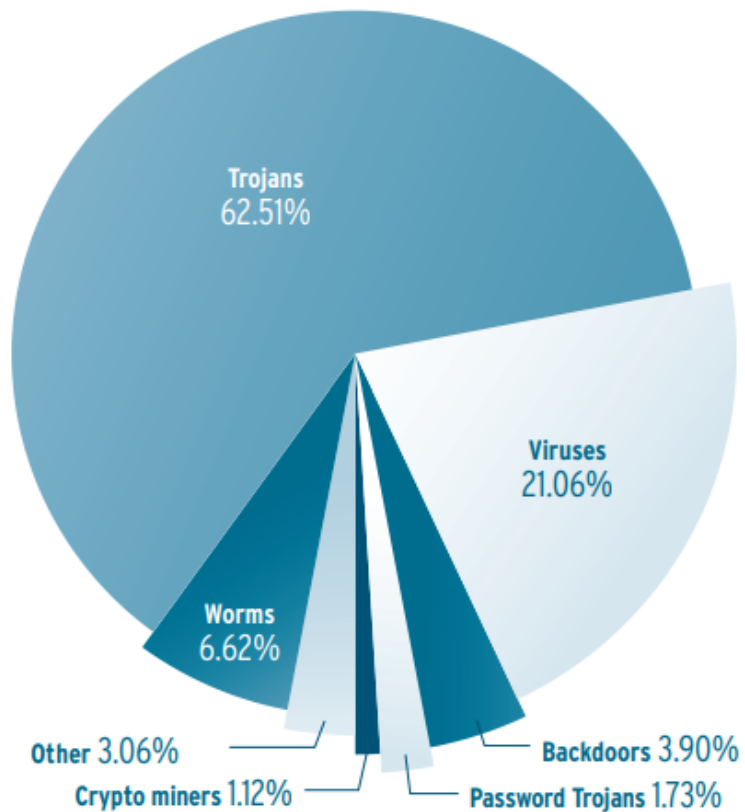
Intentionally written to cause adverse effects

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

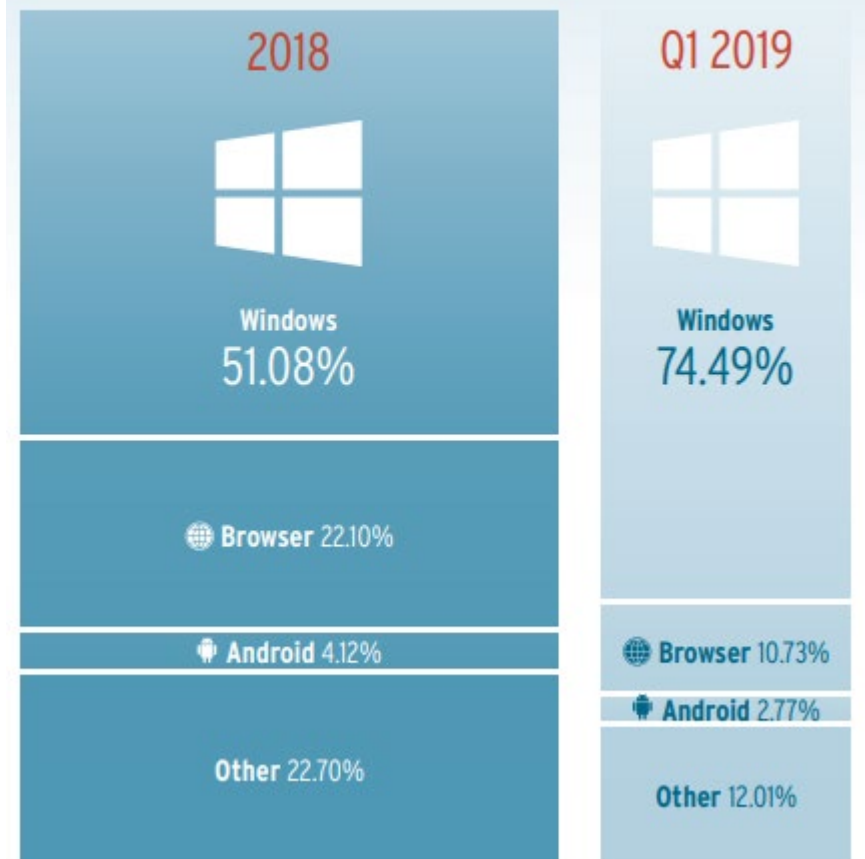
Malware != virus

Virus is a kind of Malware

Distribution of malware under Windows in 2018



Distribution of malware



Malware – why the rise?

Homogeneous computing base

Windows/Android make very tempting targets

Clueless user base

Many targets available

Unprecedented connectivity

Deploying remote / distributed attacks is increasingly easier

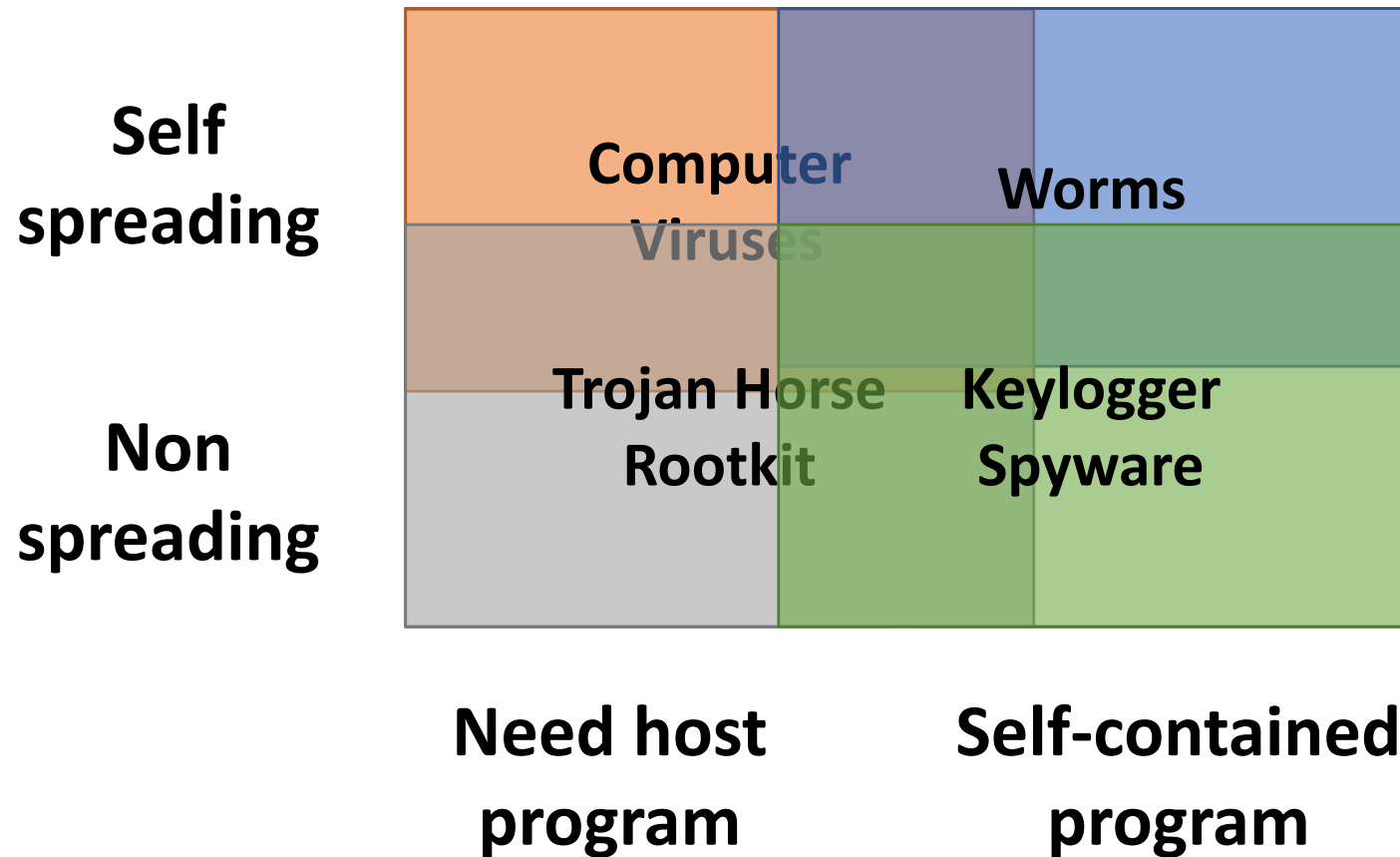
Malicious code has become profitable!

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

ATTACKER ENGINEERING PROCESS

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

Malware – taxonomy



Malware – taxonomy

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

The screenshot shows the Symantec Enterprise Security Center interface. The top navigation bar includes the Symantec logo, 'ENTERPRISE', and a search icon. Below this is a secondary navigation bar with links for 'Products & Services', 'Solutions', 'Support Center', 'Security Center' (which is underlined), 'Blogs', and 'Partner'. A breadcrumb trail shows 'Security Center / Threats'. The main content area has tabs for 'Overview', 'Threats' (which is selected), 'Risks', 'Vulnerabilities', and 'A-Z'. A large dark blue banner with the word 'Threats' in white is positioned below the tabs. Underneath the banner is a table of threats. The table has columns for 'Name', 'Type', and 'Protected*'. It lists 11 threats, with the first 10 visible. The table is paginated, showing 'Showing 1 to 10 of 11 entries' and a page number '1'.

Name	Type	Protected*
SONAR.SuspLaunchlg15	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.Advind!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

[DEPRECATED]

Lists of:

- Threats
- Vulnerabilities
- Risks

Computer Security (COM-301)

Malware

Types of Malware

Carmela Troncoso

SPRING Lab

carmela.troncoso@epfl.ch

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**

Viruses modify programs to include a (possibly modified) copy of themselves

Viruses **cannot** survive without the host

What are the permissions of a virus?

The same permissions as the host!

virus can do anything that the host program is permitted to do

virus **executes** secretly when the host program is run

The host program would be acting as... ?

Yes! The confused deputy again!

Recurring problem in security!

Mitigation: follow the **least privilege principle**!

Specific to operating system and hardware

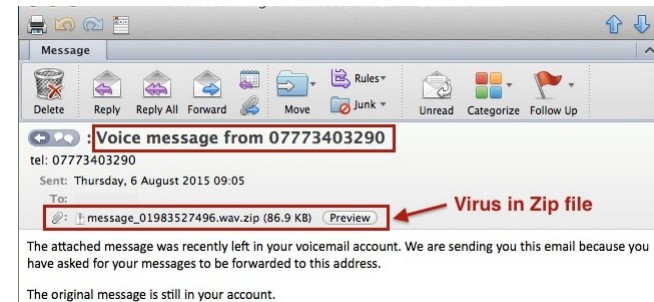
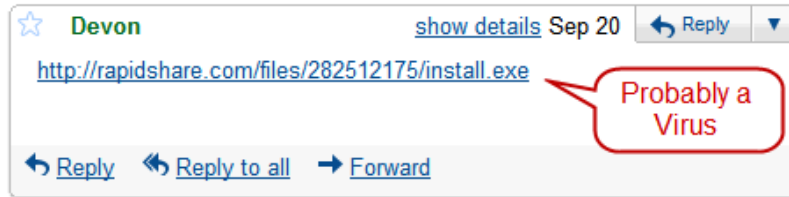
take advantage of their details and weaknesses

Virus (2)

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



WOW Inbox | X



email

web

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

Why do we even bother with security software?

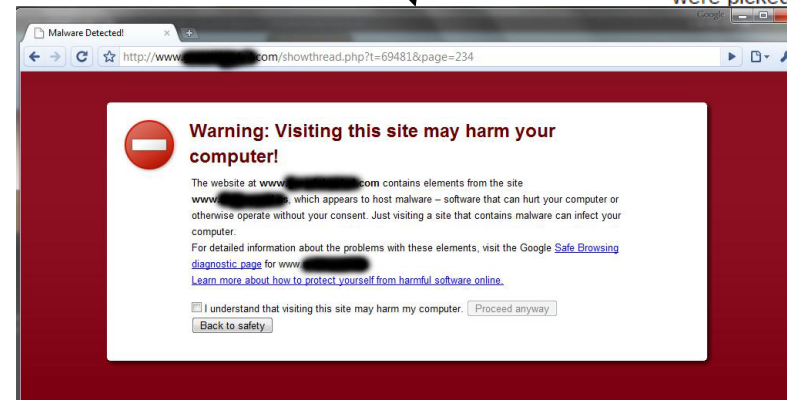
By Shaun Nichols in San Francisco 11 Apr 2016 at 21:09 115 SHARE

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 coerced into picking up and plugging in seemingly lost USB drives they find," the researchers reported this month.

whether driven by altruistic motives or human curiosity.



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)

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:

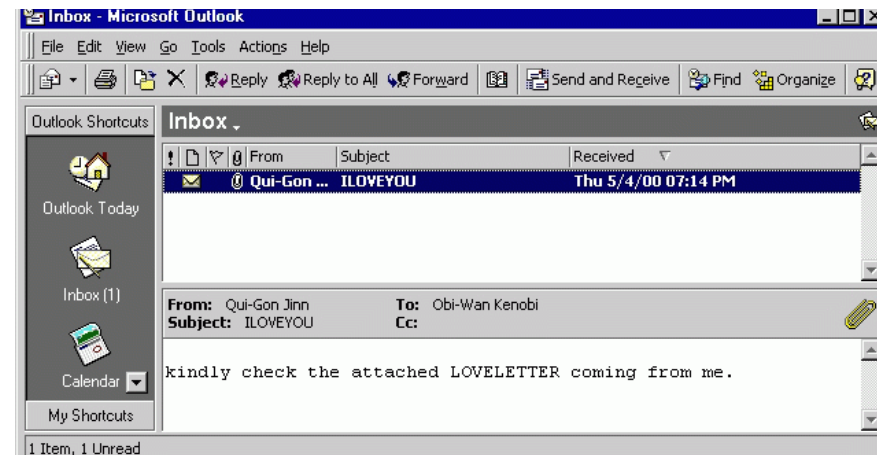
Replaced files with extensions JPG, JPEG, VBS, JS, DOC, ...

The script adds Windows Registry data for automatic startup on system boot

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

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

Damage: \$10 billion



Virus – defenses

Antivirus Software

Signature-based detection

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

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 / anomalies) 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

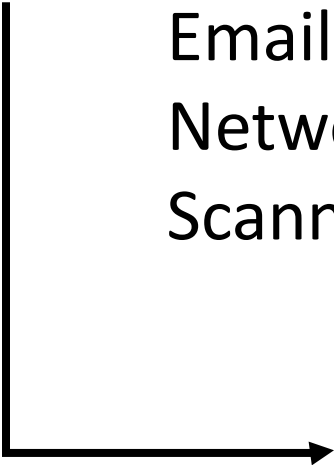
Does not need a host program to execute

Autonomous spread over the network

Email harvesting (address book, inbox, browser cache)

Network enumeration

Scanning (at random or targeted)



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

Network: automated!

Worm example - WannaCry (2017)

A case of **Ransomware**

→ **Require money to recover system**



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**

>200,000 victims

\$130,634 obtained in ransom

billions of dollars in damage, UK Hospitals affected

Worm example - WannaCry (2017)

A case of **Ransomware**

→ **Require money to recover system**



How did it end?

The worm “kill switch” was found

Upon installation, the malware checked the 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

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 number of 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

Defense: Train users!

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)

and follow least
privilege principle!

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**

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

Rootkit+Worm example: Stuxnet (2010)

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

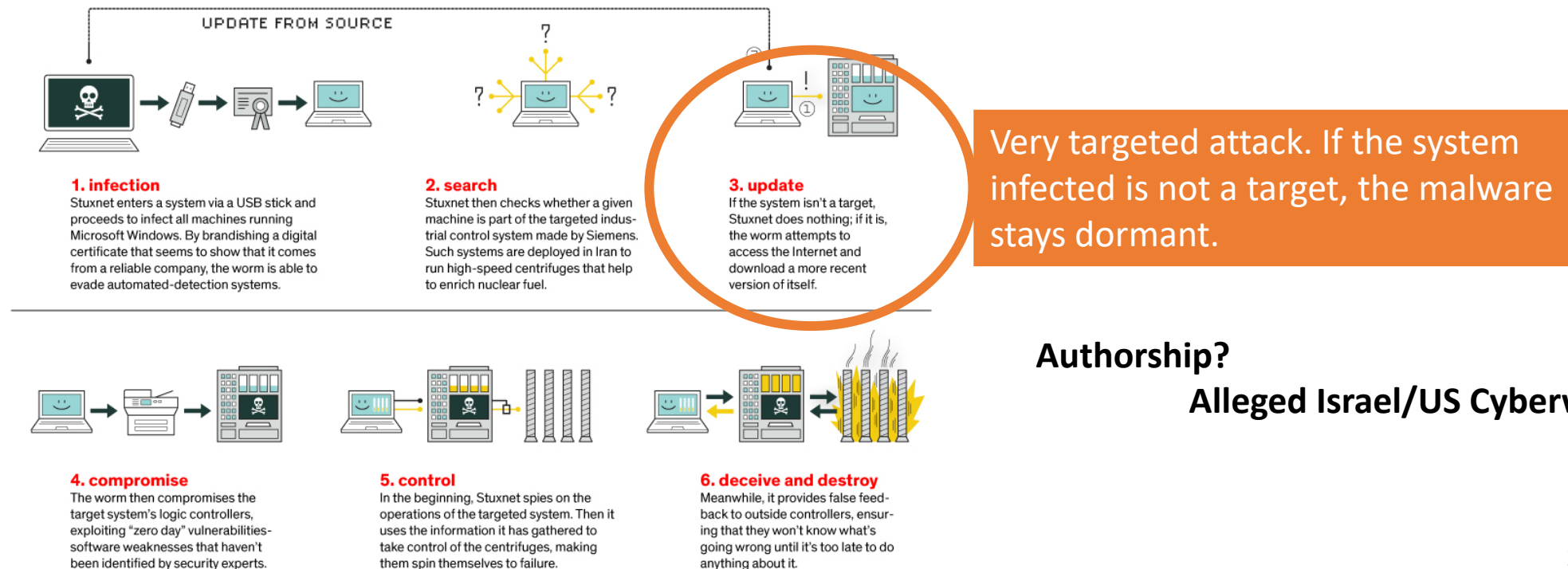
Three modules:

Worm: spread Stuxnet's payload

Link file: executed malicious code

Rootkit: hide the presence of malicious file to avoid detection

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



Authorship?

Alleged Israel/US Cyberweapon

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 27

Computer Security (COM-301)

Malware

Botnets

Carmela Troncoso

SPRING Lab

carmela.troncoso@epfl.ch

Botnets

Attacks at scale!!



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

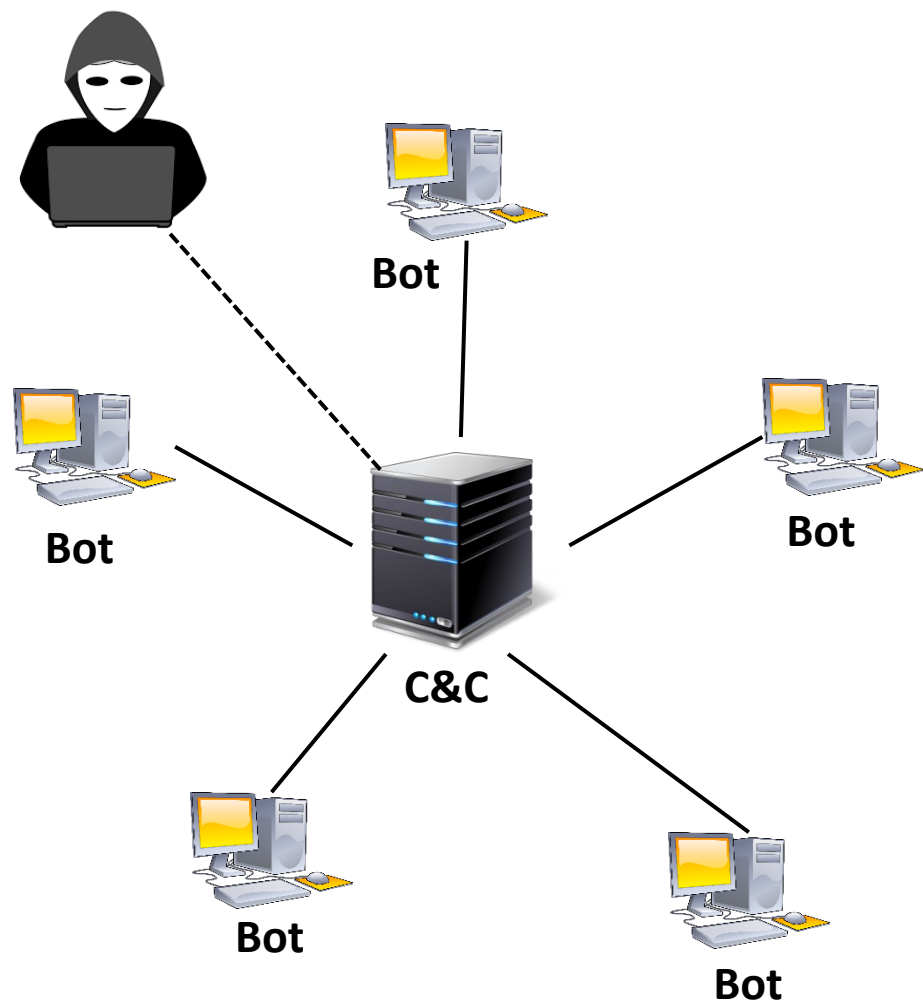
“zombies” or “bots”

uses

Bot-net command & control (C&C)

System to keep track of bots and send commands to them

Botnets - Star Topology

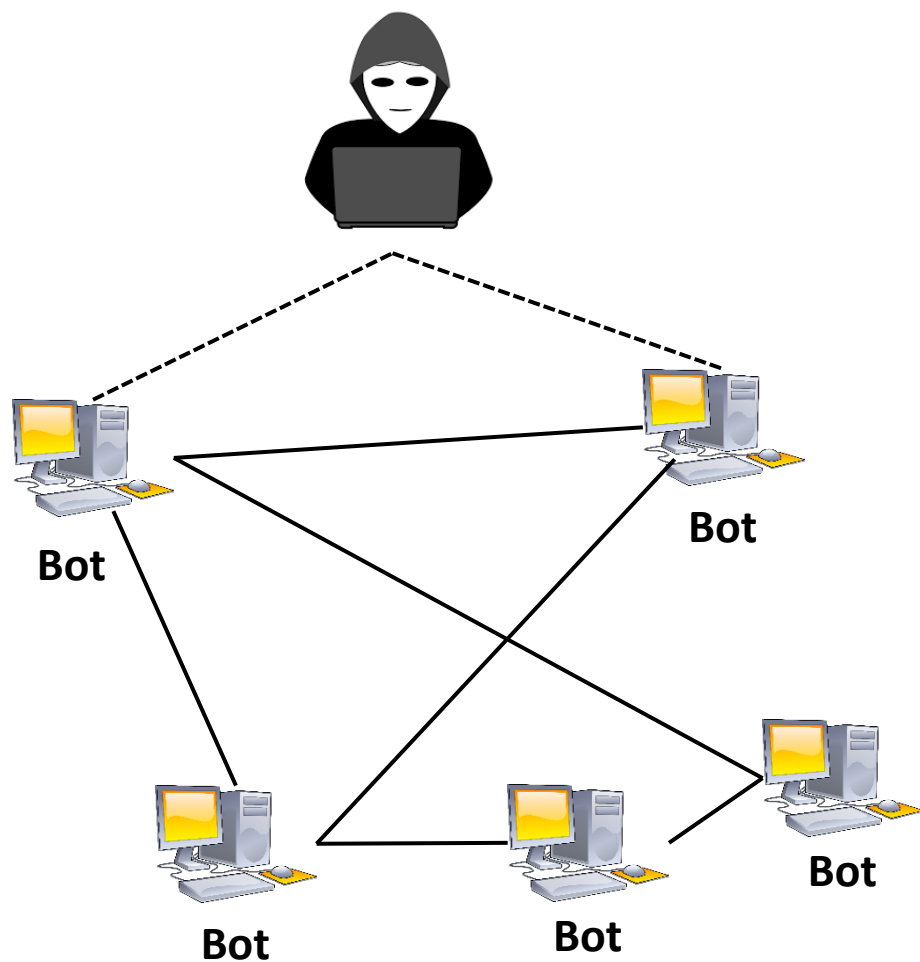


What is the problem here?

C&C single point of failure

the botnet violates the *least common mechanism* principle!

Botnets – P2P Topology

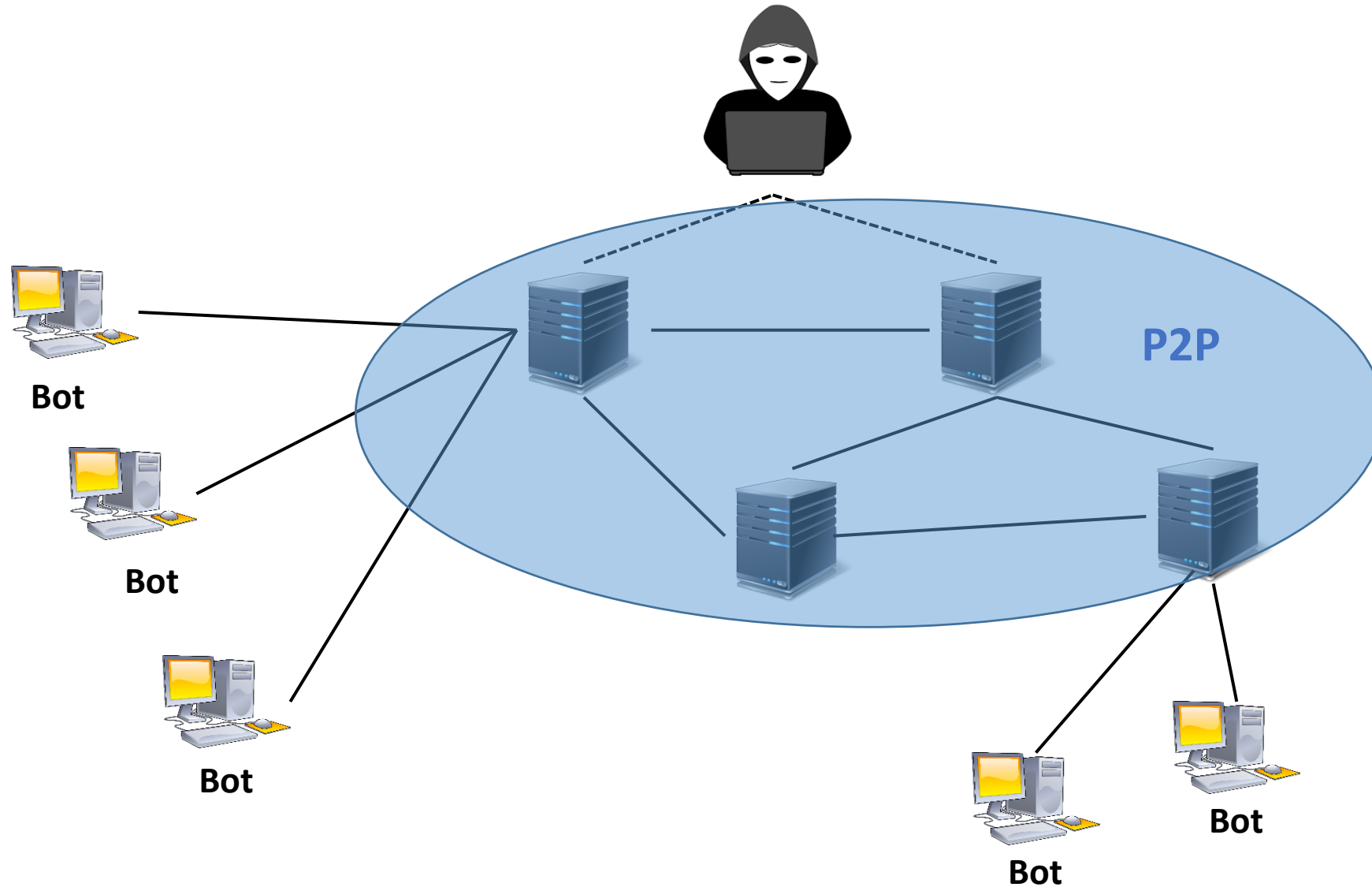


No Command and Control!!

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 your legitimate business”

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

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

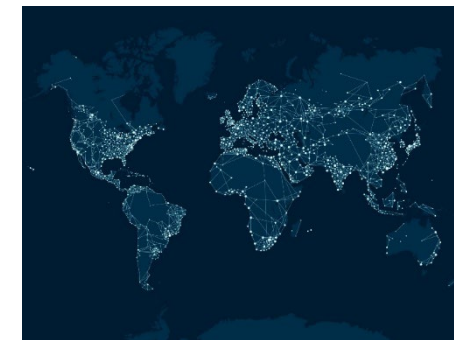
Distribute Ransomware – “I’ve encrypted your hard drive, pay!”

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

Bitcoin mining!!

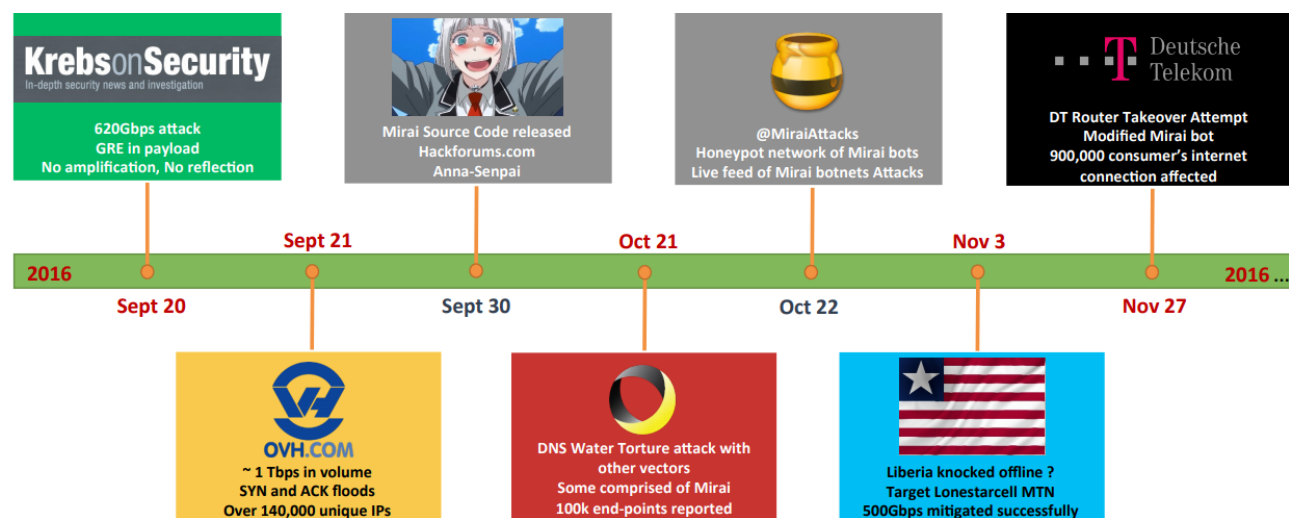
...

Example Botnet – Mirai (2016)



Target: IoT devices

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



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.

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

Summary

Malware = software intentionally malicious

Can be exploited by non-experienced adversaries

**Many types depending on
(auto) replication, need for a host**

Botnets – attacks at scale!