Roberto Guanciale

Computer Security DD2395

Malware

Malware = Malicious Software

- programs exploiting system vulnerabilities
- sophisticated threat to computer systems
- Classification
  - Propagation
  - Payload

Malware Terminology

- Adware
- Attack kit
- Auto-rooter
- Backdoor
- Downloaders
- Flooders
- Keyloggers, Spyware

Propagation
Payload

Malware Terminology

- Logic bomb
- Virus
- Ransomware
- Rootkit
- Spammers
- Trojan horse
- Worm
- Zombie, bot

Propagation
Payload

Propagation

Propagation: **Worms**

- replicating program that propagates over net
  - using email, remote exec, SW flaw (USB stick)
  - zero-day exploit
  - old bugs (NSA Leackage, Server Message Block (SMB) protocol used by discontinued Windows, WannaCry)
- has phases:
  - Does not damage the system immediately

Propagation: **Worms**

- replicating program that propagates over net
  - using email, remote exec, SW flaw (USB stick)
  - zero-day exploit
  - old bugs (NSA Leackage, Server Message Block (SMB) protocol used by discontinued Windows, WannaCry)
- has phases:
  - dormant, propagation, triggering, execution
  - propagation phase: searches for other systems, connects to it, copies self to it and runs

Morris Worm

- one of best known early worms
- released by Robert Morris in 1988
- various attacks on UNIX systems
  - cracking password file to use login/password to login to other systems
  - exploiting a bug in sendmail
- if succeed have remote shell access
  - sent bootstrap program to copy worm over
- Payload: Involuntary fork bomb

Famous worm attacks

- Code Red
  - July 2001 exploiting MS IIS bug
  - probes random IP address, does DDoS attack
  - consumes significant net capacity when active
- Code Red II
- SQL Slammer (2003)
- Stuxnet (2010)
- WannaCry attack (2017)

Famous worm attacks: MIRAI

- Botnet of IOT devices
- Search for CCTV camera with
  - Dahua firmware or
  - a generic management interface called “NETSurveillance
- Scan for telnet
- Bruteforce login using 63 default passwords
  - https://github.com/jgamblin/Mirai-Source-Code/blob/6a5941be681b839eeff8ece1de8b245bcd5ffb02/mirai/bot/scanner.c
- Download payload (Worm+DoS engine)

Propagation: **Virus**

- piece of software that infects programs
  - modifying them to include a copy of the virus
  - so it executes secretly when host program is run

- a typical virus goes through phases of:
  1) dormant
  2) propagation
  3) triggering
  4) execution

Propagation: **Virus**

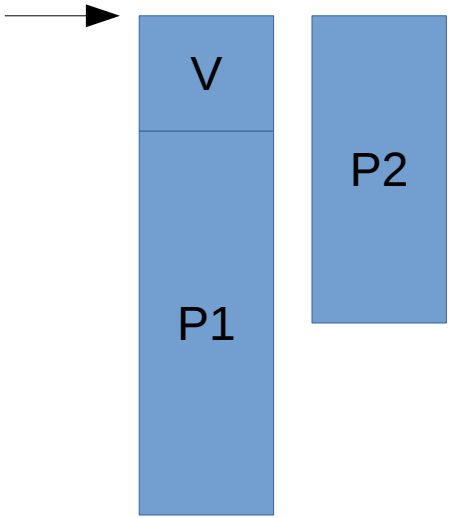
- components:
  - infection mechanism - enables replication
  - modification engine - for disguise
  - trigger - event that makes payload activate
  - payload - what it does, malicious or benign

- when infected program invoked
  - executes virus code then
  - executes original program code

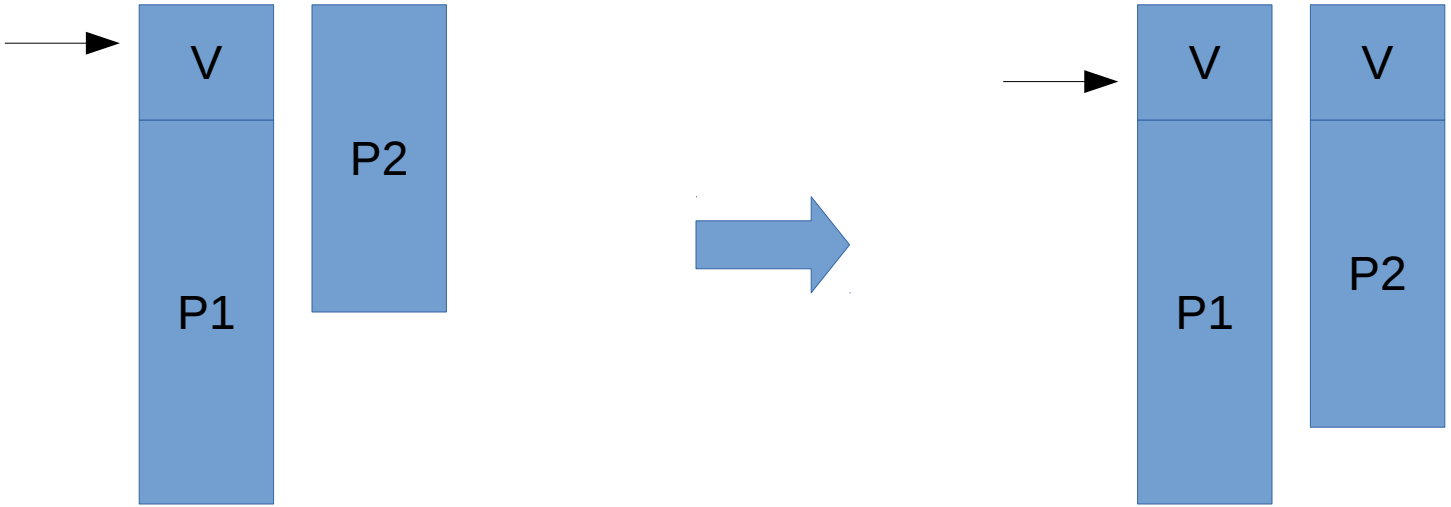
Naive Virus

```
0x00000000 goto main;
0x00000004 666;
main:      infect();
           if trigger() then do-damage();
           goto victim;
infect:    file = get-random-exec();
           if file[0x00000004] = 666 goto infect;
           prepend V to file;
           return;
do-damage: ...
trigger:   ...
victim:    ...
```

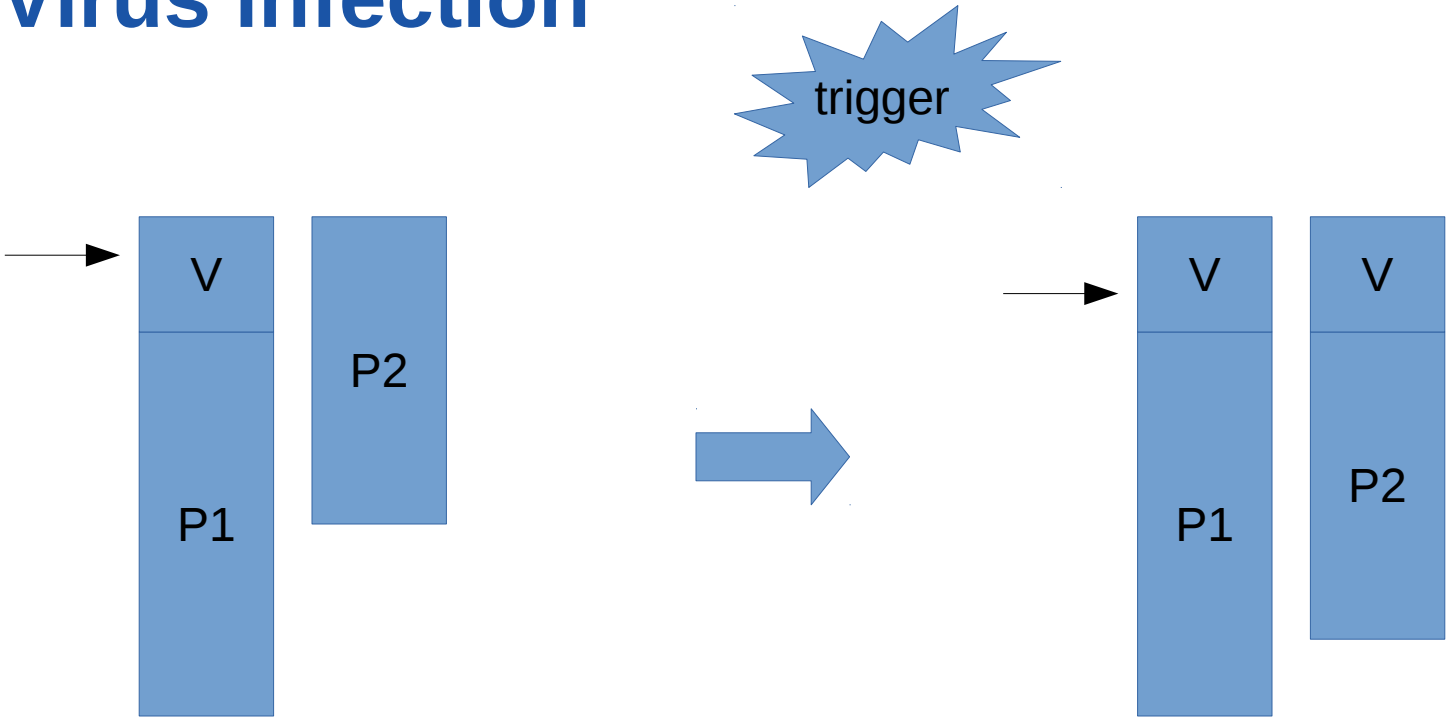
Virus infection



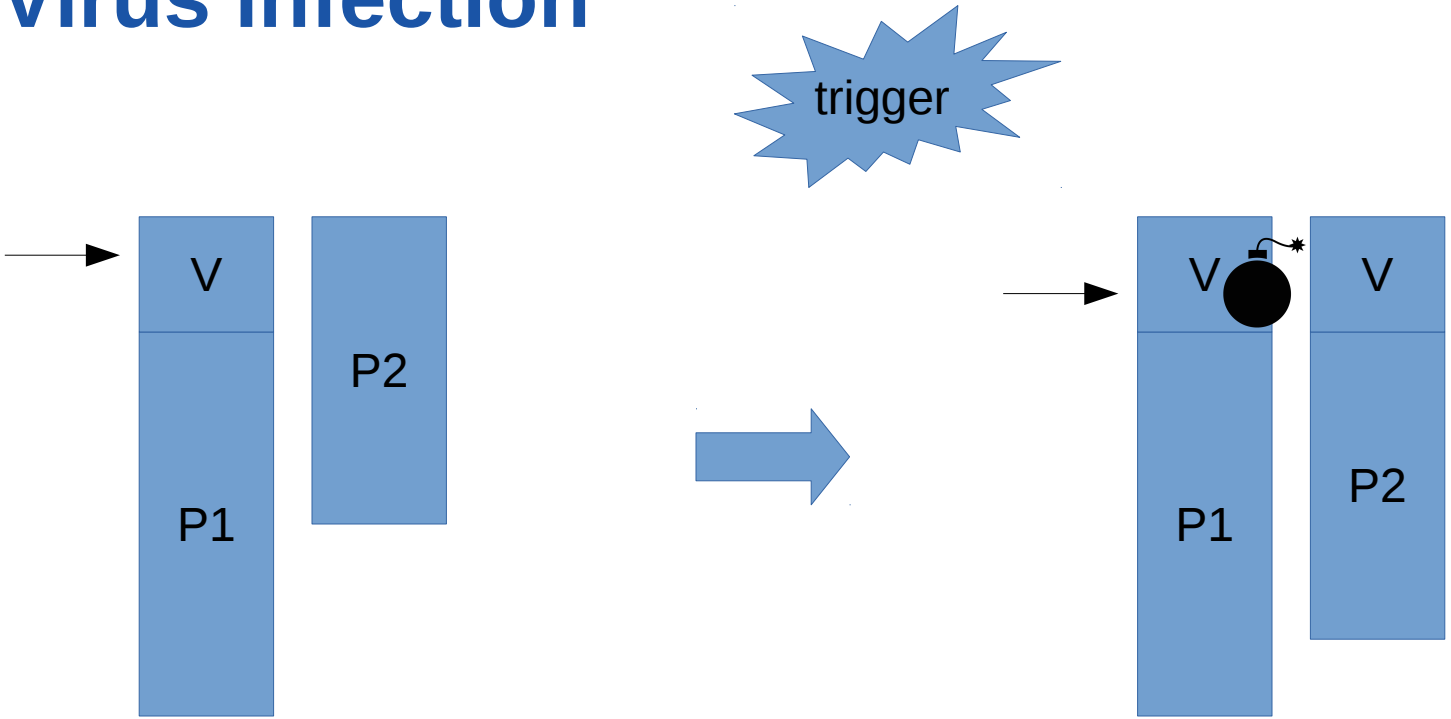
Virus infection



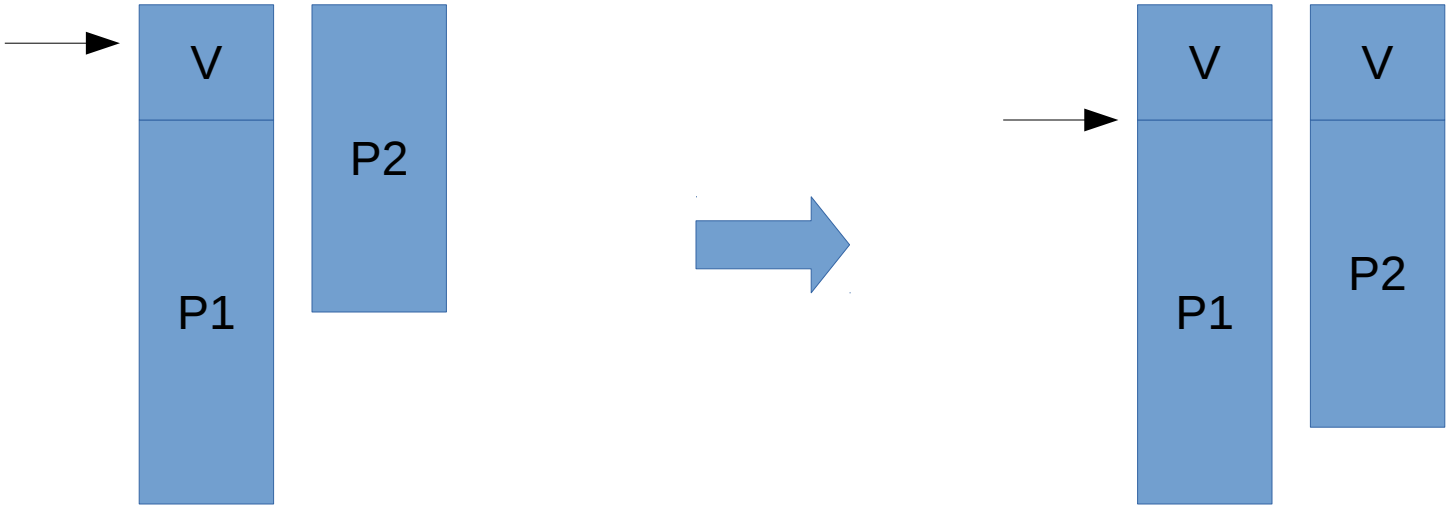
Virus infection



Virus infection



Virus infection



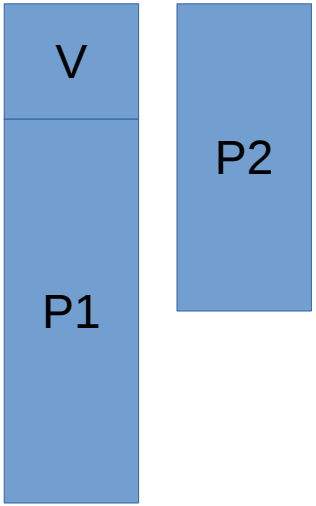
Virus Identification

- Size of the program P2 changed

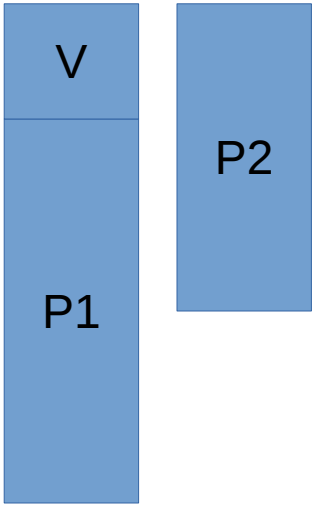
Virus 2.0 (compression)

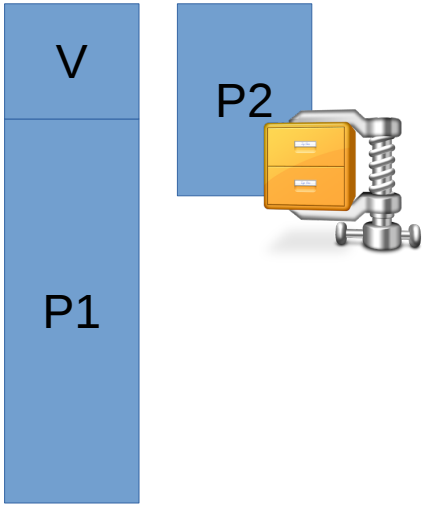
```
0x00000000 goto main;
0x00000004 666;
main:      infect();
           if trigger() then do-damage();
           uncompressVictim();
           goto victim;
infect:    file = get-random-exec();
           if file[0x00000004] = 666 goto infect;
           Compress(file);
           prepend V to file;
           return;
...
```

Virus infection

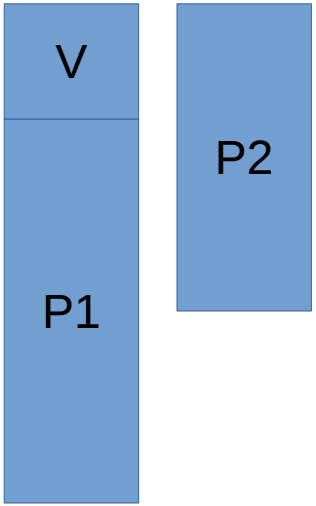


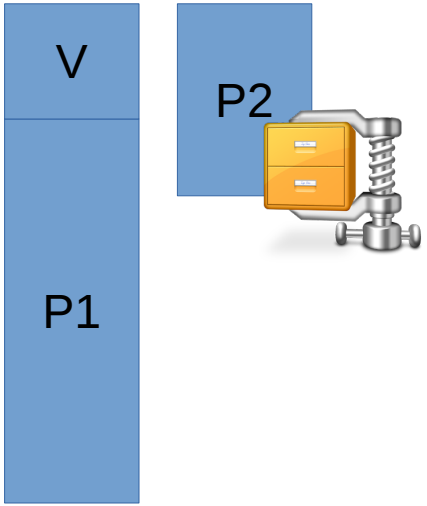
Virus infection

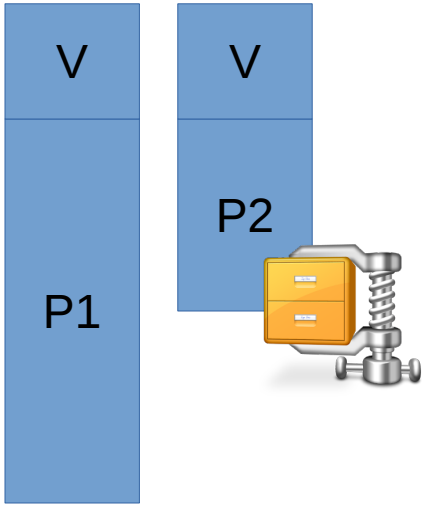




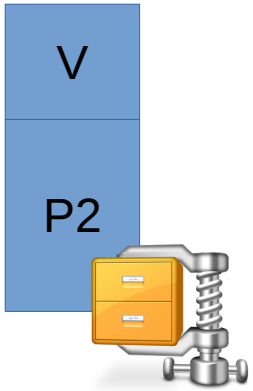
Virus infection







Virus infection



Virus Identification

- ~~Size of the program P2 changed~~

Virus Identification

- ~~Size of the program P2 changed~~
- Content of the program P2 changed

Virus Identification

- ~~Size of the program P2 changed~~
- Content of the program P2 changed
- Signature of the program P2 changed

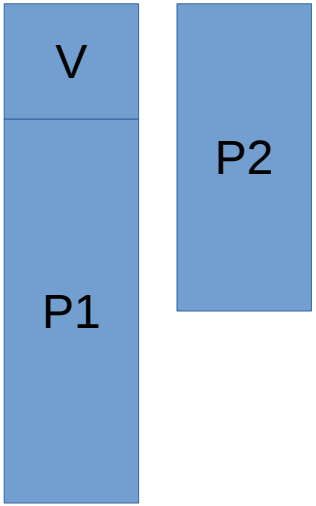
Virus Identification

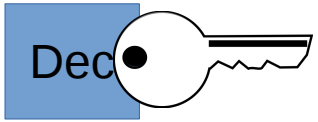
- ~~Size of the program P2 changed~~
- Content of the program P2 changed
- Signature of the program P2 changed
- P2 contains the fixed code V of the virus

Virus 3.0 (Encrypted)

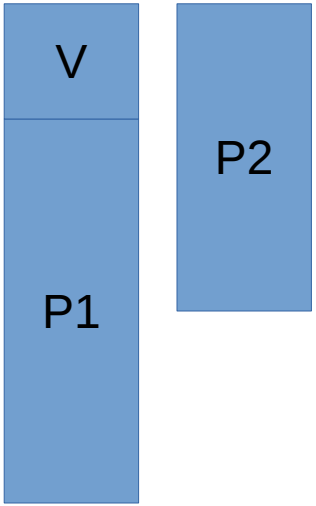
- Generate key
- Encrypt the virus body
- Copy
  - the bootstrap (decryption engine with key) and
  - the encrypted virus body
- When start
  - Decrypt the virus body
  - Execute

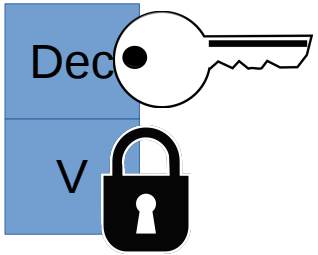
Virus infection



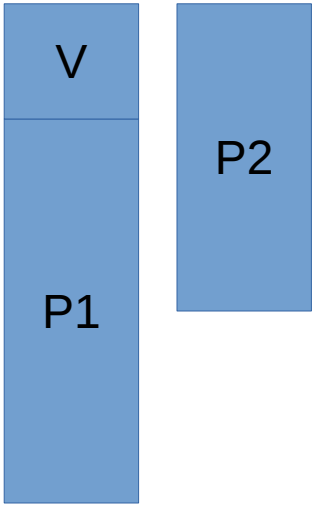


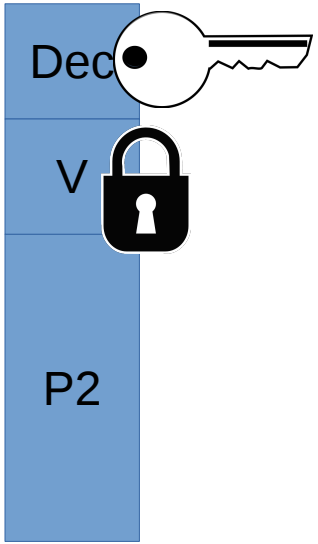
Virus infection



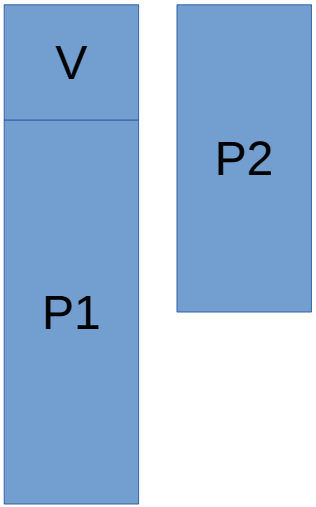


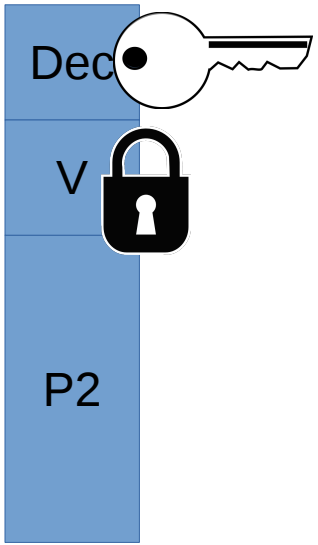
Virus infection

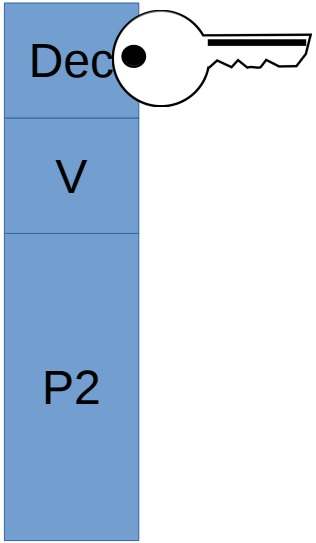




Virus infection







Virus Identification

- ~~Size of the program P2 changed~~
- Content of the program P2 changed
- Signature of the program P2 changed
- ~~P2 contains the fixed code V of the virus~~

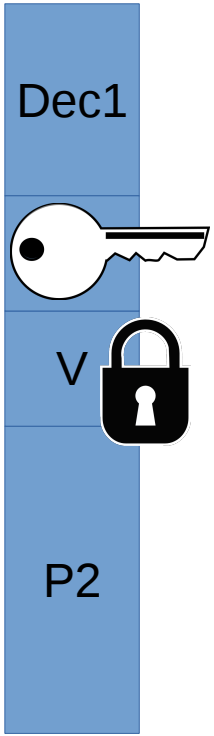
Virus Identification

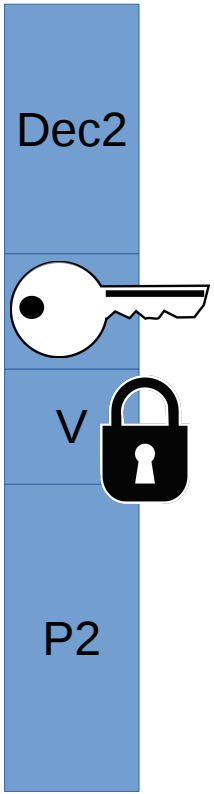
- ~~Size of the program P2 changed~~
- Content of the program P2 changed
- Signature of the program P2 changed
- ~~P2 contains the fixed code V of the virus~~
  - P2 contains a small fixed code Dec
  - We can extract the key from Dec and decrypt V

Virus 4.0 (Polymorphic)

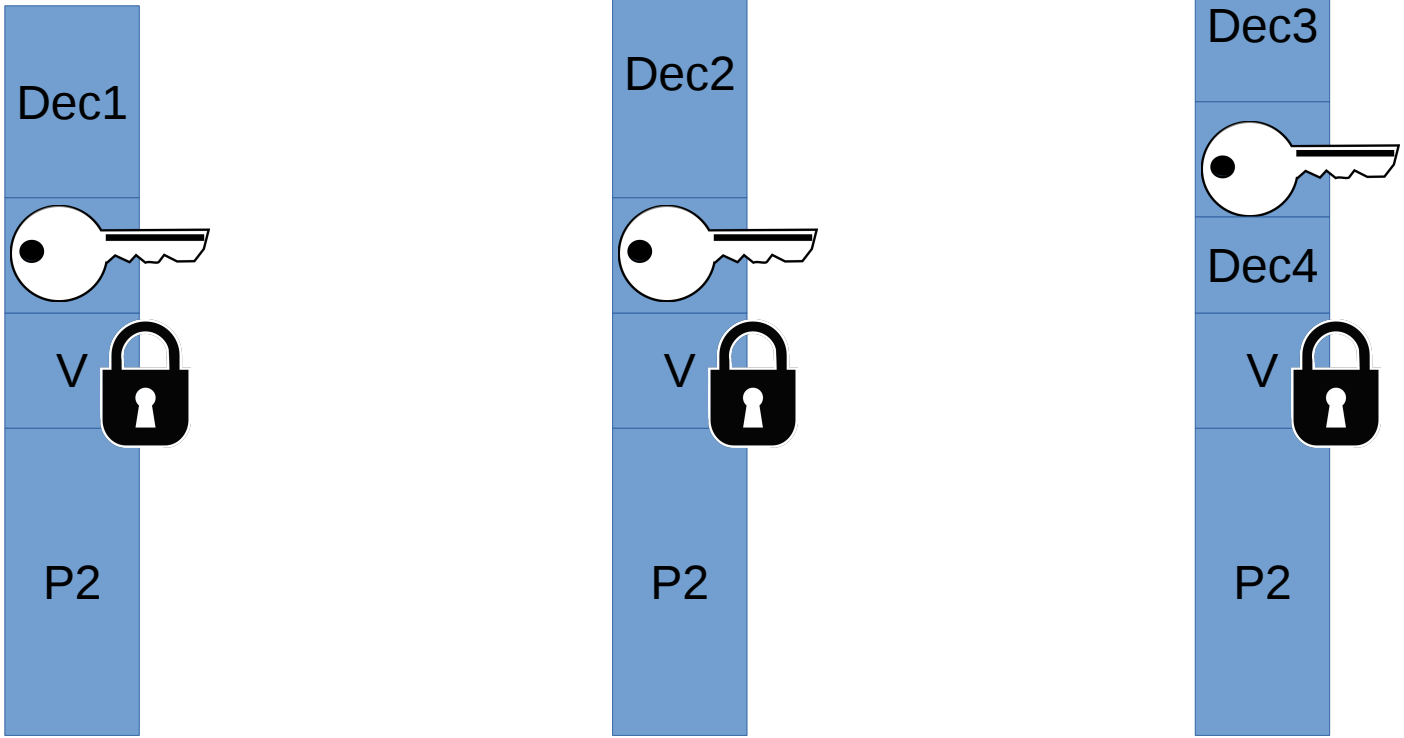
- Rebuild the whole virus at every infection to something functionally identical
- There are many ways to do nothing on a computer
  - X:=X*1 X:= 2*X – X
  - X := X + 5 === Y:= 10; X := Y + X; X:= X – 5
- Instructions can be reordered in many ways
  - X=X+1; Y=Z*2 === Y=Z*2; X=X+1
- Focus on the decryption engine

Virus infection





Virus infection



Macro Virus

- exploit macro/scripting capability of apps
  - Basic

Macro Virus

- exploit macro/scripting capability of apps
  - Basic, Elisp, …

Macro Virus

- exploit macro/scripting capability of apps
  - Basic, Elisp, Javascript, ...

Macro Virus

- exploit macro/scripting capability of apps
  - Basic, Elisp, Javascript, ...
- Why
  - platform independent
  - infects documents
  - is easily spread
- recent applications include protection (e.g. extension capabilities)
- recognized by many anti-virus programs

E-Mail Virus

- e.g. Melissa
  - exploits MS Word macro in attached doc
  - if attachment opened, macro activates
  - sends email to all users in the address-book
  - and does local damage
- versions triggered by just reading email
  - hence much faster propagation

Propagation: **Trojan Horse**

Propagation: **Trojan Horse**

- based on social engineering
- first identified at NSA in 1972 by Daniel Edwards
- it’s a program with two purposes, one obvious and one hidden from the user
- today it’s often used to install other software or backdoors
- trojan horses can be built
  - from existing programs using a special wrapper
  - or designed from the start to be one.

September 2015

- Apple: “25 apps hosted on the App Store are infected by a Malware”
- FireEye: “4000 apps (mainly Chinese) are infected”

- First large-scale attack on Apple's App Store?

September 2015

- Slow download speed from Apple Servers
- Chinese iOS developers used to download Xcode
  - from third party website
  - from colleagues
- Attackers distributed (March 2015) compromised **XcodeGhost**
- Compiler backdoor attack
  - malicious code is automatically integrated into the compiled app
  - without the developer knowledge
- Malware: deliver data, receives command from a control center

Malware Propagation

- Attack kit (SW designed to build malware)
- Backdoor (Debug/Intentional)
- Downloaders

Payload

Payload: **integrity-loss**

- Deletion of files, contacts, etc
- Ransomware
  1) encrypt files
  2) delete files
  3) collect ransom money (e.g. bitcoin)

Payload: **availability-loss**

- Deletion of files, change of firewall rules
- Physical integrity
  - Stuxnet, nuclear centrifuges via compromising Siemens control software

Payload: **logic-bomb**

- A small bit of code that triggers on a specific condition
- Typically with malicious results
- No vector for spreading
- Installed directly

Payload: **logic-bomb**

- A small bit of code that triggers on a specific co
- Typically with malicious results
- No vector for spreading
- Installed directly
- On 29 October 2008 a logic bomb was discovered at Fannie Mae. The bomb was planted by an IT contractor, was set to activate on 31 January 2009 and could have wiped all of 4000 servers. The contractor had been dismissed around 1:00pm on 24 October 2008 and managed to plant the bomb before his network access was revoked.

Payload: **logic-bomb**

- A small bit of code that triggers on a specific co
- Typically with malicious results
- No vector for spreading
- Installed directly

- E.g. Kill switch of WannaCry
  - Found by Marcus Hutchins
  - Kill switch or anti quarantine measure

Payload: **bots**

- program taking over other computers
- to launch hard to trace attacks
- if coordinated form a botnet
- characteristics:
  - remote control facility
- multi-layered networks of bots
- MIRAI

Payload: **bots**

- For Distributed Denial of Service
- For spamming
- For manipulating on-line surveys
- For using computational resources
  - (e.g. bit coin mining, password brute forcing)
- For cheating ad-providers (e.g. ad-clickers)
- For spreading new malware

Payload: **confidentiality loss**

- Key loggers / pin loggers
  - Password theft
- Spyware
  - Camera
  - Documents
- Phishing
- Espionage
- Identity theft

Payload: **backdoors**

- Software that gives access to a system
- Bypassing OS-login restrictions
- Often installed for legitimate reasons
- Only to later be abused

- Intel Management Engine?

Payload: **rootkit**

- program installed for admin access
- malicious and stealthy changes to OS

- may hide its existence
  - subverting report mechanisms on processes, files, registry entries etc

- user or kernel mode
- installed by trojan or worm
- Sony BMG copy protection rootkit and trojan

Countermeasures

Countermeasures - prevention: code signing

- Used in digital SW distribution
  - e.g. Debian APT, Android play store
- Used by HW vendor to prevent execution of arbitrary SW
  - e.g. XBox
- Used by OSes to install binary drivers
  - e.g. Windows

- SW shipped with a digital certificate
  - e.g. cert = Enc(PR-google , Hash(SW))
- check Dec(PU-google , cert) = Hash(SW)

Countermeasures - prevention: code signing

- Root of trust
  - HW checks signature of OS
  - OS checks signature of apps
  - app checks signature of plug-ins
  - …

Countermeasures - prevention: code signing

- Root of trust
  - HW checks signature of OS
  - OS checks signature of apps
  - app checks signature of plug-ins
  - …

1) Bug in OS enables unsigned apps to be executed
2) If the signature is checked only at startup, a buffer overflow can be used to execute unsigned code

Countermeasures - prevention: code signing

- Root of trust
  - HW checks signature of OS
  - OS checks signature of apps
  - app checks signature of plug-ins
  - …

1) Bug in OS enables unsigned apps to be executed
2) If the signature is checked only at startup, a buffer overflow can be used to execute unsigned code
3) Cannot be used for JIT / Documents

Countermeasures - **prevention/detection**

- Prevention Systems / Detection Systems
- At network level
  - (e.g. to temporarily prevent a worm infection due to a know bug)
- At application level
  - (e.g. SMTP servers inspecting mails)

Countermeasures - **prevention/detection**

- Prevention Systems / Detection Systems
- At network level
  - (e.g. to temporarily prevent a worm infection due to a know bug)
- At application level
  - (e.g. SMTP servers inspecting mails)
- Behavioral based
  - e.g. It is unusual to attempt and fail several TCP connections
  - e.g. An application that uses more CPU cycles that usual

Countermeasures - **prevention/detection**

- Prevention Systems / Detection Systems
- At network level
  - (e.g. to temporarily prevent a worm infection due to a know bug)
- At application level
  - (e.g. SMTP servers inspecting mails)
- Behavioral based
  - e.g. It is unusual to attempt and fail several TCP connections
  - e.g. An application that uses more CPU cycles that usual
- Honeypots:
  - Networked and local

Anti-Virus Evolution

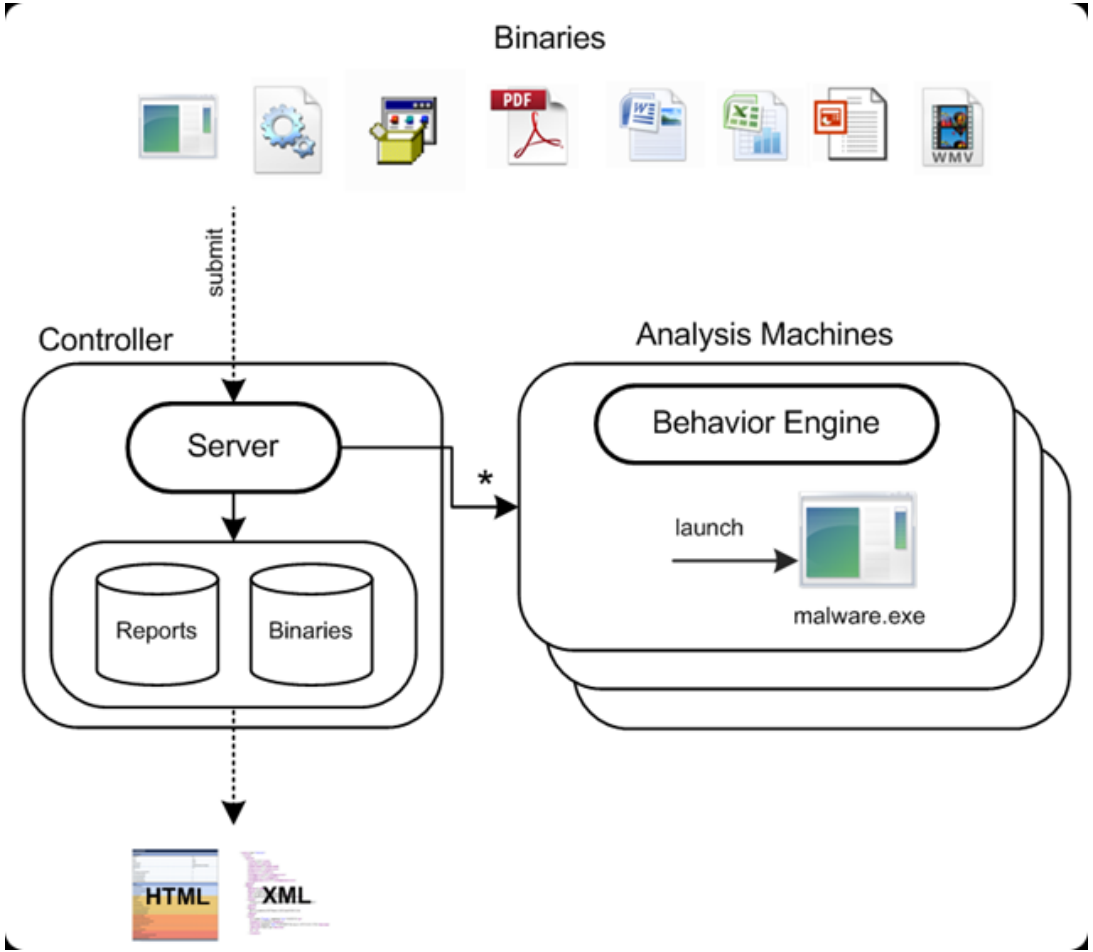
- virus & antivirus tech have both evolved
  - early viruses simple code, easily removed
- more complex viruses, more complex antiviruses
- generations
  - first - signature scanners
  - second - heuristics
  - third - identify actions
  - fourth - combination packages

Generic Decryption

- runs executable files through GD scanner:
  - CPU emulator
  - virus scanner to check known virus signatures

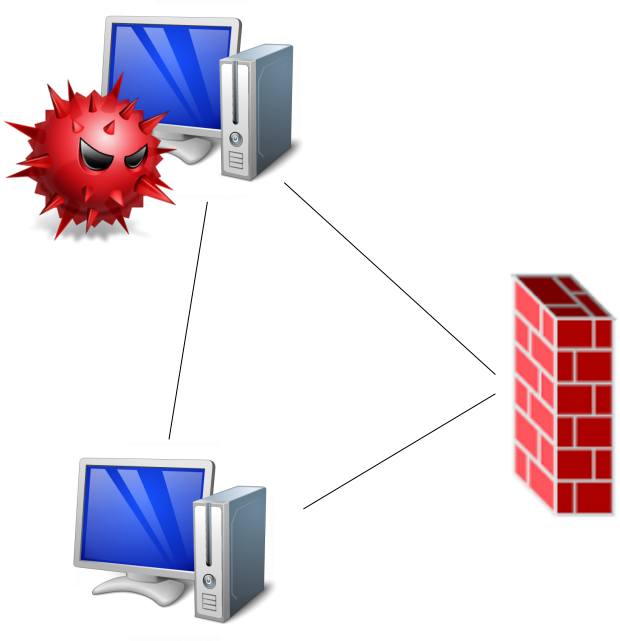
- let virus decrypt itself in interpreter
- periodically scan for virus signatures
- Issue: is long to interpret and scan
  - trade-off chance of detection vs time delay

Virtualization and Sandboxing



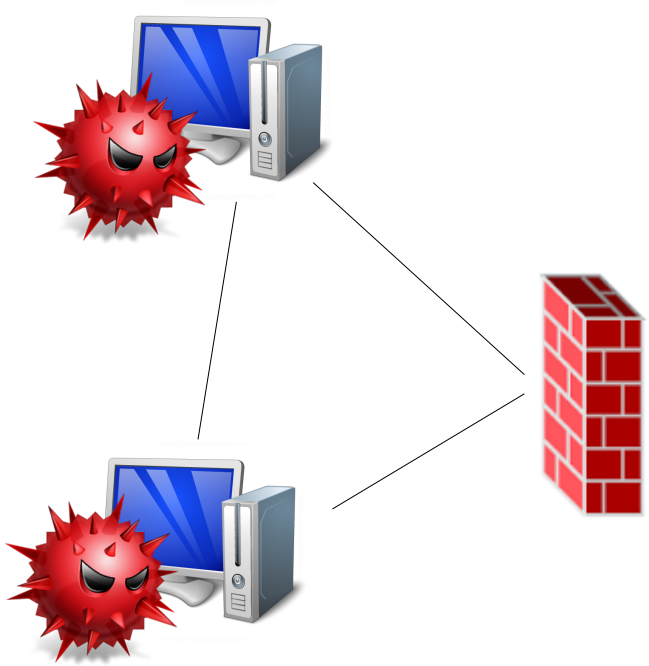
Stuxnet
Cyberwar in 2010

Stuxnet – Local area network infection



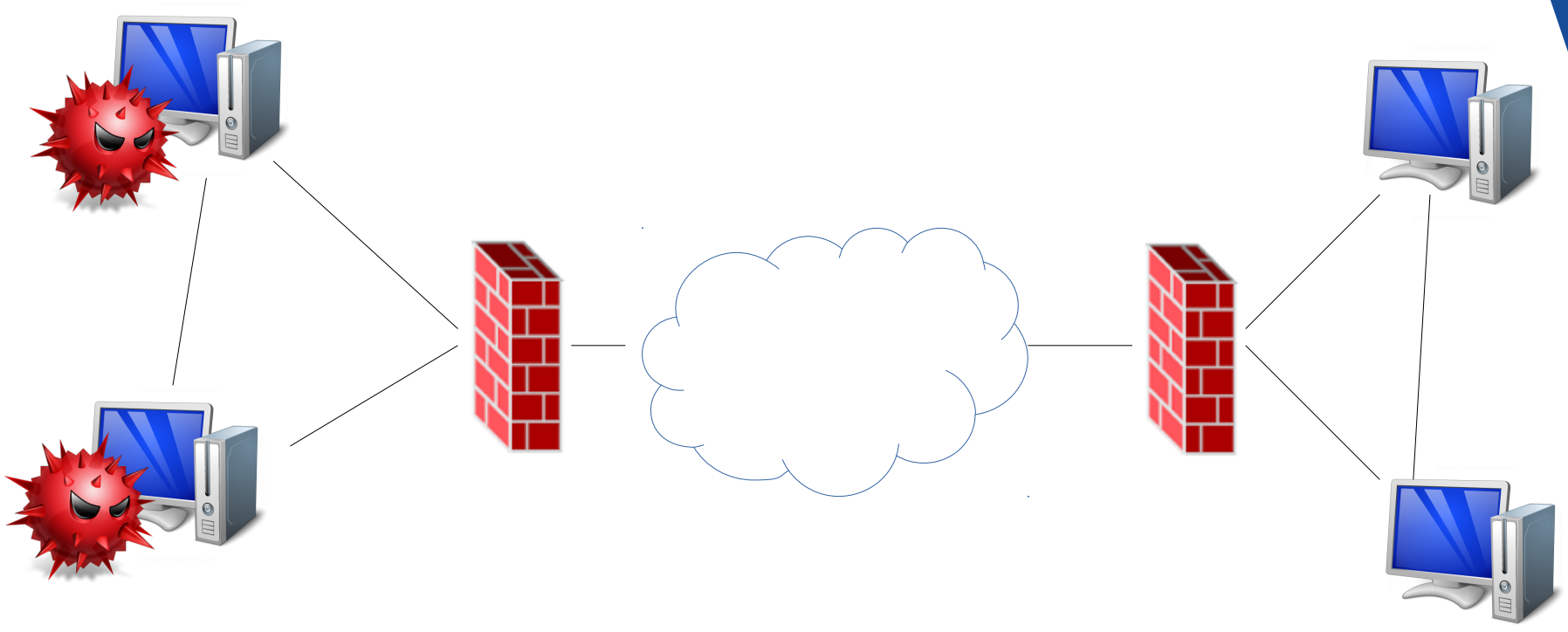
Remote code execution on a computer with Printer Sharing enabled

Stuxnet – Local area network infection

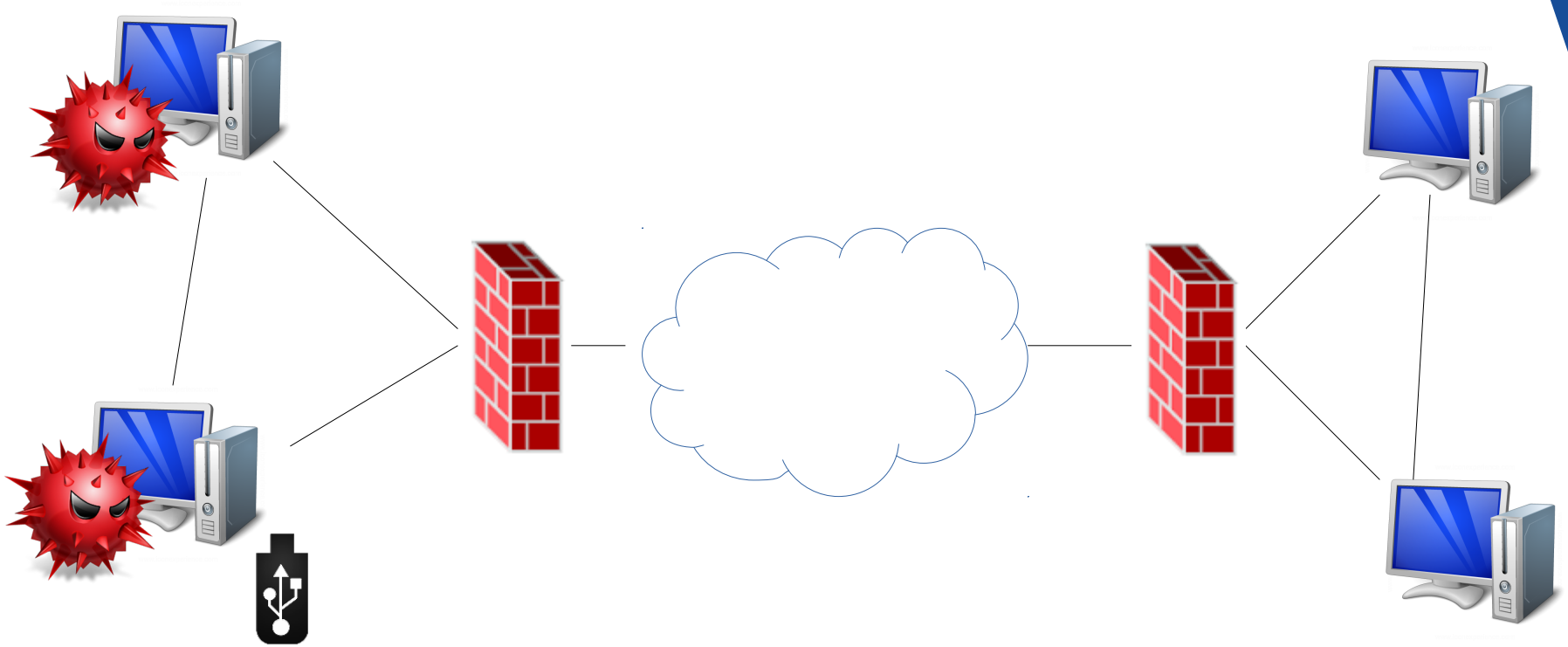


Remote code execution on a computer with Printer Sharing enabled

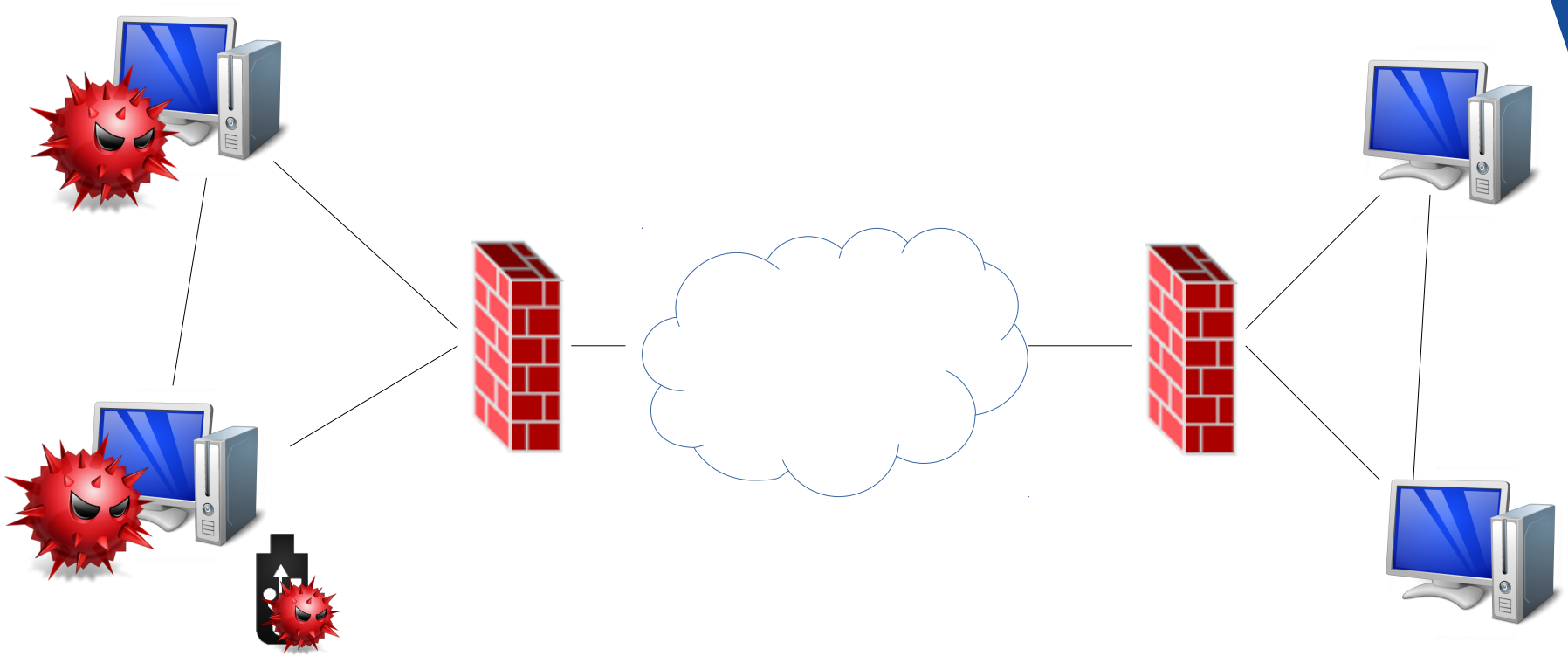
Stuxnet – propagation



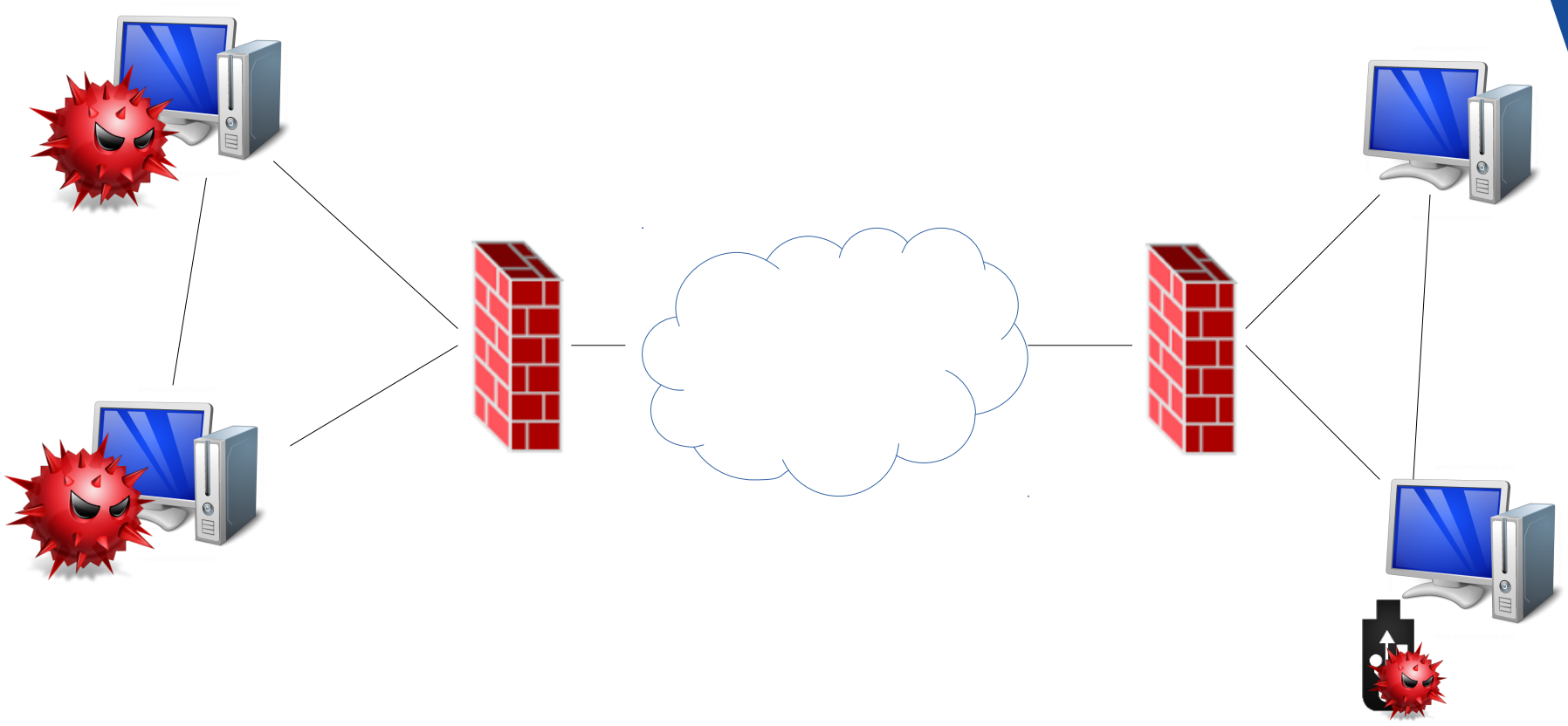
Stuxnet – propagation



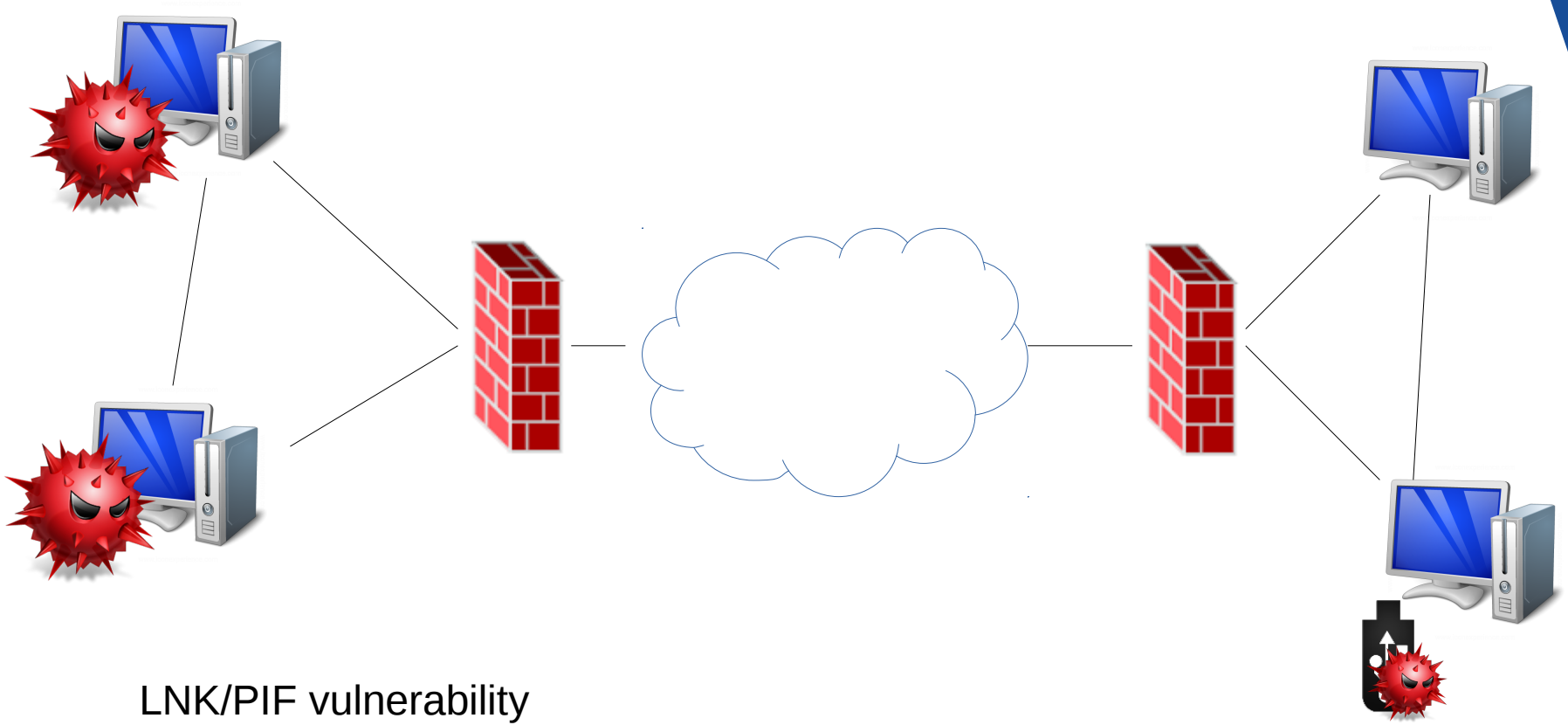
Stuxnet – propagation



Stuxnet – propagation

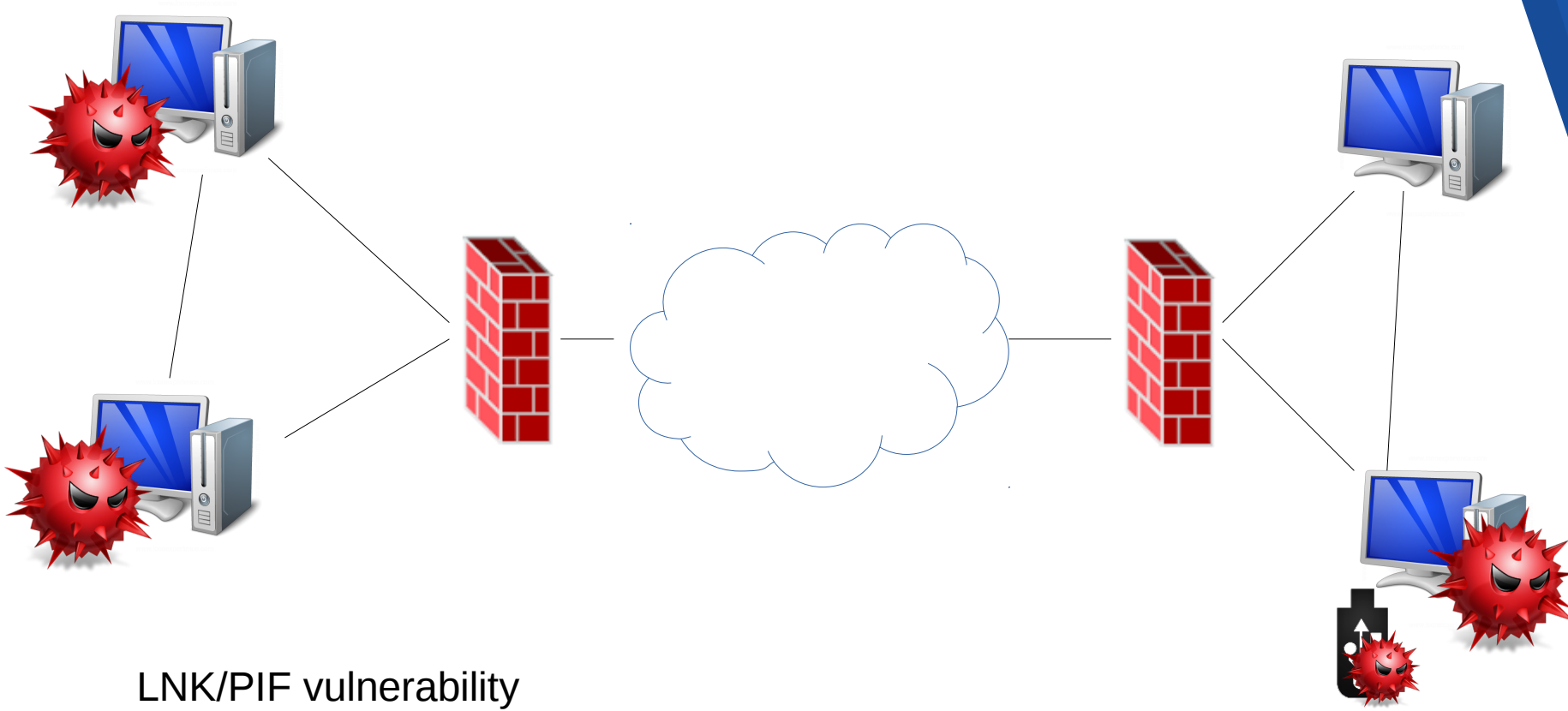


Stuxnet – propagation



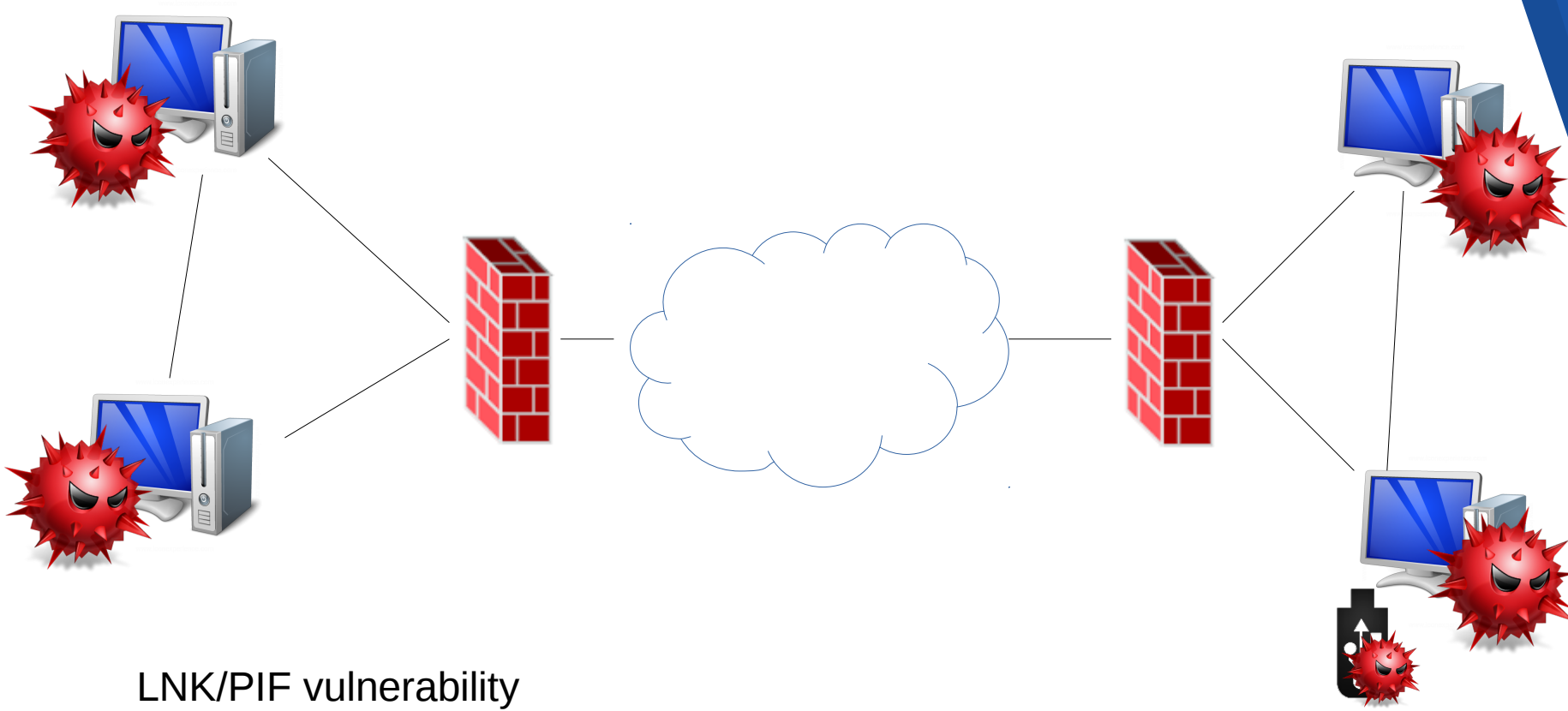
LNK/PIF vulnerability
payload execution when an icon is viewed in Windows Explorer

Stuxnet – propagation



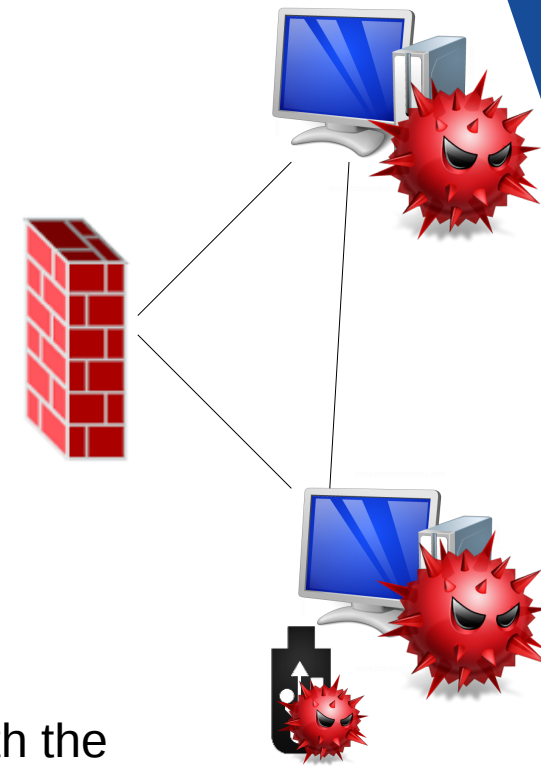
LNK/PIF vulnerability
payload execution when an icon is viewed in Windows Explorer

Stuxnet – propagation



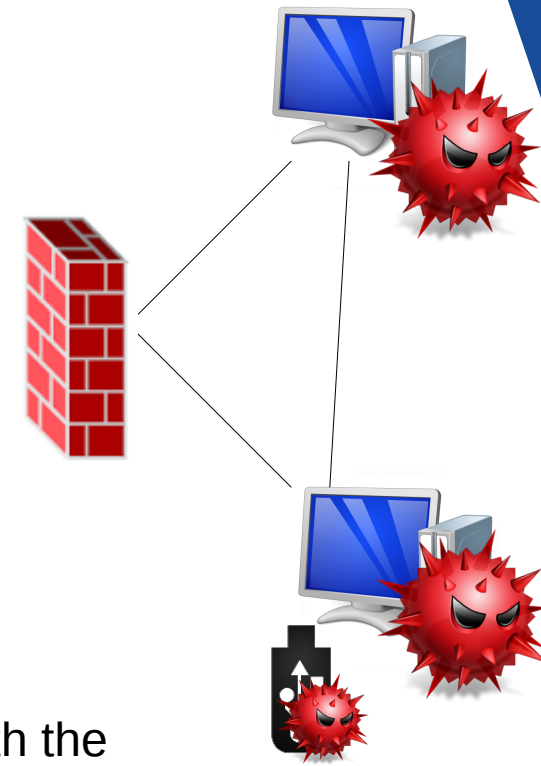
LNK/PIF vulnerability
payload execution when an icon is viewed in Windows Explorer

Stuxnet – hiding



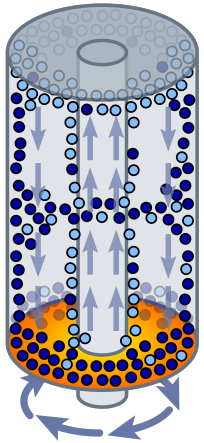
kernel-mode rootkit: device drivers digitally signed with the **private keys** of Realtek

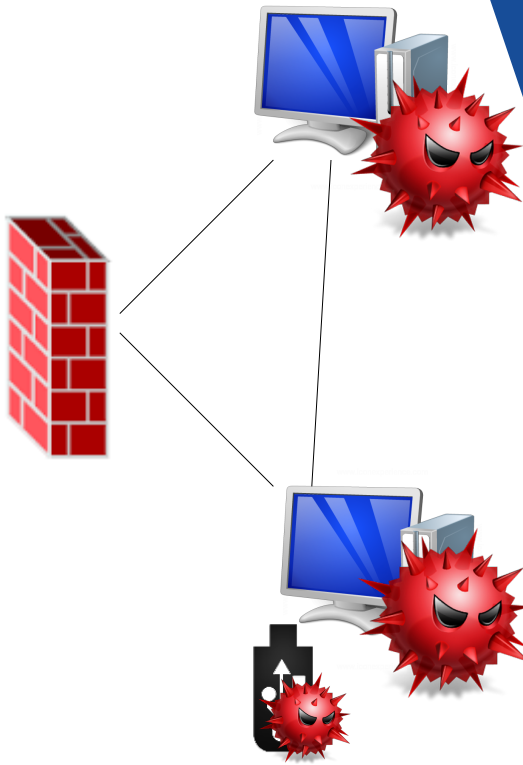
Stuxnet – hiding



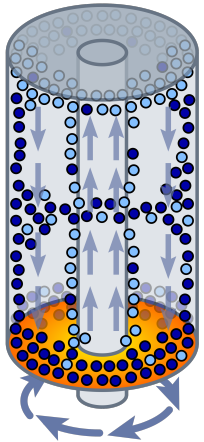
kernel-mode rootkit: device drivers digitally signed with the **private keys** of Realtek

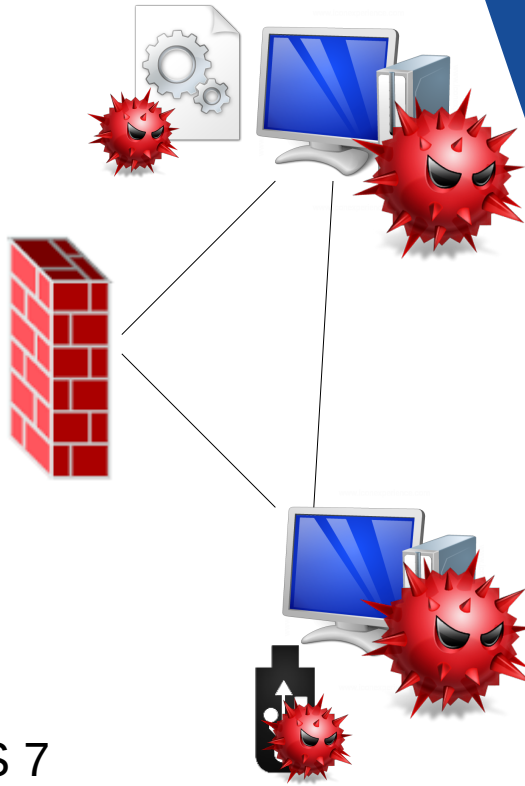
Stuxnet – Infection of Industrial Control Systems





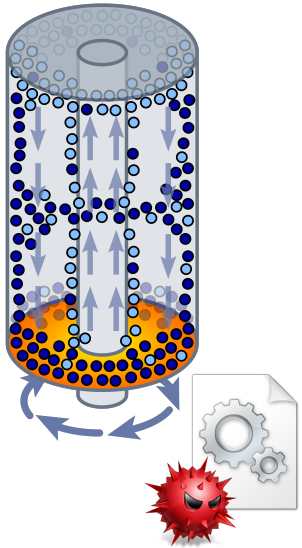
Stuxnet – Infection of Industrial Control Systems

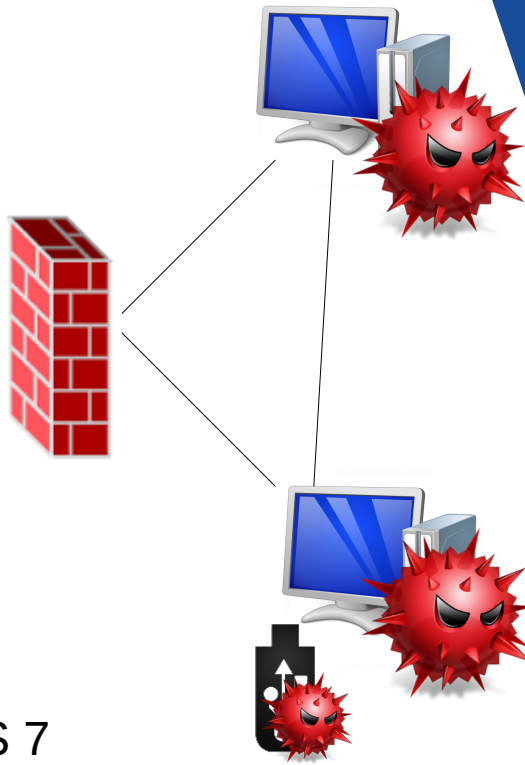




Infects project files belonging to Siemens' WinCC/PCS 7 SCADA control software

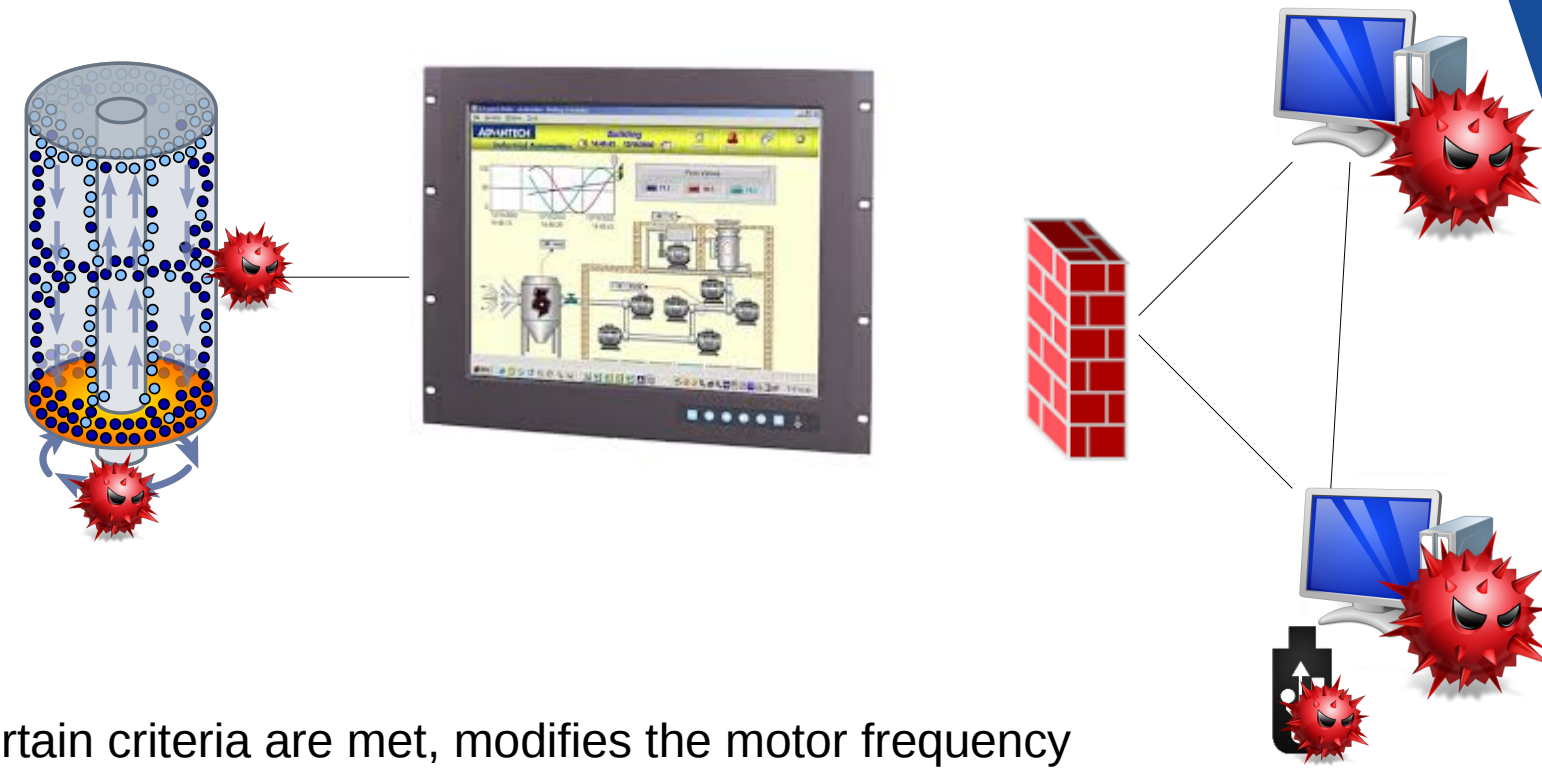
Stuxnet – Infection of Industrial Control Systems





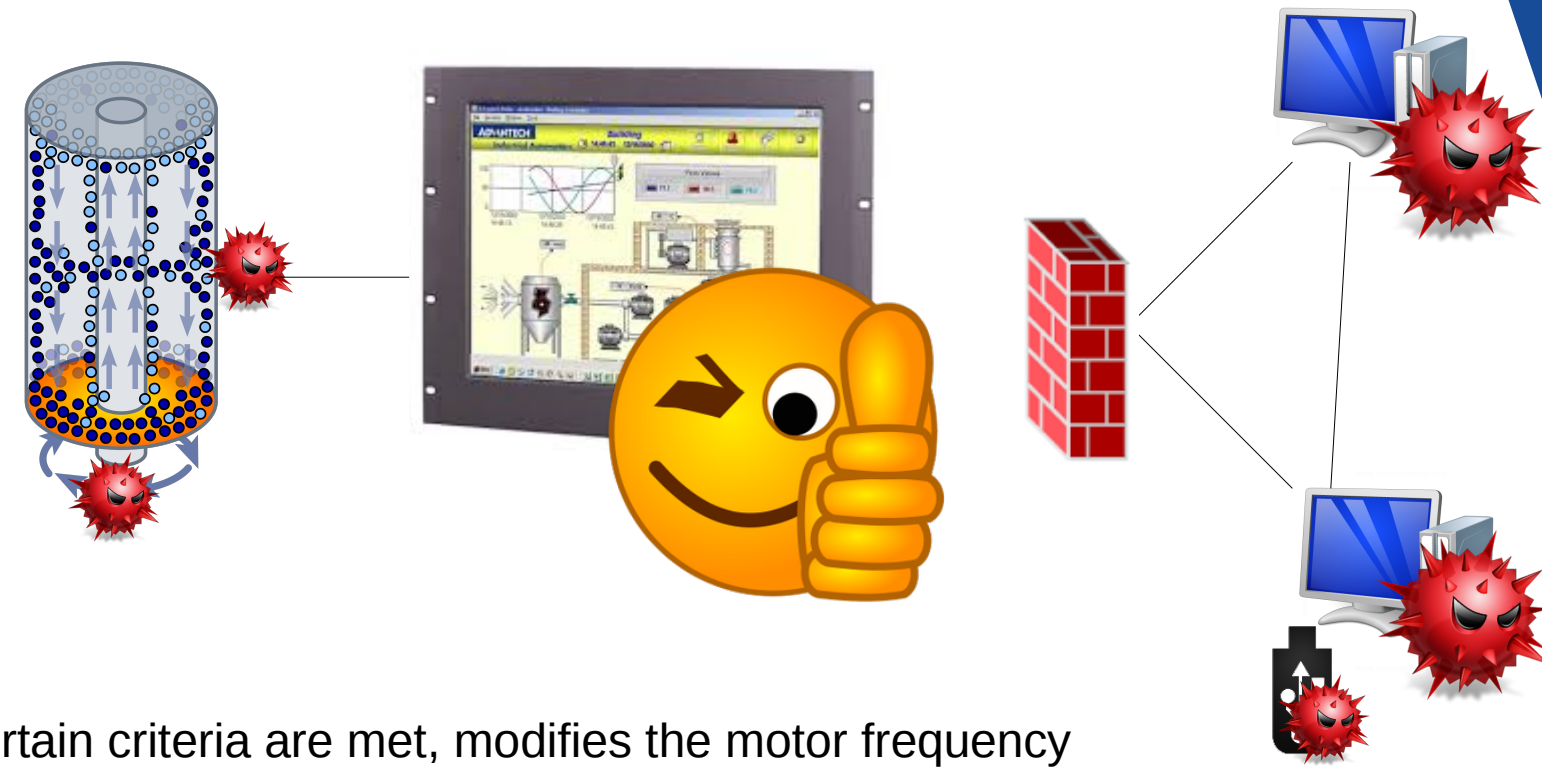
Infects project files belonging to Siemens' WinCC/PCS 7 SCADA control software

Stuxnet – Infection of Industrial Control Systems



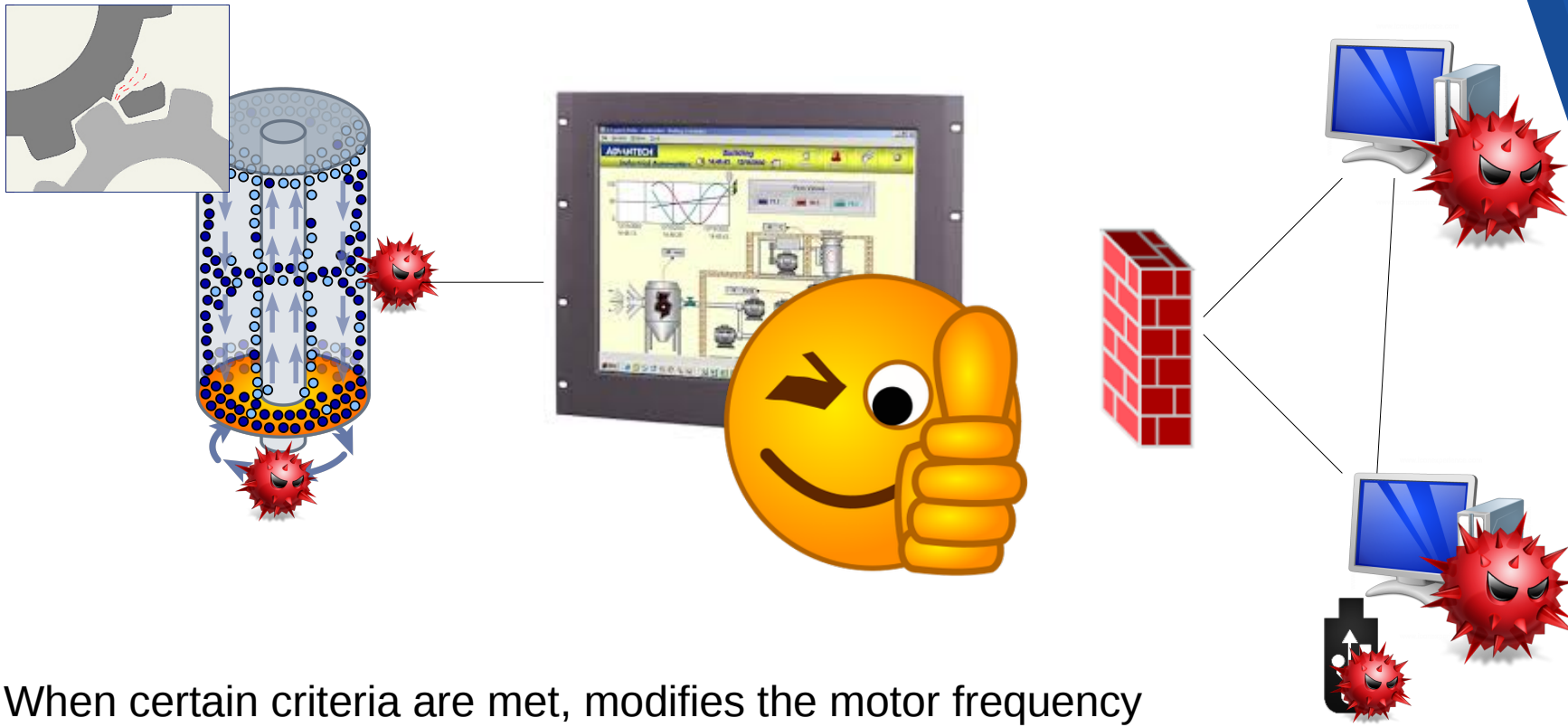
1) When certain criteria are met, modifies the motor frequency
2) Hides the changes in rotational speed from monitoring systems

Stuxnet – Infection of Industrial Control Systems



1) When certain criteria are met, modifies the motor frequency
2) Hides the changes in rotational speed from monitoring systems

Stuxnet – Infection of Industrial Control Systems



1) When certain criteria are met, modifies the motor frequency
2) Hides the changes in rotational speed from monitoring systems

THANKS!

Any questions?

You can find me at robertog@kth.se