

Malware  
CSE 548 Spring 2024

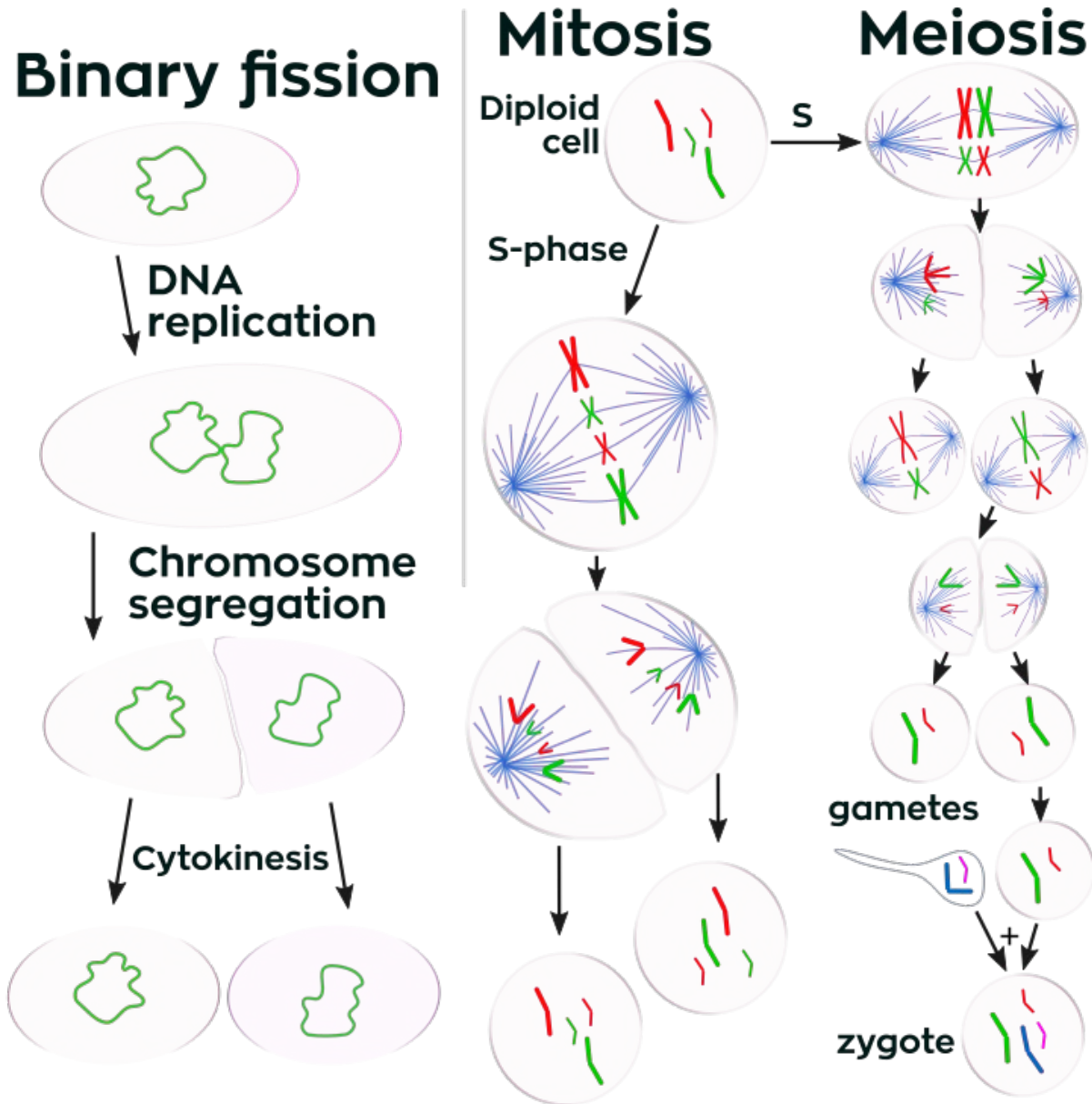
What can make a binary object illegal?

Imagine the best encrypted tunnel you can create,  
*e.g.* TLS... where are the weakest points?

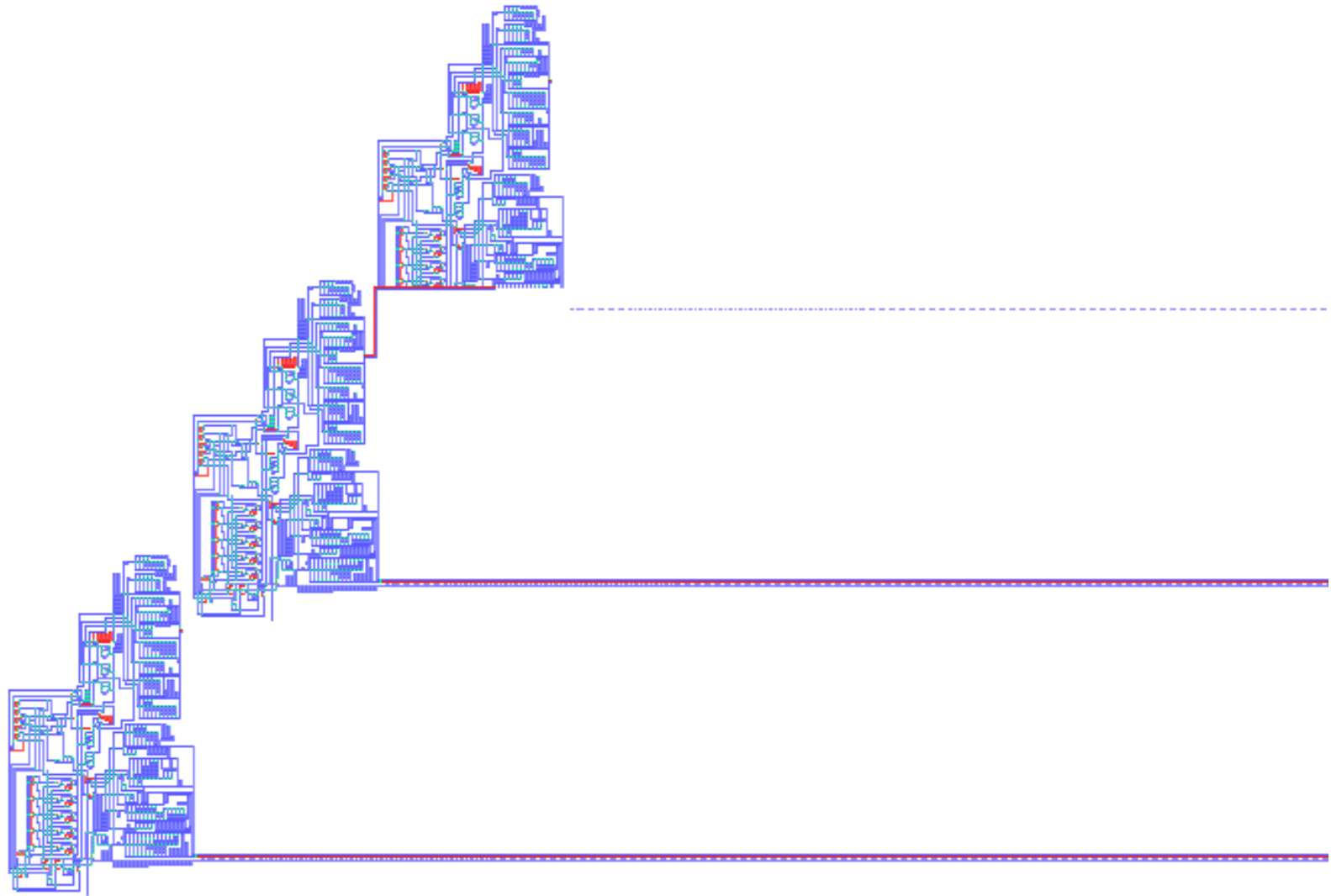
# Malware vs. viruses

- Malware
  - Some personal or political relationship between the binary object and individuals
  - Often exceeds authorization
- Viruses (including worms, *etc.*)
  - Often malicious, *i.e.*, malware
  - *Self-propagating/self-replicating*

[https://en.wikipedia.org/wiki/Cell\\_division](https://en.wikipedia.org/wiki/Cell_division)



[https://en.wikipedia.org/wiki/Von\\_Neumann\\_universal\\_constructor](https://en.wikipedia.org/wiki/Von_Neumann_universal_constructor)  
(1940s)



# The dawn of computer viruses/worms

- “Worm” came from John Brunner's *The Shockwave Rider* in 1975
  - Creeper in 1971 for TENEX systems (Reaper)
  - ANIMAL in 1975
  - Morris Worm in 1988
  - Code Red in 2001
- Elk Cloner in 1981 (Skrenta)
- “Virus” coined by Cohen in 1983 (“Information only has meaning in that it is subject to interpretation”)
  - <https://web.eecs.umich.edu/~aprakash/eecs588/handouts/cohen-viruses.html>
- A “worm” uses a computer network as its main mode of propagation
  - Also alarming to people in 2001: staying in memory and never going out to disk

# Malware gets personal

- Brain PC virus in 1986
  - Goal was to protect their copyright
  - Infected machines worldwide
- Amiga viruses (late 1980's)
- MSOffice Macroviruses (1995 to 2003ish)

```
Displacement Hex codes ASCII value
0000(0000) FA E9 4A 01 34 12 00 07 14 00 01 00 00 00 00 20 -0J04↑0Π0
0016(0010) 20 20 20 20 20 20 57 65 6C 63 6F 6D 65 20 74 6F Welcome to
0032(0020) 20 74 68 65 20 44 75 6E 67 65 6F 6E 20 20 20 20 the Dungeon
0048(0030) 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0064(0040) 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0080(0050) 20 28 63 29 20 31 39 38 36 20 42 61 73 69 74 20
0096(0060) 26 20 41 6D 6A 61 64 20 28 70 76 74 29 20 4C 74
0112(0070) 64 2E 20 20 20 20 20 20 20 20 20 20 20 20 20
0128(0080) 20 42 52 41 49 4E 20 43 4F 4D 50 55 54 45 52 20
0144(0090) 53 45 52 56 49 43 45 53 2E 2E 37 33 30 20 4E 49
0160(00A8) 5A 41 4D 20 42 4C 4F 43 4B 20 41 4C 4C 41 4D 41
0176(00B0) 20 49 51 42 41 4C 20 54 4F 57 4E 20 20 20 20 20
0192(00C0) 20 20 20 20 20 20 20 20 20 20 20 4C 41 48 4F 52
0208(00D0) 45 2D 50 41 4B 49 53 54 41 4E 2E 2E 50 48 4E
0224(00E0) 45 20 3A 34 33 30 37 39 31 2C 34 34 33 32 34 3B
0240(00F0) 2C 32 38 30 35 33 30 2E 20 20 20 20 20 20 20
```

(c) 1986 Basit & Amjad (put) Ltd.  
BRAIN COMPUTER SERVICES., 730 NI ZAM BLOCK ALLAMA IQBAL TOWN LAHDR E-PAKISTAN., PHJN E :430791,443248 ,280530.

[https://en.wikipedia.org/wiki/Brain\\_\(computer\\_virus\)#/media/File:Brain-virus.jpg](https://en.wikipedia.org/wiki/Brain_(computer_virus)#/media/File:Brain-virus.jpg)



[https://en.wikipedia.org/wiki/Amiga\\_500#/media/File:Amiga500\\_system.jpg](https://en.wikipedia.org/wiki/Amiga_500#/media/File:Amiga500_system.jpg)



# Macroviruses

- Natural evolution in the wild
  - “ON ERROR RESUME NEXT”
- <https://bontchev.nlcv.bas.bg/papers/macidpro.html>

# Where is all of this going?

(From viruses and worms to “flying Trojans”)

- Propagation
  - 0 day exploits
    - In servers, web browsers, other programs...
  - Social engineering, waterhole attacks
  - “Zero-click”
- Command and control
  - Network communication
  - Capabilities on the system
  - Privilege escalation
- Stealth (not leaving tracks)

# Outline of examples

- “Reflections on Trusting Trust”
  - Example of a Trojan Horse
- Cohen
  - Self-replication and self-propagation
- Elk Cloner
  - Stealthy? Targeted?
- Code Red and other worms from the 2000s
  - Infect as many servers as possible, as fast as possible
- Botnets
  - Command and control
- Stuxnet
  - Stealthy and targeted
- Pegasus
  - A “flying Trojan”
- XZ backdoor

# Reflections on Trusting Trust (1984)

- [https://www.cs.cmu.edu/~rdriley/487/papers/Thompson\\_1984\\_ReflectionsonTrustingTrust.pdf](https://www.cs.cmu.edu/~rdriley/487/papers/Thompson_1984_ReflectionsonTrustingTrust.pdf)
- A Trojan Horse is hidden malicious logic in a program or system

```
compile(s)
char *s;
|
|
|   if(match(s, "pattern1")) |
|       compile ("bug1");
|       return;
|
|   if(match(s, "pattern 2")) |
|       compile ("bug 2");
|       return;
|
|   ...
|
```

**FIGURE 3.3.**

# Computer Viruses: Theory and Experiments (1984)

- <https://www.cnsr.ictas.vt.edu/QEpaper/cohen.pdf>
- “Information only has meaning in that it is subject to interpretation”
- Formal undecidability of detecting viruses in the general sense
- Evolution of viruses, definition, *etc.*

[https://en.wikipedia.org/wiki/Apple\\_II](https://en.wikipedia.org/wiki/Apple_II)



# Elk Cloner (1981)

Boot #	Behavior
10th	Overwrote the reset vector so that pressing CONTROL-RESET enters the Monitor program instead of DOS.
15th	Modified the video mode so that the text on the screen was inverted.
20th	Wrote to the speaker, causing a brief click to be heard.
25th	Modified the video mode so that the text on the screen flashed.
30th	Rearranged the characters that represent the file type of a file when the CATALOG command was executed
35th	Modified the value that represented

...

(from <https://arxiv.org/pdf/2007.15759.pdf>)

# Elk Cloner (continued)

	the program instead.)
50th	Modified the reset vector so that pressing CONTROL-RESET caused the Elk Cloner poem to be displayed.
55th	Modified a constant in the diskette calibration code, causing the sound the disk calibration process made during the boot process to change. [4]
60th	Same as the 55th boot except that a different value was written to the constant in the disk calibration code.
65th	Overwrote the first instruction of the DOS command handler with a jump to the Monitor routine, so that the disk booted into the Monitor.
70th	Same as the 55th boot except that a different

...

(from <https://arxiv.org/pdf/2007.15759.pdf>)



# Elk Cloner poem

ELK CLONER :

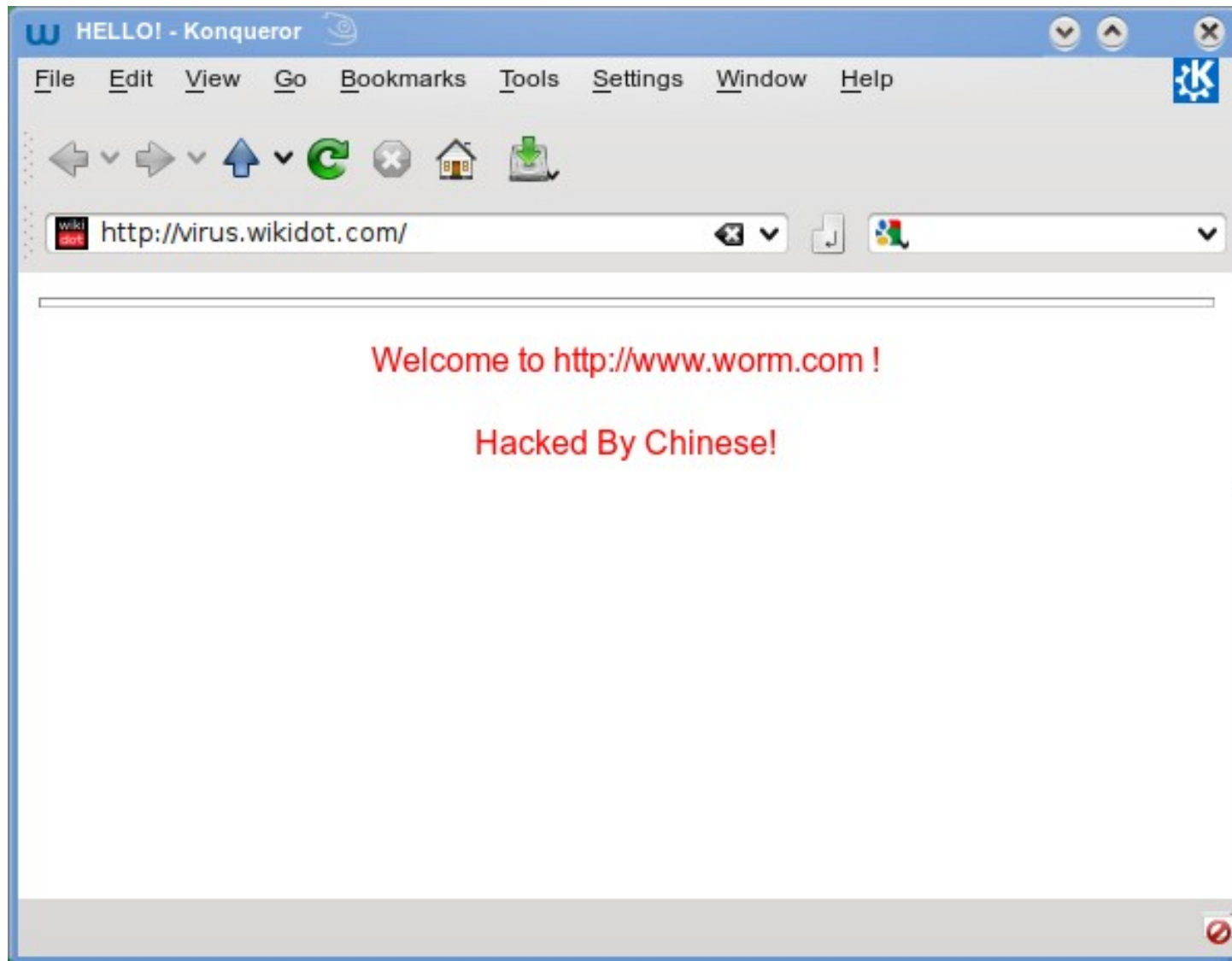
THE PROGRAM WITH A PERSONALITY

IT WILL GET ON ALL YOUR DISKS  
IT WILL INFILTRATE YOUR CHIPS  
YES IT'S CLONER!

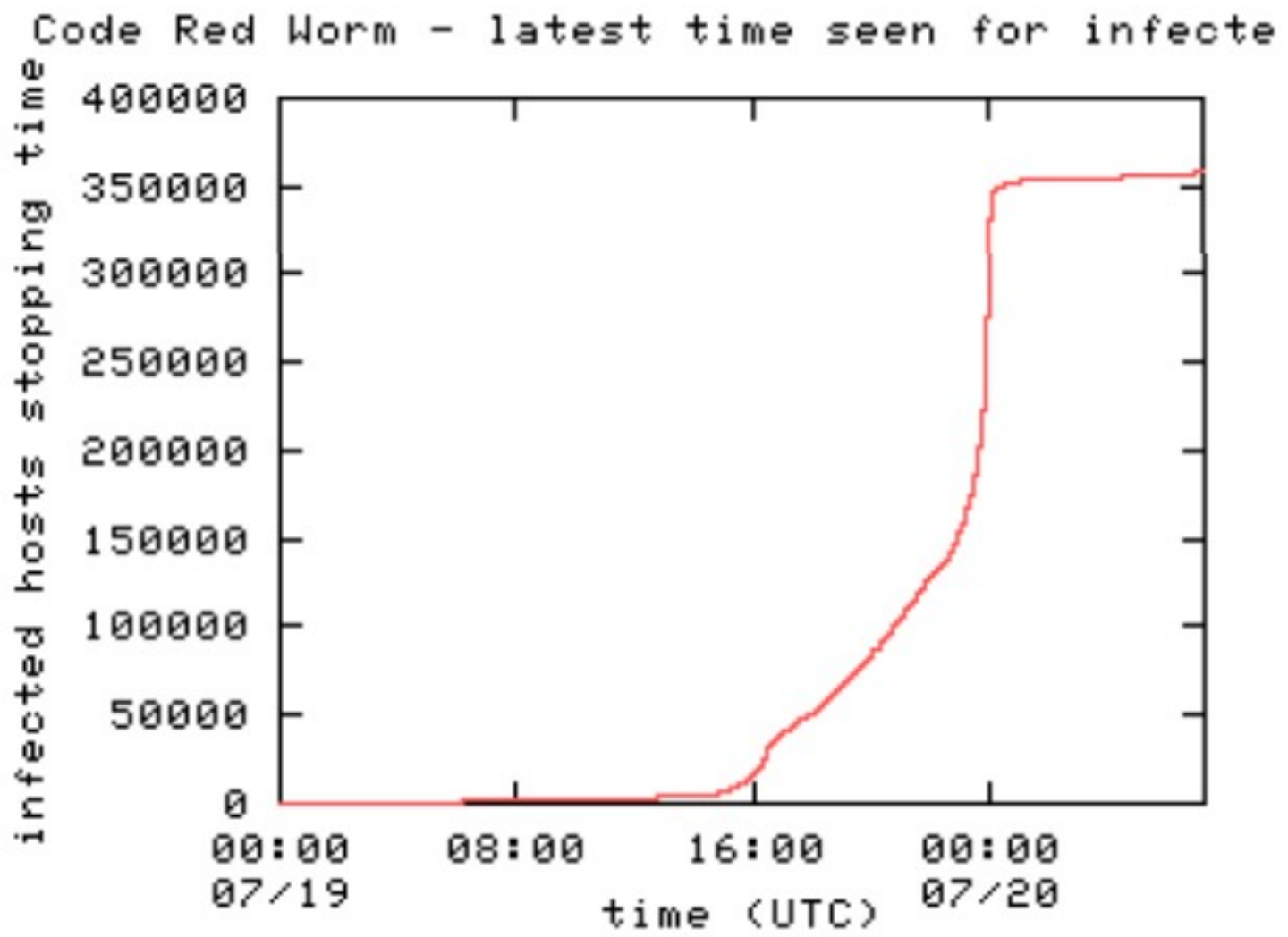
IT WILL STICK TO YOU LIKE GLUE  
IT WILL MODIFY RAM TOO  
SEND IN THE CLONER!



<https://www.cybereason.com/blog/what-is-code-red-worm>



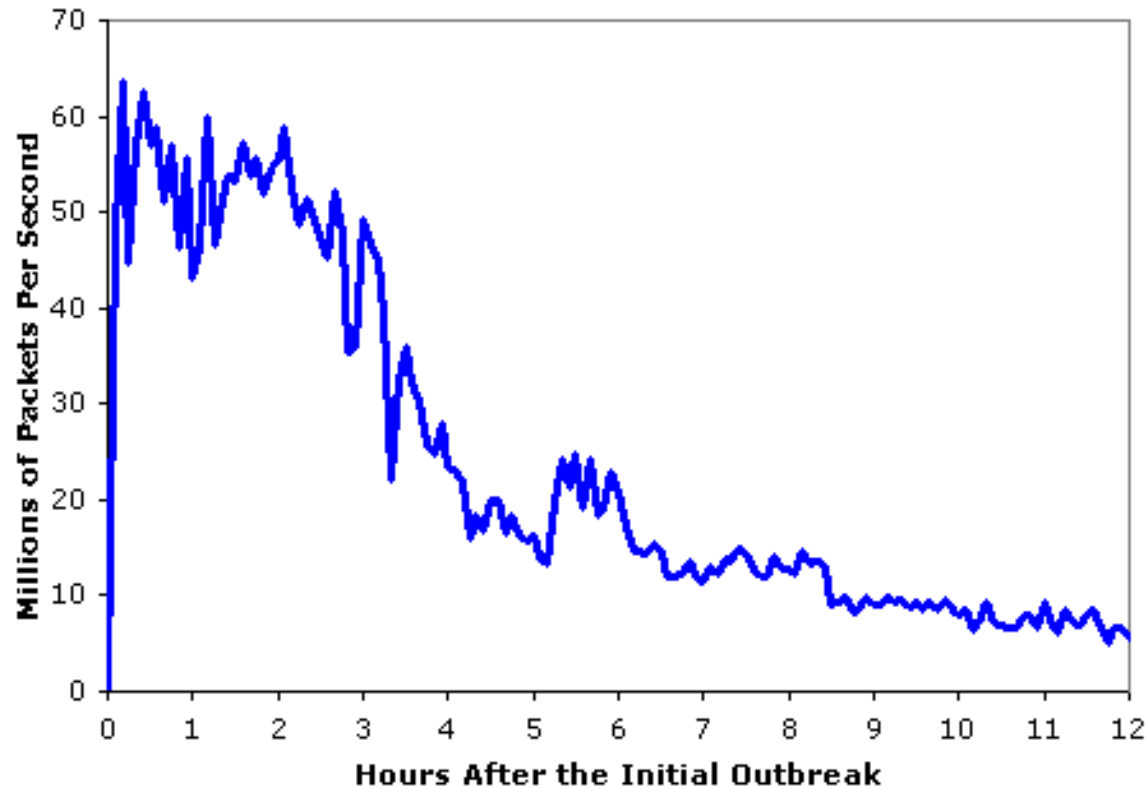
# Code Red



From: <https://www.cs.ucf.edu/~czou/research/codered.pdf>

# Slammer (2003)

Aggregate Scans/Second in the 12 Hours  
After the Initial Outbreak



Over 75K machines in 10 minutes.

(From: [https://www.caida.org/catalog/papers/2003\\_sapphire/](https://www.caida.org/catalog/papers/2003_sapphire/))

# Witty Worm (2004)

```
rand(){
  # Note that 32-bit integers obviate the need for
  # a modulus operation here.
  X = X * 214013 + 2531011;
  return X; }
srand(seed){ X = seed; }
main(){
1.   srand(get_tick_count());
2.   for (i=0; i < 20,000; ++i)
3.       dest_ip ← rand()[0...15] || rand()[0...15];
4.       dest_port ← rand()[0...15];
5.       packet_size ← 768 + rand()[0...8];
6.       packet_contents ← top of stack;
7.       sendto();
8.   if(open(physicaldisk, rand()[13...15]))
9.       overwrite_block(rand()[0...14] || 0x4e20);
10.  goto 1;
11.  else goto 2; }
```

Figure 2: Pseudocode of the Witty worm

# Botnets (mid-2000s)

- Early command-and-control was based on IRC and dynamic DNS
  - Easy to take down
- Switched to fast-flux
  - Peer-to-peer, load balancing, redirection
- Today's C&C is more sophisticated, and there is an entire market surrounding botnets



# Stuxnet (discovered 2010)





# Stuxnet

- Attacked the Iranian nuclear program
- Multiple ways of spreading
- Attempt to limit spread, several attempts
- Not as buggy as typical malware
- Attacked very specific centrifuges with a very specific frequency

<https://en.wikipedia.org/wiki/Stuxnet>

# Pegasus spyware (released 2016)

- [https://en.wikipedia.org/wiki/Pegasus\\_\(spyware\)](https://en.wikipedia.org/wiki/Pegasus_(spyware))
- NSO group
- “Flying Trojan”



[https://en.wikipedia.org/wiki/Trojan\\_Horse#/media/File:RomanVirgilFolio101r.jpg](https://en.wikipedia.org/wiki/Trojan_Horse#/media/File:RomanVirgilFolio101r.jpg)



[https://en.wikipedia.org/wiki/Pegasus#/media/File:Bellerophon\\_riding\\_Pegasus\\_and\\_killing\\_the\\_Chimera,\\_Roman\\_mosaic,\\_the\\_Rolin\\_Museum\\_in\\_Autuñ,\\_France,\\_2nd\\_to\\_3rd\\_century\\_AD.jpg](https://en.wikipedia.org/wiki/Pegasus#/media/File:Bellerophon_riding_Pegasus_and_killing_the_Chimera,_Roman_mosaic,_the_Rolin_Museum_in_Autuñ,_France,_2nd_to_3rd_century_AD.jpg)

# Pegasus

- Supposedly for law enforcement, antiterrorism efforts, *etc.*
- Often used against civil society
  - Full control of the infected system, including calls, microphone, camera, messages, passwords, files, *etc.*
  - Can be used to plant evidence
- Often delivered *via* sophisticated zero-click zero-day exploits

# Pegasus examples

- Ahmed Mansoor in 2016 (first technical analysis of Pegasus by the Citizen Lab and Lookout Security)
  - <https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/>
- Many more examples from Mexico, Saudi Arabia, Bahrain, Jordan, and more...
  - <https://citizenlab.ca/tag/pegasus/>
- Bhima Koregaon 16
  - <https://www.arsenalexperts.com/>
  - <https://netalert.me/bhima-koregaon.html>

# Targeted threats

- Stealthy, targeted, sophisticated (socially and/or technically), well-resourced
- Different methods of delivery
  - Social engineering (targeted email)
  - Waterholing attacks
  - MiTM attacks (I expect this to be a future trend)
- Threat to civil society all over the world
  - See, e.g., <https://tibcert.org/>

<https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/hardy>

From Cheng Li <chengli.brookings@aol.com>

Reply Reply All Forward Archive Junk Delete

Subject: Happy Tib Losar and Ask You a Favour

2012-02-23 02:00

To: [REDACTED]

Dear [REDACTED]

I am Cheng Li from John L. Thornton China Center of Brookings. I will attend a annual meeting on Religious Research with CIIS in Shanghai next week, plan to take the chance to visit Tibet. Attached is a list of Tibetans who have self-immolated from 2009 which my assistant prepared for me, but I am not sure of its accuracy. Would you please have a look and make necessary corrections. I will be really much appreciated if you could do me the favor and offer some more information about the latest happenings inside tibet.

Thank you again and happy Tib losar!

Cheng Li  
Director of Research, John L. Thornton China Center  
Brookings Institution

1 attachment: list\_of\_self\_immolations.xls 116.5 KB

# ESET Research: Chinese-speaking Evasive Panda group spreads malware via updates of legitimate apps and targets NGO in China

Listed under: [ESET Research](#)



Next story



Editor  
26 Apr 2023

- Users in mainland China at an international NGO were targeted with malware delivered through updates for software developed by Chinese companies.
- With high confidence, we attribute this activity to the Chinese-speaking Evasive Panda APT group.
- The backdoor MgBot is used for cyberespionage.

## Other Research

[ESET Research dives into the onboarding and scamming processes of Telekopye online fraudsters](#)

---

[ESET Research: Chinese-speaking Evasive Panda group spreads malware via updates of legitimate apps and targets NGO in China](#)



# Wildberries...

## Russian Trusted Root CA

Identity: Russian Trusted Root CA

Verified by: Russian Trusted Root CA

Expires: 02/27/2032

### ▼ Details

#### Subject Name

C (Country): RU

O (Organization): The Ministry of Digital Development and Communications

CN (Common Name): Russian Trusted Root CA

#### Issuer Name

C (Country): RU

O (Organization): The Ministry of Digital Development and Communications

CN (Common Name): Russian Trusted Root CA

#### Issued Certificate

Version: 3

Serial Number: 10 00

Not Valid Before: 2022-03-01

Not Valid After: 2032-02-27

#### Certificate Fingerprints

SHA1: 8F F9 15 CC AB 7B C1 6F 8C 5C 80 99 D5 3E 0E 11 5B 3A EC 2F

MD5: 7F BB 1F BB D1 29 47 E7 28 DC BF A4 56 8C 64 CD

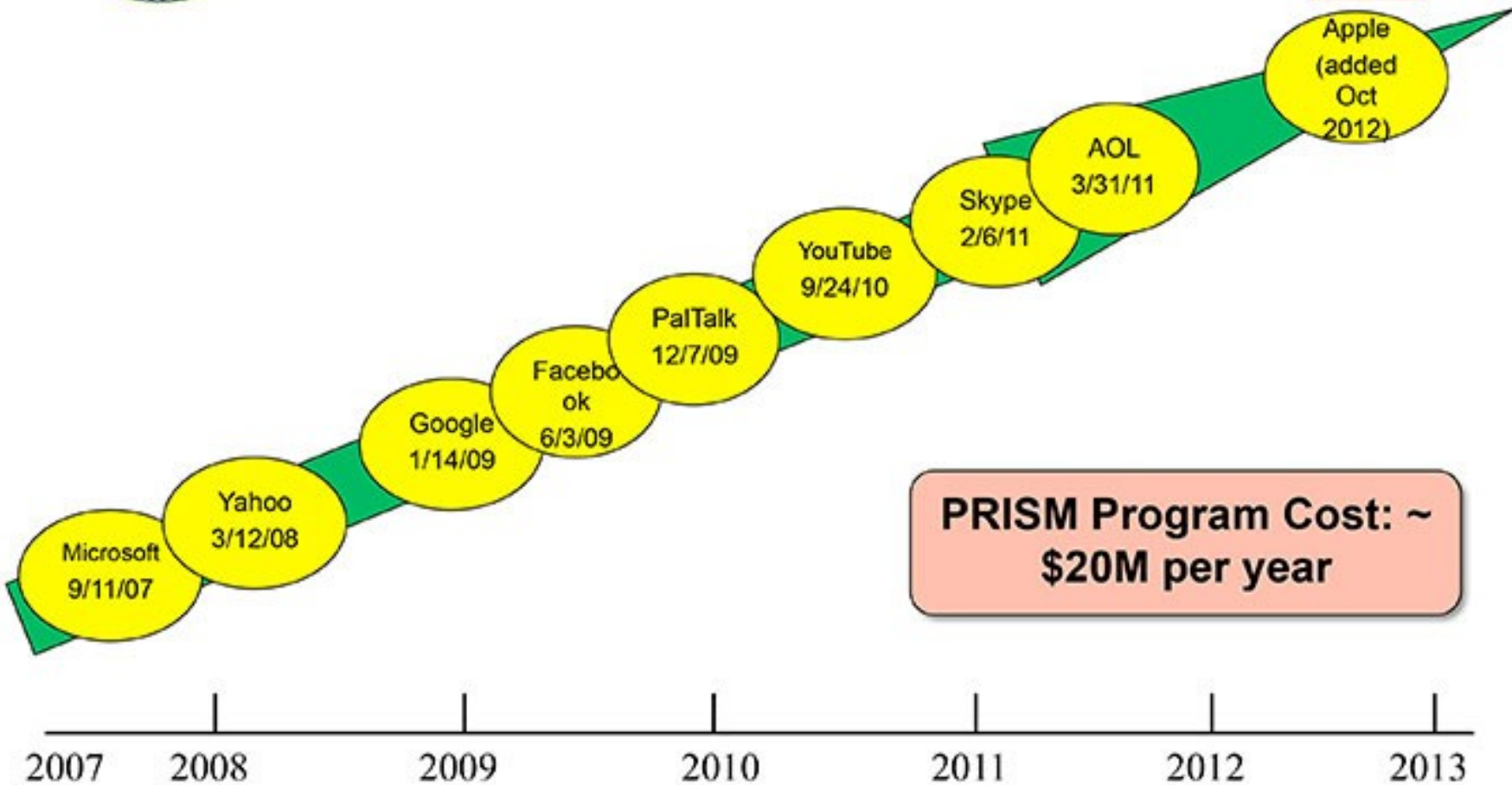


# Unspecified telco app...

- Many cell phones come with apps preinstalled by the telco
- Many such apps in a particular region of the world contain a Software Development Kit (SDK) to save the telco money
  - If you try to dial the phone number of the telco's tech support, it will redirect you to an Internet IP address instead (IP PBX)
- List of phone number to IP mappings comes signed by the vendor of the SDK



# (TS//SI//NF) Dates When PRISM Collection Began For Each Provider



**PRISM Program Cost: ~ \$20M per year**

<https://arstechnica.com/security/2024/04/what-we-know-about-the-xz-utils-backdoor-that-almost-infected-the-world/>

# XZ Outbreak (CVE-2024-3094)



XZ Utils is a collection of open-source tools and libraries for the XZ compression format, that are used for high compression ratios with support for multiple compression algorithms, notably LZMA2.

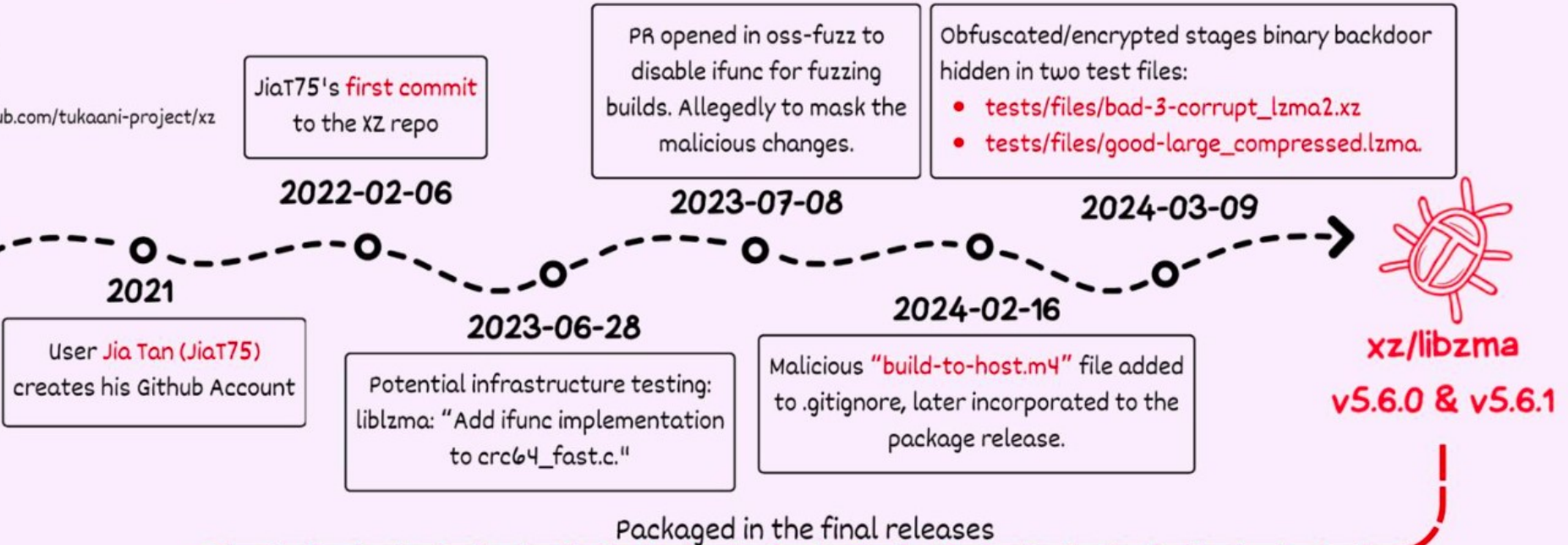


On Friday 29th of March, Andres Freund (principal software engineer at Microsoft) emailed oss-security informing the community of the discovery of a backdoor in xz/liblzma version 5.6.0 and 5.6.1.



## Github Activity Summary (user: JiaT75)

Repository: <https://github.com/tukaani-project/xz>



m4/build-to-host.m4

tests/files/bad-3-corrupt\_lzma2.xz  
Substitution to uncorrupt





Packaged in the final releases

## m4/build-to-host.m4

The M4 macro is executed during the build process and runs the malicious code below.



```
...
63 gl_[${1}]_config='sed \"r\n\" $gl_am_configmake |
eval $gl_path_map | $gl_[${1}]_prefix -d 2>/dev/null'
...
95 gl_path_map='tr "\t \-\" \" \t_\-\"'
...
```

Read Bytes

### tests/files/bad-3-corrupt\_lzma2.xz

Substitution to uncorrupt malformed xz file

- 0x09 (\t) are replaced with 0x20
- 0x20 (whitespace) are replaced with 0x09
- 0x2d (-) are replaced with 0x5f
- 0x5f (\_) are replaced with 0x2d



**\*Uncorrupted\***  
bad-3-corrupt\_lzma2.xz

## Stage 1 - Bash File

v5.6.0

- Bytes in comment: 86 F9 5A F7 2E 68 6A BC
- Custom substitution (byte value mapping)

v5.6.1

- Bytes in comment: E5 55 89 B7 24 04 D8 17
- Check if script running on Linux
- Custom substitution (byte value mapping)

### tests/files/good-large\_compressed.lzma

1. Decompress the file with `xz -dc`
2. Remove junk data from the file using multiple `head` tool calls
3. Portion of the file is discarded (contains the binary backdoor)
4. Use custom substitution cipher to decipher the data
5. Deciphered data is decompressed using `xz -F raw --lzma1 -dc`



Bash script

## Stage 2 - Bash File



## Stage 2 - Bash File

### v5.6.0 Backdoor extraction

An .o file extracted & integrated into compilation/linking

1. Extract & decipher `tests/files/good-large_compressed.lzma`
2. Manipulate output with: `LC_ALL=C sed "s/(.)/\1n/g"`
3. Decrypt using AWK script (RC4-like)
4. Decompress with `xz -dc --single-stream`
5. Binary backdoor stored as `liblzma_la-crc64-fast.o`

**liblzma\_la-crc64-fast.o is then added to the compilation/linking process!**



### v5.6.1 Extension Mechanism

1. Search Files: use `grep -broaF` in `tests/files/` for signatures:

a. `"~!:_w", "|_{-"`      output: `"file_name:offset:signature"`

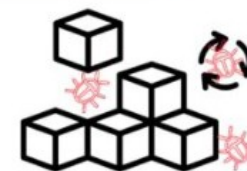
b. `"jv!.^%", "%R.lZ"`

2. If Found:

- a. Save first offset + 7 as \$start
- b. Save second file's offset as \$end

3. Next Steps:

- a. Merge found segments
- b. Decipher with custom byte mapping
- c. Decompress & execute data



No files with the signatures were found, however it highlights the framework's potential modularity for future updates

**@FRÖGGER\_**  
**THOMAS ROCCIA**

# This semester in a nutshell...

- Even after quantum computers are built, Alice and Bob can communicate securely over the Internet, but...
  - There are vulnerabilities in the crypto and software
    - Side channels make solving this especially hard
- If Alice is using an unrooted Android device and Bob is in cahoots with the government, things get really bleak for civil society
  - Most Internet traffic looks something like this

# But there is hope...

- Deep Packet Inspection can be evaded
- Trust relationships can be investigated
  - *E.g.*, TLS certificates and DNS records
- Awareness can lead to change
  - New protocols (IETF, IRTF)
  - New laws and policies
  - New user behaviors



# Join those fighting for Internet freedom!

- <https://censorbib.nymity.ch/>
- <https://apply.opentech.fund/>
- <https://github.com/net4people/bbs>
- <https://www.torproject.org/>
- <https://ooni.org/>
- <https://ntc.party/>
- <https://censoredplanet.org/>
- <https://netalert.me/>
- <https://citizenlab.ca/>

# Conferences you should check out

- IEEE Symposium on Security and Privacy (Oakland)
- USENIX Security Symposium
  - Also check out the workshops like FOCI and WOOT
- ACM Conference on Computer and Communications Security (CCS)
- Network and Distributed System Security Symposium (NDSS)
- Privacy-Enhancing Technologies Symposium (PETS)
  - Also PoPETS
- Also RAID for intrusion detection, DFRWS for forensics, CSF for policy and theory, Eurocrypt and Crypto, Blackhat, DEFCON, phrack, 2600 magazine, WPES and WEIS

# More resources

- *Cryptovirology* by Young and Yung
- *The Art of Computer Virus Research and Defense* by Szor
- *Practical Malware Analysis* by Honig and Sikorski
- <http://www.forensicswiki.org/wiki/Tools>