HY-457: Assignment 2

Implementation of a Ransomware Protection Software Suite

Papadogiannakis Manos
papamano@csd.uoc.gr

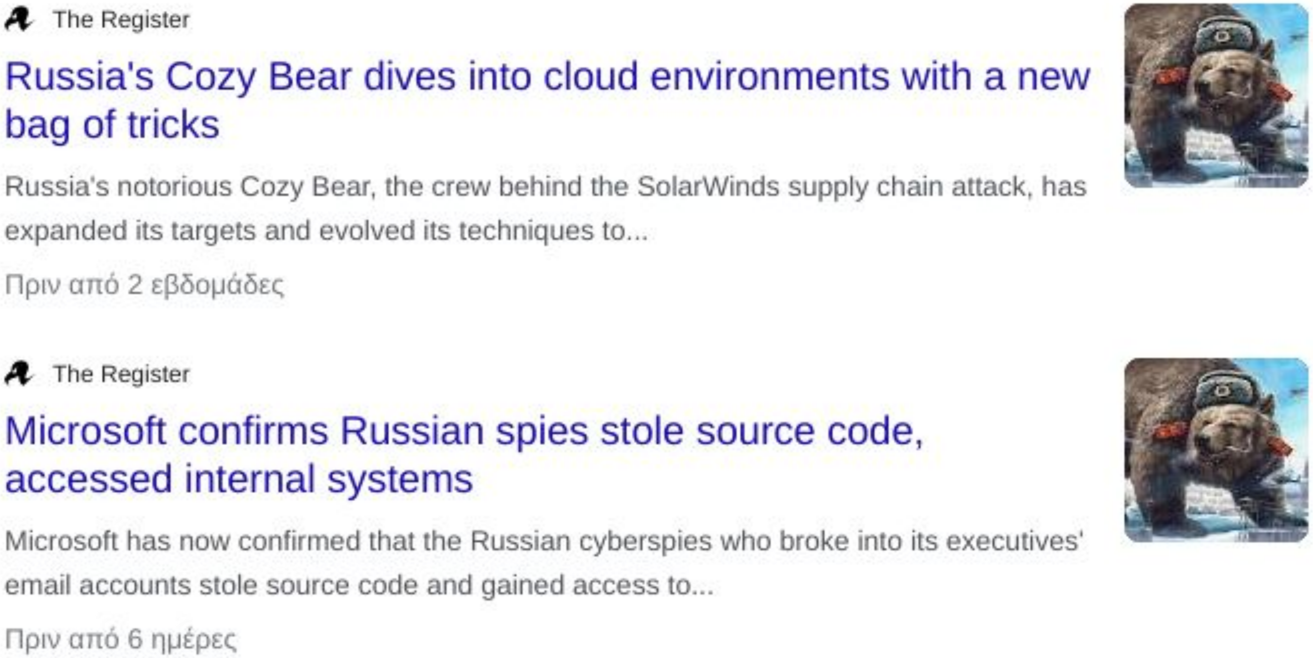
CS-457: Introduction to Information Security Systems
Computer Science Department
University of Crete

Outline

0. Motivating Scenario
1. Scanning for Infected Files
2. Detecting Potential Harmful Network Traffic
3. Securing Valuable Files
4. Protecting from Unauthorized Access
5. Notes

0. Motivation

Motivation



The Register

Russia's Cozy Bear dives into cloud environments with a new bag of tricks

Russia's notorious Cozy Bear, the crew behind the SolarWinds supply chain attack, has expanded its targets and evolved its techniques to...

Πριν από 2 εβδομάδες

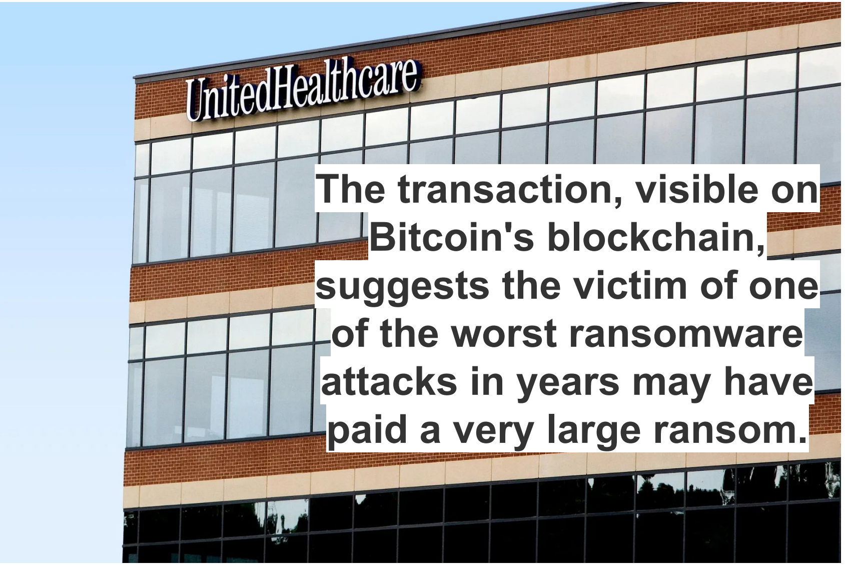
The Register

Microsoft confirms Russian spies stole source code, accessed internal systems

Microsoft has now confirmed that the Russian cyberspies who broke into its executives' email accounts stole source code and gained access to...

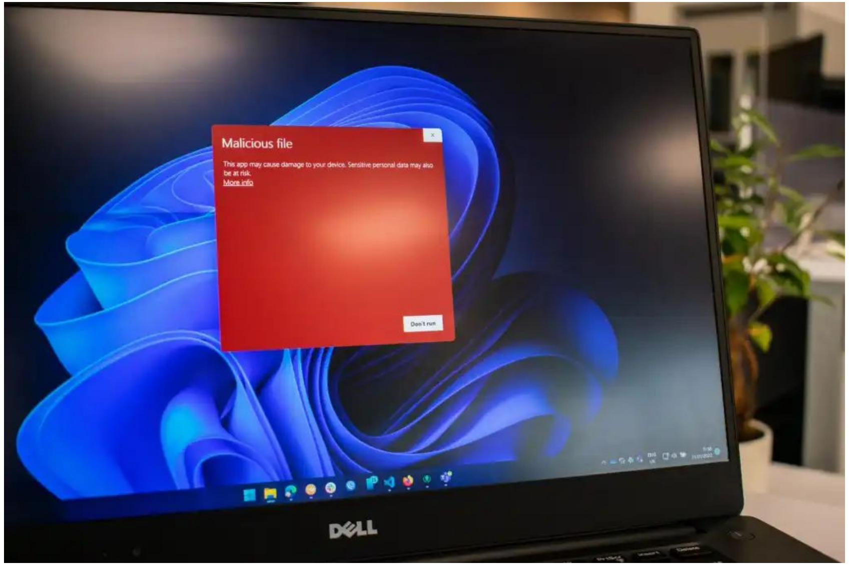
Πριν από 6 ημέρες

Motivation



The transaction, visible on Bitcoin's blockchain, suggests the victim of one of the worst ransomware attacks in years may have paid a very large ransom.

Hackers Behind the Change Healthcare Ransomware Attack Just Received a $22 Million Payment
https://www.wired.com/story/alphv-change-healthcare-ransomware-payment/



Malicious file

This app may cause damage to your device. Sensitive personal data may also be at risk.

Windows includes built-in ransomware protection. Here's how to turn it on
https://www.pcworld.com/article/2245853/how-to-turn-on-microsoft-windows-ransomware-protection.html

Motivation

- Hypothetical attack at a corporate infrastructure

- Discover files already infected with malware
- Discover ransomware that locks files

1. Scanning for Infected Files

Infected Files

1. Some files have been infected by virus

2. Some files are malicious shared libraries

3. Some files are used by the attackers as utilities

Implementation

- **Goal: Find these files**

- **Search for:**
  a. File Signatures (i.e. specific files)
  b. Virus Signature (i.e. bytes inside files)
  c. Bitcoin Address (i.e. text inside files)

Running

- **Follow execution instructions:**

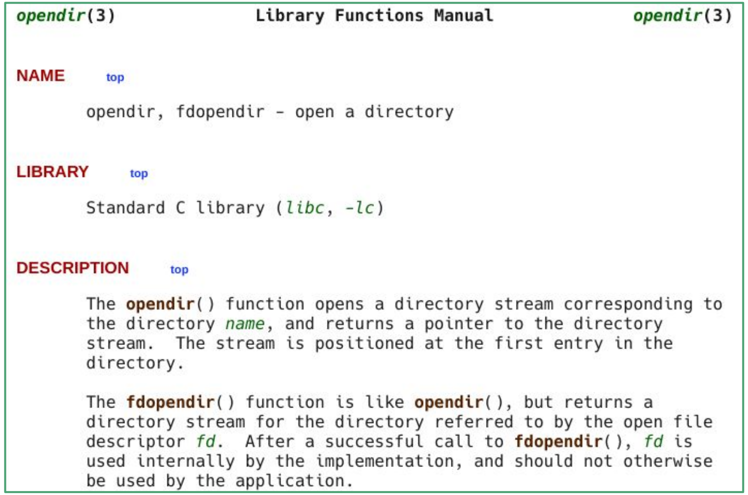
```
$ ./antivirus scan /home/ceo/Downloads
```

- **Need to handle all files in the given directory**

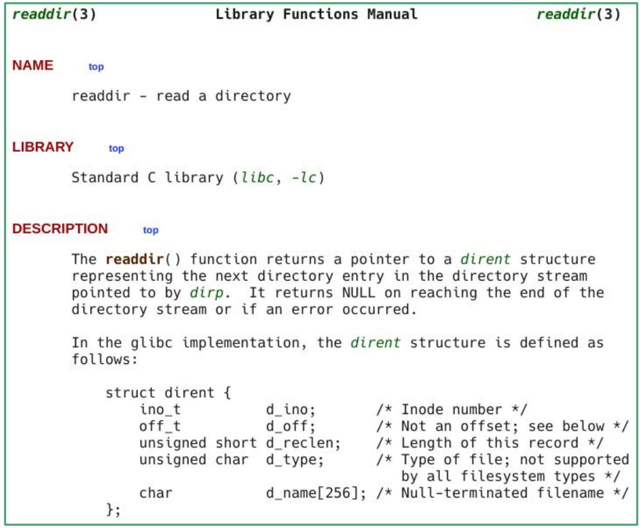
- **What if files are binary?**
  - Still need to search for strings inside the files

Subdirectories

- **Need to parse subdirectories as well**
  - Sounds like recursion!



```
opendir(3)              Library Functions Manual              opendir(3)

NAME     top

       opendir, fdopendir - open a directory

LIBRARY     top

       Standard C library (libc, -lc)

DESCRIPTION     top

       The opendir() function opens a directory stream corresponding to
       the directory name, and returns a pointer to the directory
       stream.  The stream is positioned at the first entry in the
       directory.

       The fdopendir() function is like opendir(), but returns a
       directory stream for the directory referred to by the open file
       descriptor fd.  After a successful call to fdopendir(), fd is
       used internally by the implementation, and should not otherwise
       be used by the application.
```



```
readdir(3)              Library Functions Manual              readdir(3)

NAME     top

       readdir - read a directory

LIBRARY     top

       Standard C library (libc, -lc)

DESCRIPTION     top

       The readdir() function returns a pointer to a dirent structure
       representing the next directory entry in the directory stream
       pointed to by dirp.  It returns NULL on reaching the end of the
       directory stream or if an error occurred.

       In the glibc implementation, the dirent structure is defined as
       follows:

           struct dirent {
               ino_t          d_ino;       /* Inode number */
               off_t          d_off;       /* Not an offset; see below */
               unsigned short d_reclen;    /* Length of this record */
               unsigned char  d_type;      /* Type of file; not supported
                                              by all filesystem types */
               char           d_name[256]; /* Null-terminated filename */
           };
```

Hash Generators

- **Need to compute the hash value of all files**
  - Acts like a fingerprint or file signature
- **Allowed to use OpenSSL**
  - No need to reinvent the wheel
- **Read the docs!**
  - MD5, SHA256

Needle in Haystack

- **How do I search for specific bytes inside the file?**
  - Simply go over the file’s content byte-by-byte
- **How do I search for strings inside the file?**
  - Naive approach would be strstr
- **What about binary files?**
  - Can extract sequences of **printable** characters

2. Detecting Potential Harmful Network Traffic

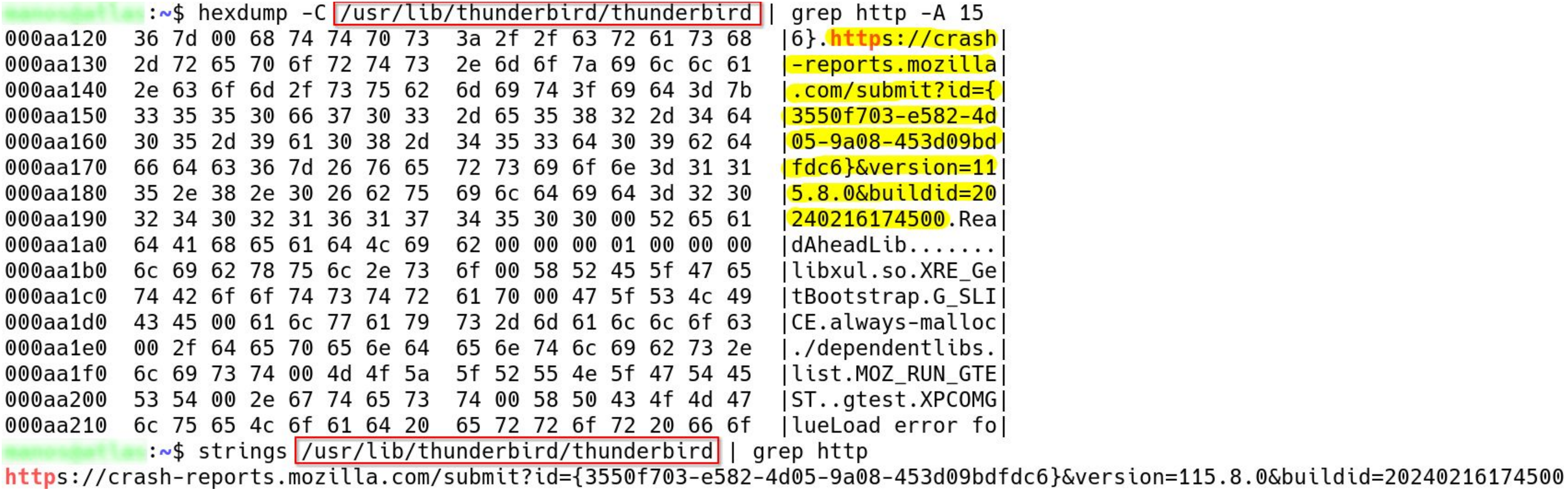
Network Traffic

- Programs connect to the Web/Internet and exchange data

- Often the addresses are **hardcoded**

- We can extract them and know who an application talks to

Hardcoded Addresses



```
[redacted]:~$ hexdump -C /usr/lib/thunderbird/thunderbird | grep http -A 15
000aa120  36 7d 00 68 74 74 70 73  3a 2f 2f 63 72 61 73 68  |6}.https://crash|
000aa130  2d 72 65 70 6f 72 74 73  2e 6d 6f 7a 69 6c 6c 61  |-reports.mozilla|
000aa140  2e 63 6f 6d 2f 73 75 62  6d 69 74 3f 69 64 3d 7b  |.com/submit?id={|
000aa150  33 35 35 30 66 37 30 33  2d 65 35 38 32 2d 34 64  |3550f703-e582-4d|
000aa160  30 35 2d 39 61 30 38 2d  34 35 33 64 30 39 62 64  |05-9a08-453d09bd|
000aa170  66 64 63 36 7d 26 76 65  72 73 69 6f 6e 3d 31 31  |fdc6}&version=11|
000aa180  35 2e 38 2e 30 26 62 75  69 6c 64 69 64 3d 32 30  |5.8.0&buildid=20|
000aa190  32 34 30 32 31 36 31 37  34 35 30 30 00 52 65 61  |240216174500.Rea|
000aa1a0  64 41 68 65 61 64 4c 69  62 00 00 00 01 00 00 00  |dAheadLib.......|
000aa1b0  6c 69 62 78 75 6c 2e 73  6f 00 58 52 45 5f 47 65  |libxul.so.XRE_Ge|
000aa1c0  74 42 6f 6f 74 73 74 72  61 70 00 47 5f 53 4c 49  |tBootstrap.G_SLI|
000aa1d0  43 45 00 61 6c 77 61 79  73 2d 6d 61 6c 6c 6f 63  |CE.always-malloc|
000aa1e0  00 2f 64 65 70 65 6e 64  65 6e 74 6c 69 62 73 2e  |./dependentlibs.|
000aa1f0  6c 69 73 74 00 4d 4f 5a  5f 52 55 4e 5f 47 54 45  |list.MOZ_RUN_GTE|
000aa200  53 54 00 2e 67 74 65 73  74 00 58 50 43 4f 4d 47  |ST..gtest.XPCOMG|
000aa210  6c 75 65 4c 6f 61 64 20  65 72 72 6f 72 20 66 6f  |lueLoad error fo|
[redacted]:~$ strings /usr/lib/thunderbird/thunderbird | grep http
https://crash-reports.mozilla.com/submit?id={3550f703-e582-4d05-9a08-453d09bdfdc6}&version=115.8.0&buildid=20240216174500
```

Implementation

- **Need to examine content of files**
  - What if they are binary?
  - Not allowed to use the `strings` utility tool!

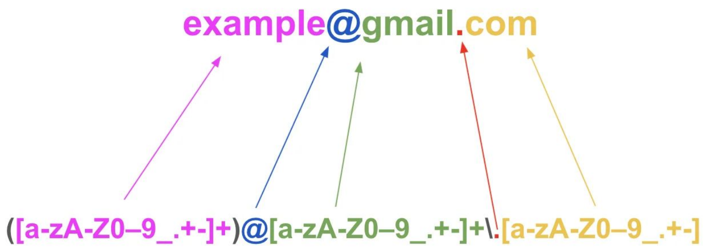
- **Need to extract all strings from file**
  - String == Sequence of printable characters
  - If something is 3 chars long, can it be an address?

Discovering Addresses

- **From previous step we have collected an array of strings**
  - Are they all addresses?

- **Use regular expressions**
  - Free to use any you think is good enough
  - `#include <regex.h>`
  - `$ man -S3 regex`

- **Might need to play around a bit**
  - https://regexr.com/



Malicious Domains

1.1.1.1

FOR FAMILIES

- **We've now formed a list of domains**
  - Are they all malicious?
- **How to tell if a domain is "bad"?**
  - **Cloudflare's** Malware and Adult Content Filter
- **Free DNS resolver**
  - Automatically filters out bad sites
  - "Malware Blocking Only" or "Malware and Adult Content"

Sending Requests

- **How to use?**
  - Send a simple request and handle response
  - No need to parse JSON response

- **Use libcurl for sending requests programmatically**
  - C API
  - Super powerful but you’ll need ~10 LOC

- **Flags you might need:**
  - CURLOPT_URL
  - CURLOPT_HTTPHEADER
  - CURLOPT_WRITEFUNCTION
  - CURLOPT_WRITEDATA

Example

```
$ curl -H "accept: application/dns-json"
‘https://1.1.1.1/dns-query?name=cretalive.gr’
```

```
{
  “Status”: 0,
  “TC”: false,
  ...
  “Answer”: [
    {
      “name”: “cretalive.gr”
      “type”: 1,
      “TTL”: 295,
      “data”: “104.21.54.106”
...
```

```
$ curl -H "accept: application/dns-json"
‘https://family.cloudflare-dns.com/dns-query?name=biawwer.com’
```

```
{
  ...
  “Answer”: [
    {
      “name”: “biawwer.com”
       “type”: 1,
      “TTL”: 60,
      “data”: “0.0.0.0”
...

  “Comment”: [

    “EDE(16) Censored”
```

Cloudflare Endpoints

- **Various endpoints available**

- **Default DNS resolver**
  - https://cloudflare-dns.com/dns-query?name=example.com
  - https://1.1.1.1/dns-query?name=example.com

- **Block malware**
  - https://security.cloudflare-dns.com/dns-query?name=example.com
  - https://1.1.1.2/dns-query?name=example.com

- **Block malware and adult content**
  - https://family.cloudflare-dns.com/dns-query?name=example.com
  - https://1.1.1.3/dns-query?name=example.com

Execution

```
$ ./antivirus scan /home/ceo/Downloads
```

```
[INFO] [9046] [14-Mar-24 13:53:43] Application Started
[INFO] [9046] [14-Mar-24 13:53:43] Scanning directory /home/ceo/
[INFO] [9046] [14-Mar-24 13:53:45] Found 18312 files
[INFO] [9046] [14-Mar-24 13:53:45] Searching…
[INFO] [9046] [14-Mar-24 13:53:55] Operation finished
[INFO] [9046] [14-Mar-24 13:53:55] Processed 18312 files.
```

```
| FILE    | PATH                     | DOMAIN          | EXECUTABLE | RESULT  |
==========================================================================
| foo.exe | /home/ceo/docs/secret | www.google.com     | True  |  Safe    |
| bar.txt | /home/ceo/hy457grade/ | alphaxiom.com      | False |  Malware |
| libd.so | /home/ceo/Desktop/    | https://bbc.com    | False |  Safe    |
| wget.sh | /home/ceo/aws/plugin  | biawwer.com        | True  |  Malware |
```

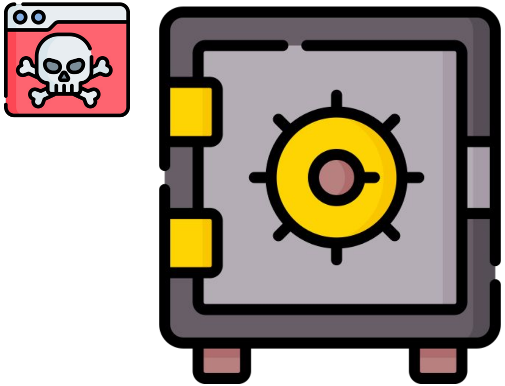
3. Securing Valuable Files

Motivation

- **Need to create a “safe” where we can place important files**

- **A directory that will be constantly monitored**

- **When a ransomware tries to lock our files we will be notified**
    - Monitor filesystem events



Implementation

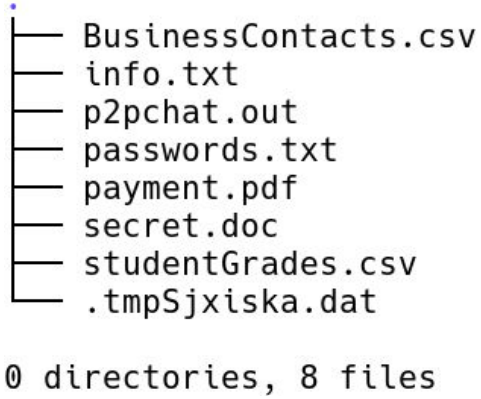
- **Search for specific behavior**
  a. File x is opened
  b. File x.locked is created
  c. File x.locked is stored
  d. File x is deleted

- **Monitor filesystem events using inotify**
  - API that can monitor specific files or entire directories

- **How to test?**
  - Open another terminal and create/modify/delete
  - Create your own dummy ransomware

Example

```
$ antivirus monitor /root/vault/

[INFO] [9046] [14-Mar-24 13:53:43] Application Started
[INFO] [9046] [14-Mar-24 13:53:43] Monitoring directory /root/vault/
[INFO] [9046] [14-Mar-24 13:53:43] Waiting for events...
File ‘info.txt’ was created
File ‘info.txt’ was opened
File ‘info.txt’ that was not opened for writing was closed
File ‘passwords.txt’ was opened
File ‘passwords.txt’ was accessed
File ‘.tmpSjxiska.dat’ was deleted from watched directory
File ‘passwords.txt.locked’ was created
File ‘passwords.txt.locked’ was modified
File ‘passwords.txt.locked’ that was opened for writing was closed
File ‘passwords.txt’ was deleted from watched directory
[WARN] Ransomware attack detected on file passwords.txt
File ‘.tmpSIfwiunew.dat’ was created
File ‘studentGrades.csv’ was opened
```



```
.
├── BusinessContacts.csv
├── info.txt
├── p2pchat.out
├── passwords.txt
├── payment.pdf
├── secret.doc
├── studentGrades.csv
└── .tmpSjxiska.dat

0 directories, 8 files
```

inotify

- An API that monitors filesystem events
- Can monitor individual files or entire directories
- Use can specify which events to monitor
  - e.g. file was accessed
  - directory deleted

inotify

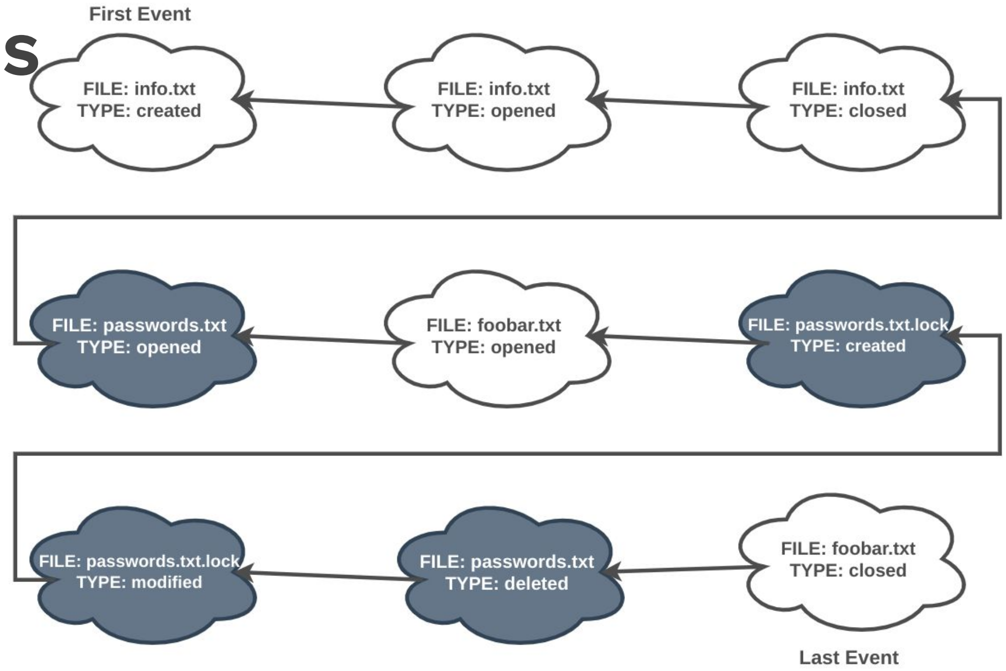
- **Event-based**
  - Need to specify which events to monitor
  - Implemented as bit masks

- **Need to handle events**
  - Use poll() function and while(1) loop
  - If event is X …
    if event is Y …
    if event is Z …

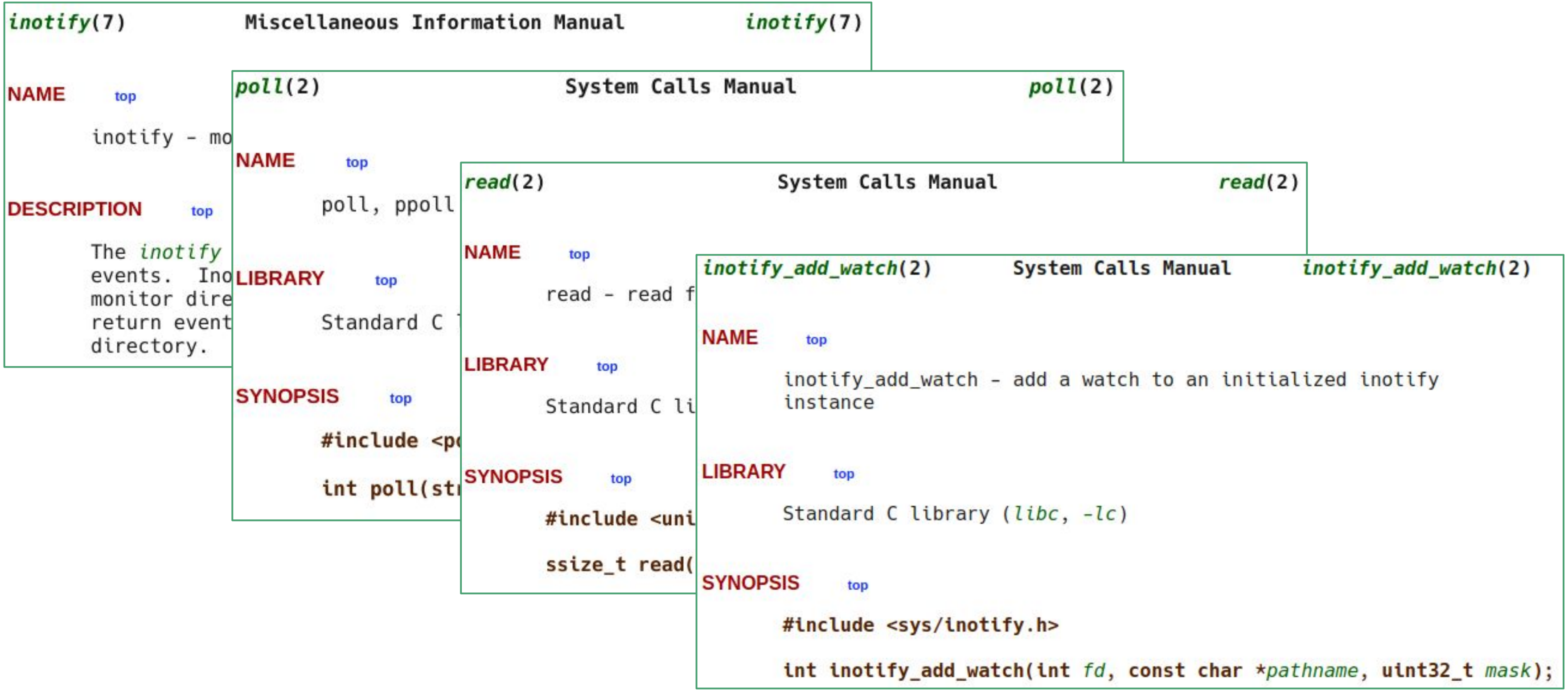
- **Need to remember what has already happened**
  - Need to store previous events

List of Events

- **Need to store previous events**
  - Can convert them to something easier to use
  - E.g. array of strings or array of custom structs



C Standard Library



```
inotify(7)          Miscellaneous Information Manual          inotify(7)

NAME       top

       inotify - mo

DESCRIPTION       top

       The inotify
       events.  Ino
       monitor dire
       return event
       directory.
```

```
poll(2)                    System Calls Manual                    poll(2)

NAME       top

       poll, ppoll

LIBRARY       top

       Standard C l

SYNOPSIS       top

       #include <po

       int poll(str
```

```
read(2)                    System Calls Manual                    read(2)

NAME       top

       read - read f

LIBRARY       top

       Standard C li

SYNOPSIS       top

       #include <uni

       ssize_t read(
```

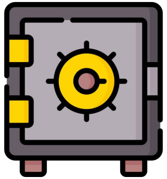
```
inotify_add_watch(2)      System Calls Manual      inotify_add_watch(2)

NAME       top

       inotify_add_watch - add a watch to an initialized inotify
       instance

LIBRARY       top

       Standard C library (libc, -lc)

SYNOPSIS       top

       #include <sys/inotify.h>

       int inotify_add_watch(int fd, const char *pathname, uint32_t mask);
```

4. Protecting From Unauthorized Access

Motivation

- **Place important documents inside a “vault”**
  - Files inside the vault are encrypted and safe

- **No single individual can open the “vault” on their own**



Secret Sharing System

- **Implement a secret sharing mechanism**
  - In this case the secret would be the encryption key
  - No need to implement encryption

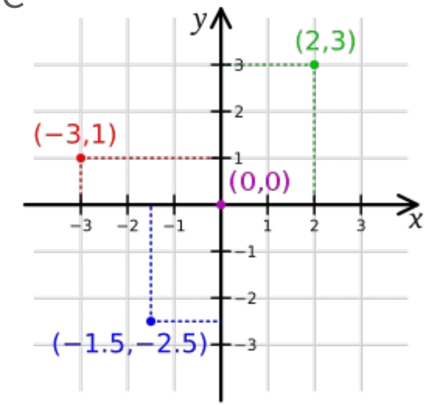
- **Distribute a secret among a group so that the secret cannot be revealed unless X people are present**

Overview

- Assume there are three friends: Alice, Bob and Carol
- The three people will share a secret number “c” by each taking a piece of the number
- Only when all three pieces are presented then all of them are able to reconstruct the secret number “c”

Introduction

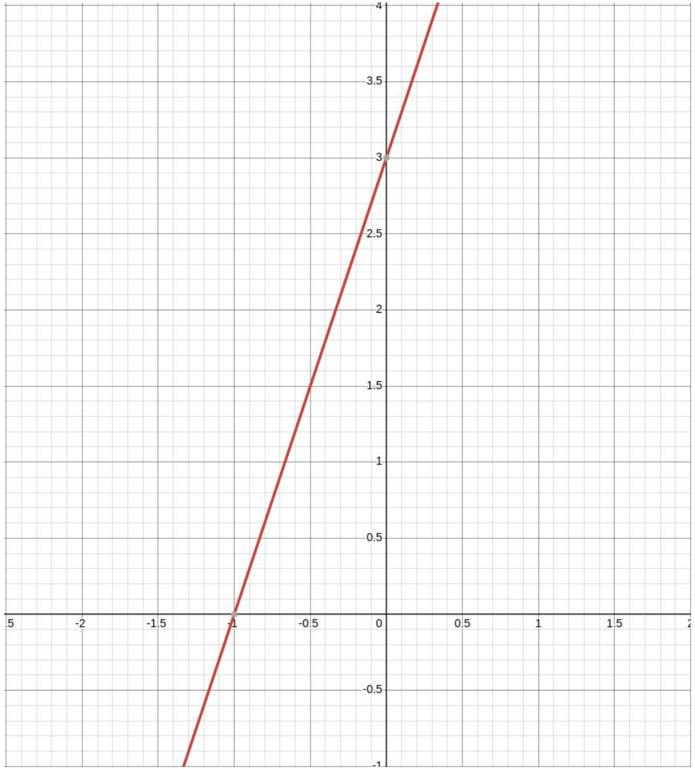
- **How can we achieve this?**
  - Let’s assume the secret we want to share is a Euclidean line
- **How many lines are there?**
  - Infinite
- **What defines a line?**
  - Two points
  - We need to know them to reconstruct the line
- **Can one person on their own reconstruct the line?**



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

Euclidean Plane

$y = 3x + 3$



Simple Example

- **Let's assume that Alice and Bob want to share the secret number 72**
  - Since they are 2, the polynomial degree is 1
- **They randomly choose "a" to be 14 and "b" is the secret number**
  - $f(x) = a \cdot x + b = 14 \cdot x + 72$
- **They calculate $f(1) = 86$ and $f(2) = 100$**
  - Alice gets (1, 86)
  - Bob gets (2, 100)

Simple Example

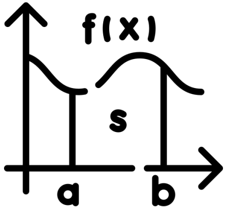
- To **reconstruct** the secret, they present their points and reconstruct the polynomial

$$\left.\begin{array}{l} 86 = a + b \\ 100 = 2a + b \end{array}\right\} \left.\begin{array}{l} a = 86 - b \\ 100 = 2(86 - b) + b \end{array}\right\} 100 = 172 - 2b + b \;\Big\}\; -72 = - b \;\Big\}\; b = 72$$

Implementation

- **Secret Sharing achieved by constructing the polynomial** $f(x) = a \cdot x^2 + b \cdot x + c$

- **Each person will take a point of the polynomial**
  - Alice: (1, f(1))
  - Bob: (2, f(2))
  - Carol: (3, f(3))



- **When all 3 points are presented, they reconstruct the polynomial to retrieve secret number "c"**

Implementation Details

- **Using the slice option and a secret number the program generates the 3 points**

  -
    ```
    $ ./antivirus slice 9
    > (1, 16), (2, 27), (3, 42)
    ```

- **Using the unlock option and the 3 points, the program reconstructs the secret number**

  -
    ```
    $ ./antivirus unlock (1, 16), (2, 27), (3, 42)
    > 9
    ```

- **Generalize your solution for N friends.**
  - When any three present their points they are able to reconstruct the secret

Implementation Details

- **Allowed to look up how to solve system of linear equations with three variables**
  - Our focus is not maths ⇢ Cramer is your friend

- **No need to create parser for unlock function**
  - Use any format you like and assume the input will always be correct

- **Feel free to generate random numbers any way you like**
  - srand, rand
  - /dev/urandom

Advanced Example

- **Select a secret shared number "c"**
  - E.g. c is **9**

- $\mathbf{f(x) = 2\cdot x^2 + 5\cdot x + 9}$ **where a, b were randomly generated**

- **When 3 shares (x_1, x_2, x_3) are present:**
  - $f(x_1) = a\cdot x_1^2 + b\cdot x_1 + c$ $f(1) = 16$ → $16 = a + b + c$ (1)
  - $f(x_2) = a\cdot x_2^2 + b\cdot x_2 + c$ $f(2) = 27$ $27 = 4\cdot a + 2\cdot b + c$ (2)
  - $f(x_3) = a*x_3^2 + b\cdot x_3 + c$ $f(3) = 42$ $42 = 9\cdot a + 3\cdot b + c$ (3)

Advanced Example

$$\begin{aligned} 16 &= a + b + c && (1)\\ 27 &= 4\cdot a + 2\cdot b + c && (2)\\ 42 &= 9\cdot a + 3\cdot b + c && (3)\end{aligned} \quad \xrightarrow{(3) - 3\cdot(1)} \quad \begin{aligned} 42 &= 9\cdot a + 3\cdot b + c\\ \underline{-3\cdot 16} &\underline{= 3\cdot a + 3\cdot b + 3\cdot c}\\ -6 &= 6\cdot a + 0 - 2\cdot c \;\Leftrightarrow\; \mathbf{a = (2\cdot c - 6)/6}\end{aligned}$$

$$\begin{aligned} 16 &= a + b + c && (1)\\ 27 &= 4\cdot a + 2\cdot b + c && (2)\\ 42 &= 9\cdot a + 3\cdot b + c && (3)\end{aligned} \quad \xrightarrow{(2) - 2\cdot(1)} \quad \begin{aligned} 27 &= 4*a + 2*b + c\\ \underline{-2*16} &\underline{= 2*a + 2*b + 2*c}\\ -5 &= 2*a - c \;\Leftrightarrow\; \mathbf{a = (c - 5)/2}\end{aligned}$$

Advanced Example

- **We have computed that:**
  - $a = (2 \cdot c - 6)/6$
  - $a = (c - 5)/2$

- $\mathbf{(2 \cdot c - 6)/6 = (c-5)/2 \Leftrightarrow c-5 = \tfrac{2}{3} \cdot c - 2 \Leftrightarrow \tfrac{1}{3}\, c = 3 \Leftrightarrow c = 9}$

Notes

Notes

- **Your final implementation should be a single executable file**
  - Many source code files

- **Follow the execution instructions**
  - e.g. CLI arguments, arguments order

- **Allowed to use mentioned libraries**

Optional Task

- **There is an optional task**
  - Bonus +1 point (maximum)

- **Need to write a simple YARA rule for the hypothetical attack**

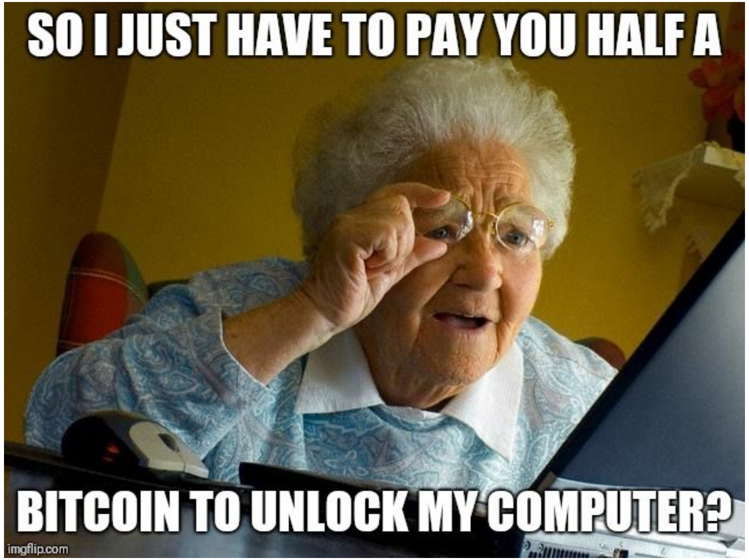
- **Use a tool to generate test files based on YARA rules**

YARA Rule

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true

    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

    condition:

        $a or $b or $c

}
```

This rule is telling YARA that any file containing one of the three strings must be reported as silent_banker.

https://virustotal.github.io/yara/


SO I JUST HAVE TO PAY YOU HALF A
BITCOIN TO UNLOCK MY COMPUTER?
imgflip.com

Credit

Icons from FlatIcon, made by Freepik

Thank You!

✉

papamano@csd.uoc.gr

Questions?