

Introduction to Cybersecurity

A brief introduction

Naavin Ravinthran

MUMTEC Cybersecurity SIG

February 26, 2022

First of all...

- ▶ What do *you* think about cybersecurity?

First of all...

- ▶ What do *you* think about cybersecurity?
- ▶ View poll at
<https://app.sli.do/event/dEmQ7GxJkGeBkfbzDP42B2J>

First of all...

- ▶ What do *you* think about cybersecurity?
- ▶ View poll at
<https://app.sli.do/event/dEmQ7GxJkGeBkFzDP42B2J>
- ▶ ... Or visit [sli.do](https://www.sli.do) and enter the code #899414

First of all...

- ▶ What do *you* think about cybersecurity?
- ▶ View poll at
<https://app.sli.do/event/dEmQ7GxJkGeBkFzDP42B2J>
- ▶ ... Or visit [slido.com](https://www.slido.com) and enter the code #899414
- ▶ P.S: You can ask me questions here too!

A bit about myself

- ▶ My name is Naavin.

A bit about myself

- ▶ My name is Naavin.
- ▶ Second Year Second Semester student.

A bit about myself

- ▶ My name is Naavin.
- ▶ Second Year Second Semester student.
- ▶ I like all things to do with cybersecurity :)

A bit about myself

- ▶ My name is Naavin.
- ▶ Second Year Second Semester student.
- ▶ I like all things to do with cybersecurity :)
- ▶ I did a bit of SIG cybersecurity events last semester too. Hosted workshops, did CTFs, made cybersecurity “challenges”.

A bit about myself

- ▶ My name is Naavin.
- ▶ Second Year Second Semester student.
- ▶ I like all things to do with cybersecurity :)
- ▶ I did a bit of SIG cybersecurity events last semester too. Hosted workshops, did CTFs, made cybersecurity “challenges”.
- ▶ I tried to keep this introduction as non-technical as I can.

Cryptography

A CRYPTO NERD'S IMAGINATION:

HIS LAPTOP'S ENCRYPTED.
LET'S BUILD A MILLION-DOLLAR
CLUSTER TO CRACK IT.

BLAST! OUR
EVIL PLAN
IS FOILED!

NO GOOD! IT'S
4096-BIT RSA!



WHAT WOULD ACTUALLY HAPPEN:

HIS LAPTOP'S ENCRYPTED.
DRUG HIM AND HIT HIM WITH
THIS \$5 WRENCH UNTIL
HE TELLS US THE PASSWORD.

GOT IT.



Cryptography

- ▶ Has a lot of mathematics, particularly number theory.
- ▶ Isn't exclusive to just Encryption/Decryption. Besides confidentiality, cryptography can also provide *Integrity* and *Authentication*, meaning ways to verify data wasn't tampered with in any way (integrity), and that data was proved to be from a particular entity (authentication).

Cryptography

- ▶ Has a lot of mathematics, particularly number theory.
- ▶ Isn't exclusive to just Encryption/Decryption. Besides confidentiality, cryptography can also provide *Integrity* and *Authentication*, meaning ways to verify data wasn't tampered with in any way (integrity), and that data was proved to be from a particular entity (authentication).
- ▶ Motivation: We want to keep things secret to keep our privacy.
- ▶ Has a long and interesting history, probably as long as written word itself.

Cryptography (cont.)

- ▶ Julius Caesar, a famous roman general used the Caesar Cipher.
- ▶ This is a substitution cipher where each letter is “shifted” by a fixed number, and replaced with a different letter.
- ▶ For a shift by one, the letter 'A' becomes letter 'B', and the letter 'B' becomes the letter 'C', ... , and the letter 'Z' becomes the letter 'A'.

Cryptography (cont.)

- ▶ Julius Caesar, a famous roman general used the Caesar Cipher.
- ▶ This is a substitution cipher where each letter is “shifted” by a fixed number, and replaced with a different letter.
- ▶ For a shift by one, the letter 'A' becomes letter 'B', and the letter 'B' becomes the letter 'C', ... , and the letter 'Z' becomes the letter 'A'.
- ▶ An example of the device they used at the time.



Cryptography (cont.)

- ▶ For example, say we want to encrypt the message “BOB” using a shift/rotation of 2.
 1. B \rightarrow D
 2. O \rightarrow Q
 3. B \rightarrow D
- ▶ Hence, we get the *encrypted* text “DQD”. The receiver of the message just needs do the process backwards.

¹Try see if you can figure out what these attacks are! If you find it fun, look up “Cryptograms”!

Cryptography (cont.)

- ▶ For example, say we want to encrypt the message “BOB” using a shift/rotation of 2.
 1. B \rightarrow D
 2. O \rightarrow Q
 3. B \rightarrow D
- ▶ Hence, we get the *encrypted* text “DQD”. The receiver of the message just needs do the process backwards.
- ▶ (The message BOB is usually called the *plaintext*, and the resulting encrypted message is called the *ciphertext*).
- ▶ There are many attacks on this cipher and this has long since been considered insecure.¹

¹Try see if you can figure out what these attacks are! If you find it fun, look up “Cryptograms”!

Cryptography (cont.)

- ▶ Let's skip ahead a bit. There were some advancements, but let's jump to World War 2.

Cryptography (cont.)

- ▶ Let's skip ahead a bit. There were some advancements, but let's jump to World War 2.
- ▶ There is a saying that war promotes technological advancement. During the war, nations used more advanced ciphers to encrypt their messages, as to ensure that if the messages were intercepted by their enemies, they could not glean a tactical advantage.
- ▶ Cryptanalysts studied these machines for weaknesses and managed to crack many of these ciphertexts.

Cryptography (cont.)

- ▶ Let's skip ahead a bit. There were some advancements, but let's jump to World War 2.
- ▶ There is a saying that war promotes technological advancement. During the war, nations used more advanced ciphers to encrypt their messages, as to ensure that if the messages were intercepted by their enemies, they could not glean a tactical advantage.
- ▶ Cryptanalysts studied these machines for weaknesses and managed to crack many of these ciphertexts.



Public Key Cryptography

- ▶ Thus far, the same key has been used for encryption and decryption.
- ▶ In the 1970s, researchers and cryptographers at GCHQ (The UK's government intelligence agency) invented this.
- ▶ It allows for a separate key to be used for encryption and decryption.

- ▶ Cryptography has lots of maths involved. Elliptic curves, integer factorisation, etc.
- ▶ Take for example, the Fundamental Theorem of Arithmetic, that states that every integer greater than 1 either is a prime number itself or can be represented as the product of prime numbers.
- ▶ Say you take two large prime numbers p_1 and p_2 and multiply them to get c . Because of the Fundamental Theorem of Arithmetic, we know there is only one prime factorisation of c , which is through p_1 and p_2 . There isn't a known efficient algorithm to quickly find these factors p_1 and p_2 .²

²For classical computers at least, for quantum computers there is Shor's algorithm.

Reverse Engineering

Reverse Engineering

Just to scare you a little bit, here's me reverse engineering a simple sample program I wrote.

Reverse Engineering

Just to scare you a little bit, here's me reverse engineering a simple sample program I wrote.

The screenshot shows a remote desktop environment with a Ghidra interface. The main window displays assembly code for two sections: '0x Disassembly (plt)' and 'Stack (pop 2540:SP)'. The assembly code includes instructions such as 'lea rax, []', 'mov rdx, []', and 'call __imp_gets'. The registers window shows values for rax, rdx, rcx, etc. In the top right corner, there are two small video feeds: one of a man wearing glasses and a headset, and another showing a close-up of a hand holding a small object, possibly a tool or a component.

What is Reverse Engineering?

reverse engineer verb



reverse engineered; reverse engineering; reverse engineers

Definition of *reverse engineer*

transitive verb

: to disassemble and examine or analyze in detail (a product or device) to discover the concepts involved in manufacture usually in order to produce something similar

Reverse Engineering

- ▶ Primarily done to work around proprietary software.
- ▶ Security Analysts reverse engineer new malware and figure out how they work, and what new flaws they exploit.
- ▶ For historical reasons, old hardware can be reverse-engineered so that future generations can see how the old software was used via emulation.
- ▶ Perhaps you just want to figure out how a particular software does something.
- ▶ You could reverse engineer a program to find a vulnerability to hack and gain access to the system running the program.

Reverse Engineering (cont.)

- ▶ There are techniques to make reverse-engineering harder³. These techniques are employed by malware authors and companies that wish to try and hide their techniques in their software (e.g: Enforce DRM so that software only works on a fixed number of devices, or hide their algorithms from competitors).

³Look into Obfuscation or anti-debug techniques for example. 

Reverse Engineering (cont.)

- ▶ There are techniques to make reverse-engineering harder³. These techniques are employed by malware authors and companies that wish to try and hide their techniques in their software (e.g: Enforce DRM so that software only works on a fixed number of devices, or hide their algorithms from competitors).
- ▶ But generally, reverse engineering is almost always possible.

³Look into Obfuscation or anti-debug techniques for example. 

Example: Pegasus Malware

- ▶ Amnesty International Security Lab released a report based on their forensic analysis and reverse engineering of the Pegasus malware, which they found mostly on journalists and activists. They found a 0-day vulnerability⁴ in iMessage being actively exploited in the wild.



WHO WE ARE

WHAT WE DO

COUNTRIES

6



Forensic Methodology Report: How to catch NSO Group's Pegasus

A copy of this report is available for download [here](#).

Introduction

NSO Group claims that its Pegasus spyware is only used to “investigate terrorism and crime” and “leaves no traces whatsoever”. This Forensic Methodology Report shows that neither of these statements are true. This report accompanies the release of the Pegasus Project, a collaborative investigation that involves more than 80 journalists from 17 media organizations in 10 countries coordinated by Forbidden Stories with technical support of Amnesty International’s Security Lab.^[1]

Amnesty International’s Security Lab has performed in-depth forensic analysis of numerous mobile devices from human rights defenders (HRDs) and journalists around the world. This research has uncovered widespread, persistent and ongoing unlawful surveillance and human rights abuses perpetrated using NSO Group’s Pegasus spyware.

As laid out in the UN Guiding Principles on Business and Human Rights, NSO Group should urgently take proactive steps to ensure that it does not cause or contribute to human rights abuses within its global operations, and to respond to any human rights abuses when they do occur. In order to meet that responsibility, NSO Group must carry out adequate human rights due diligence and take steps to ensure that HRDs and journalists do not continue to become targets of unlawful surveillance.

In this Forensic Methodology Report, Amnesty International is sharing its methodology and publishing an open-source mobile forensics tool and detailed technical indicators, in order to assist information security researchers and civil society with detecting and responding to these serious threats.

Example: GTA5

- ▶ Some guy found the long loading times for GTA5 quite odd, so he started reverse-engineering it and found a few bugs that made the loading way, way slower than it should have been.
- ▶ <https://nee.lv/2021/02/28/How-I-cut-GTA-Online-loading-times-by-70/>
- ▶ He was awarded \$10k USD by Rockstar for finding this.
- ▶ He looked at the disassembled code (after bypassing the obfuscation Rockstar put in place) and found the problem was with problems with the JSON file parsing, which has a lot of redundant checks.

Example: Old software

- ▶ Hobbyists have reverse-engineered old hardware that is no longer supported. (defunct company, they moved on to more powerful computers, etc.)
- ▶ This allows for old software to be archived for historical purposes via emulation, as hardware eventually deteriorates.

Poking around for fun

- ▶ Sometimes, you can find unused code or assets that didn't make it to the final product.
- ▶ In games, you might find some old concept art or scrapped levels.
- ▶ Sometimes you can find vulnerabilities in them too. Glitched speedruns are a thing for some video games.
- ▶ Someone found a way to do code injection in Super Mario World, and made a Flappy Bird clone inside it using just the SNES controller. In newer devices, code injection can also be used to inject malware into devices.

The screenshot shows the EarthBound Wiki page. At the top, there's a navigation bar with "Page" and "Discussion" tabs. Below that is a search bar and a message: "If you appreciate the work done within the wiki, please consider supporting The Cutting Room Floor on Patreon. Thanks for all your support!".

The main content area is titled "EarthBound" and includes a description: "EarthBound is Shigesato's charming, cut-NO-RPG about aliens, teenagers, and psychic powers." Below this is a "To do" list:

- Various unused text, documented in Matt's logs/old localization book, some is exclusive to the Japanese version.
- Unused map templates?

A "Contents" box lists various sections:

1. Sub-Pages
2. Unused Graphics
3. Unused Enemy
4. Unused Items
 - 4.1. Unused Relic
 - 4.2. Temporary Goods
5. Unused Music
 - 5.1. Unused Unused
 - 5.2. Unused Unused
 - 5.3. Unused Unused
6. Copy Protection
 - 6.1. Layer One - Region Protection
 - 6.2. Layer Two - World Check
 - 6.3. Layer Three - Increased Encounters
 - 6.4. Layer Four - Unknown
 - 6.5. Layer Five - The Grand Finale
 - 6.6. Trivia
7. Unused Events
8. Miscellaneous Differences
 - 8.1. Japanese-Only Changes
 - 8.2. FEZ Adaptations
 - 8.3. Customer Changes

On the right side, there's a "Media" section with a thumbnail of the EarthBound box art. Below it, a table lists game details:

EarthBound
Also known as: <i>64</i> (EU), <i>2</i> (USA), <i>The Game no Quadratura da Sfera</i> (JP)
Developers: <i>Apex</i> , <i>HAL Laboratory</i> , <i>Flag Corporation</i>
Publisher: <i>Nintendo</i>
Platforms: <i>SNES</i>
Released in JP: <i>August 27, 1994</i>
Released in EU: <i>June 5, 1995</i>
Released in US: <i>July 12, 2010 (DS-U Virtual Console)</i>

Below the table, there are several tags indicating features like "This game has unused enemies", "This game has unused graphics", "This game has unused items", "This game has unused music", "This game has obsolescent material", "This game has a hidden sound track", "This game has regional differences", "This game has regional differences", "This game has unused references", and "This game has unused references".

So what is a CTF?

- ▶ Stands for “Capture the Flag”.
- ▶ It’s a competition/game.
- ▶ The “Flag” is a secret word that you need to find.
- ▶ Recovering a “Flag” will give you points.
- ▶ There is usually a prize for the team or individual with the most points.

So what is a CTF?

- ▶ Stands for “Capture the Flag” .
- ▶ It’s a competition/game.
- ▶ The “Flag” is a secret word that you need to find.
- ▶ Recovering a “Flag” will give you points.
- ▶ There is usually a prize for the team or individual with the most points.
- ▶ Usually this secret flag is hidden somehow, e.g: It’s encrypted and you need to figure out a way to decrypt it, or it’s only accessible if you can reverse engineer the program.

So what is a CTF?

- ▶ Stands for “Capture the Flag”.
- ▶ It’s a competition/game.
- ▶ The “Flag” is a secret word that you need to find.
- ▶ Recovering a “Flag” will give you points.
- ▶ There is usually a prize for the team or individual with the most points.
- ▶ Usually this secret flag is hidden somehow, e.g: It’s encrypted and you need to figure out a way to decrypt it, or it’s only accessible if you can reverse engineer the program.
- ▶ Besides being “fun”, it is always a way to harness your skills, because it usually requires you have a deeper understanding of computers (e.g: You need to understand lower-level languages like assembly, or understand the maths behind some badly-implemented encryption.)

New ▾ Open Save Save As Undo ▾ Redo ▾

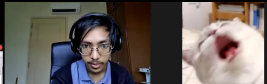
a.out ×

0000:0000	7F 45 4C 46	02 01 01 00	00 00 00 00	00 00 00 00	.ELF.....
0000:0010	03 00 3E 00	01 00 00 00	30 11 00 00	00 00 00 00	..>.....0.....
0000:0020	40 00 00 00	00 00 00 00	C0 3A 00 00	00 00 00 00	@.....À:.....
0000:0030	00 00 00 00	40 00 38 00	0B 00 40 00	1E 00 1D 00@.8...@.....
0000:0040	06 00 00 00	04 00 00 00	40 00 00 00	00 00 00 00@.....
0000:0050	40 00 00 00	00 00 00 00	40 00 00 00	00 00 00 00	@.....@.....
0000:0060	68 02 00 00	00 00 00 00	68 02 00 00	00 00 00 00	h.....h.....
0000:0070	08 00 00 00	00 00 00 00	03 00 00 00	04 00 00 00
0000:0080	A8 02 00 00	00 00 00 00	A8 02 00 00	00 00 00 00
0000:0090	A8 02 00 00	00 00 00 00	1C 00 00 00	00 00 00 00
0000:00A0	1C 00 00 00	00 00 00 00	01 00 00 00	00 00 00 00
0000:00B0	01 00 00 00	04 00 00 00	00 00 00 00	00 00 00 00
0000:00C0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000:00D0	48 06 00 00	00 00 00 00	48 06 00 00	00 00 00 00	H.....H.....
0000:00E0	00 10 00 00	00 00 00 00	01 00 00 00	05 00 00 00
0000:00F0	00 10 00 00	00 00 00 00	00 10 00 00	00 00 00 00
0000:0100	00 10 00 00	00 00 00 00	8D 02 00 00	00 00 00 00
0000:0110	8D 02 00 00	00 00 00 00	00 10 00 00	00 00 00 00
0000:0120	01 00 00 00	04 00 00 00	00 20 00 00	00 00 00 00
0000:0130	00 20 00 00	00 00 00 00	00 20 00 00	00 00 00 00

```

1  class Tutor {
2      private String name;
3      private int age;
4      private String gender;
5      private String address;
6
7      // 获取 Tutor 信息
8      public String getTutorInfo() {
9          return "Tutor: " + name + ", Age: " + age + ", Gender: " + gender + ", Address: " + address;
10     }
11
12     // 获取 Tutor 姓名
13     public String getTutorName() {
14         return name;
15     }
16
17     // 获取 Tutor 年龄
18     public int getTutorAge() {
19         return age;
20     }
21
22     // 获取 Tutor 性别
23     public String getTutorGender() {
24         return gender;
25     }
26
27     // 获取 Tutor 地址
28     public String getTutorAddress() {
29         return address;
30     }
31
32     // 设置 Tutor 信息
33     public void setTutorInfo(String name, int age, String gender, String address) {
34         this.name = name;
35         this.age = age;
36         this.gender = gender;
37         this.address = address;
38     }
39
40     // 设置 Tutor 姓名
41     public void setTutorName(String name) {
42         this.name = name;
43     }
44
45     // 设置 Tutor 年龄
46     public void setTutorAge(int age) {
47         this.age = age;
48     }
49
50     // 设置 Tutor 性别
51     public void setTutorGender(String gender) {
52         this.gender = gender;
53     }
54
55     // 设置 Tutor 地址
56     public void setTutorAddress(String address) {
57         this.address = address;
58     }
59 }

```



CTF Competition Examples

1. Google CTF
2. DefCon CTF
3. F-Secure Cybersecurity challenge (they have a local branch)
4. ...Many more

Real life

- ▶ I kinda lied to you...

Real life

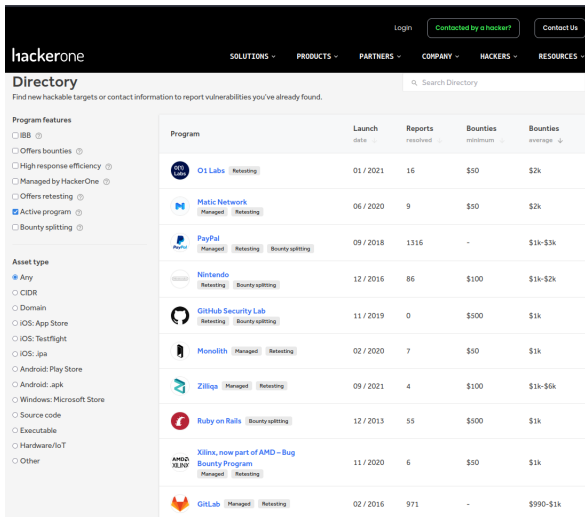
- ▶ I kinda lied to you...
- ▶ In real life, it's probably more to do with writing reports, auditing security practices for companies, etc.
- ▶ There are some rarer exceptions. Google has their “Project Zero” team tasked to find 0-day vulnerabilities (vulnerabilities that hasn't been patched yet).

Real life











- ▶ I kinda lied to you...
- ▶ In real life, it's probably more to do with writing reports, auditing security practices for companies, etc.
- ▶ There are some rarer exceptions. Google has their “Project Zero” team tasked to find 0-day vulnerabilities (vulnerabilities that hasn't been patched yet).
- ▶ There are also bug bounty programs offered by some companies though.

Bug bounties

- ▶ Hackerone is one of the sites that acts as a middle-man for this.



The screenshot shows the Hackerone Directory interface. At the top, there's a navigation bar with 'SOLUTIONS', 'PRODUCTS', 'PARTNERS', 'COMPANY', 'HACKERS', and 'RESOURCES'. The main header includes the 'hackerone' logo, a search bar, and buttons for 'Login', 'Contacted by a hacker?', and 'Contact Us'. Below the header, the 'Directory' section is titled 'Find new hackable targets or contact information to report vulnerabilities you've already found.' On the left, there are filters for 'Program features' and 'Asset type'. The main content is a table of programs.

Program	Launch date	Reports resolved	Bounties minimum	Bounties average
 01 Labs Retesting	01 / 2021	16	\$50	\$2k
 Matic Network Managed Retesting	06 / 2020	9	\$50	\$2k
 PayPal Managed Retesting Bounty splitting	09 / 2018	1316	-	\$1k-\$3k
 Nintendo Retesting Bounty splitting	12 / 2016	86	\$100	\$1k-\$2k
 GitHub Security Lab Retesting Bounty splitting	11 / 2019	0	\$500	\$1k
 Monolith Managed Retesting	02 / 2020	7	\$50	\$1k
 Zilliqa Managed Retesting	09 / 2021	4	\$100	\$1k-\$6k
 Ruby on Rails Bounty splitting	12 / 2013	55	\$500	\$1k
 AMD Xilinx Xilinx, now part of AMD - Bug Bounty Program Managed Retesting	11 / 2020	6	\$50	\$1k
 GitLab Managed Retesting	02 / 2016	971	-	\$990-\$1k

- ▶ You get paid based on how severe it is. If you have a Proof Of Concept that it's possible to completely take over their system, you get paid more.
- ▶ Other companies like Google, Facebook, Mozilla, etc. also have their own bug bounty programs.
- ▶ You communicate with their security team explaining in-depth what the bug is and the impact, giving Proof-Of-Concepts (POCs) if necessary.
- ▶ You agree not to give their team a period of time, such as 90 days, to fix their problem before disclosing it to the public.

Other stuff...

- ▶ Cybersecurity is a wide field, and I couldn't cover everything.
- ▶ Some other points of interest you may like.
 1. OSINT (Open Source Intelligence)
 2. Hardware Hacking (For you electronic nerds out there :D)
 3. Web Hacking
 4. Stegonography (hiding information in plain sight)
 5. Attacking Network Protocols
 6. More in-depth topics of things we discussed.

Fin.

- ▶ Thanks for listening!
- ▶ Let me know what you find interesting!
- ▶ Let me know how much you understood!