Axel Munoz: CSDN, BAH DarkLabs

Practical Applications of Reverse Engineering using NSA's GHIDRA

README

- $whoami
- Cybersecurity Landscape
- Reverse Engineering
- Offensive RE
- Defensive RE
- GHIDRA
- Reverse Engineering Walkthrough

$whoami

Custom-Designed Major:

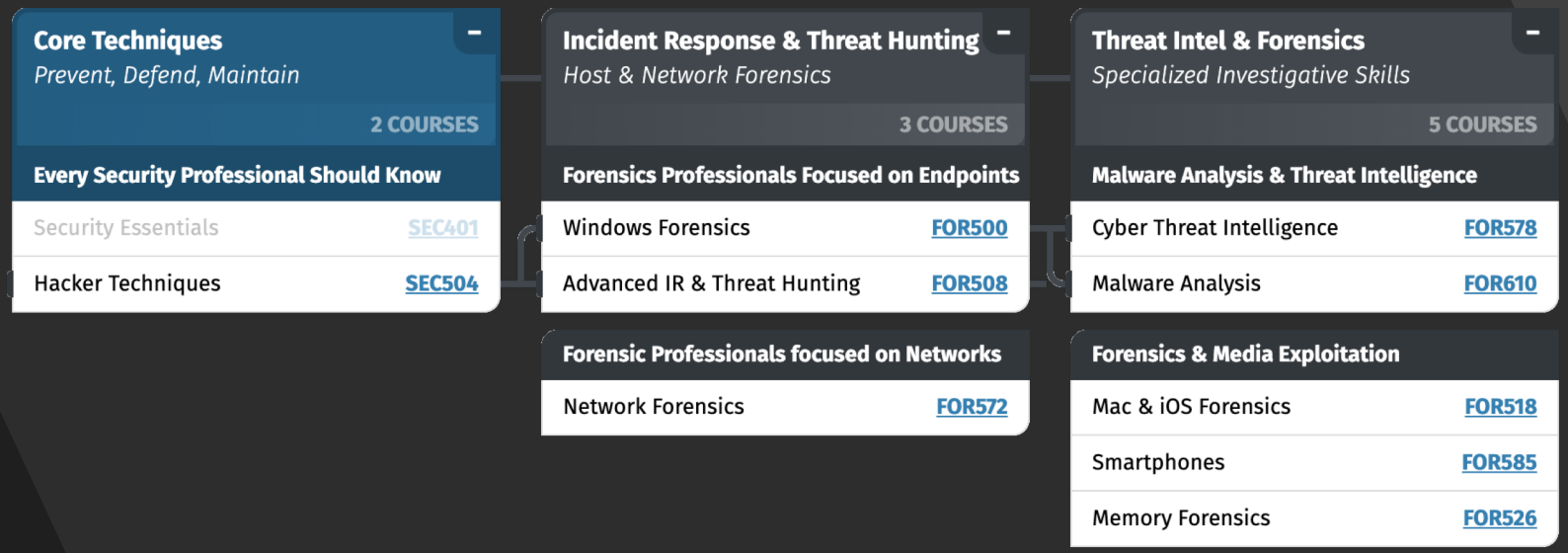
- Cryptology and Intelligence Analysis
- Enables students to pursue individualized course of studies not available through existing majors or combination of majors and minors.
- Designed for highly motivated students who have interdisciplinary curiosity and career ambitions.
- The Major:
  - Requires an intensive, but guided, application process
  - Emphasizes self-directed learning
  - Incorporates early and intensive research
  - Culminates in a faculty mentored capstone project

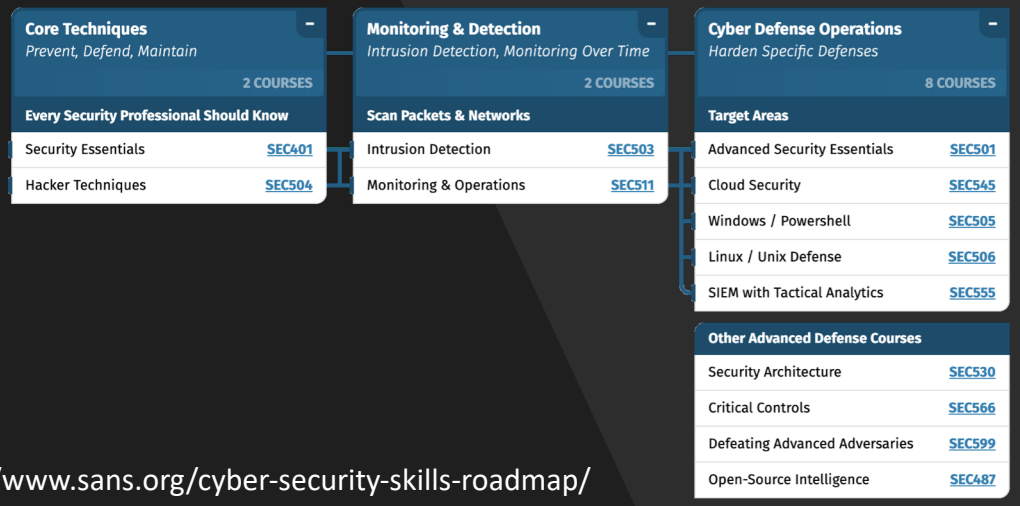
Booz Allen Hamilton:

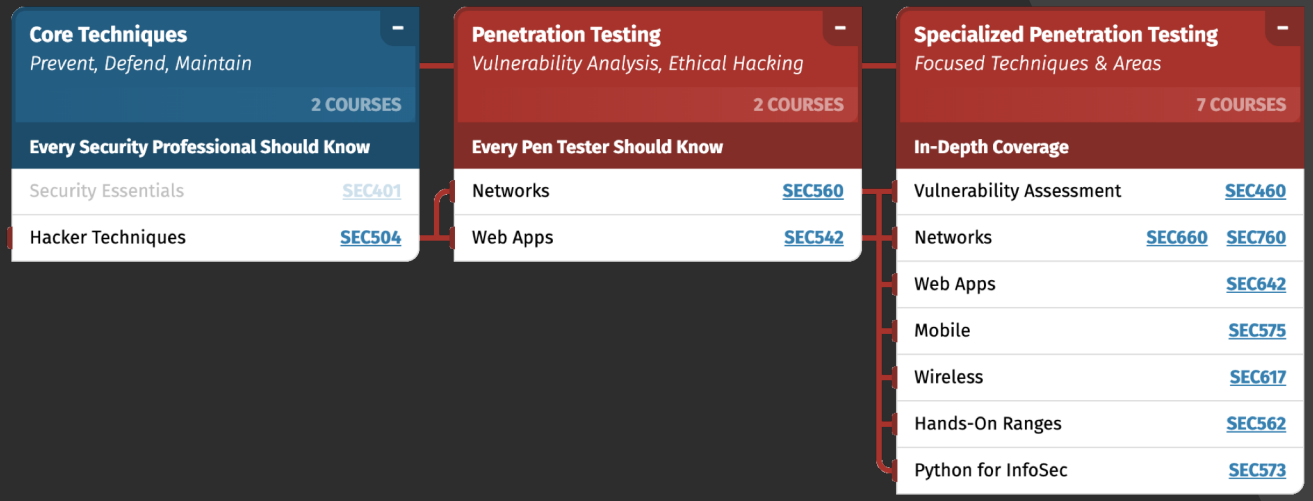
- Senior Threat Hunter
- DarkLabs
  - Client Hunting
  - Research & Development
- Developing novel Hunt methods

https://drexel.edu/pennoni/custom-designed-major/about/

Cybersecurity Landscape







https://www.sans.org/cyber-security-skills-roadmap/

Reverse Engineering

- Machine Language are the `1`'s and `0`'s your computer understands
- Even the most trivial code gets compiled into complex, hard-to-understand assembly in order to run
- Malware and security software that doesn't want to be analyzed can fight back
  - Assembly can change during the analysis process
  - Assembly could be "packed" or encrypted
- Malware doesn't want to be analyzed so you can't detect it in the future
- Security software doesn't want to be analyzed so attackers can't find vulnerabilities

Reverse Engineering Hello, Nerd Night! -> Machine Code

- Python

```
print("Hello, Nerd Night!”)
```

- C

```
#include <stdio.h>
#include <stdlib.h>
 int main(void) {
    puts("Hello World!");
    return EXIT_SUCCESS;
}
```

- Javascript

```
console.log("Hello, Nerd Night!");
```

- Fortran

```
PROGRAM HELLO
WRITE (*,100)
STOP
100 FORMAT ('Hello, Nerd Night!' /)
END
```

```
➜ RE Examples cat hello_nerd_night.c
#include <stdio.h>
#include <stdlib.h>

int main(void) {
  puts("Hello, Nerd Night!");
  return EXIT_SUCCESS;
}
➜ RE Examples gcc -Wall hello_nerd_night.c -o hello_nerd_night.o
➜ RE Examples ./hello_nerd_night.o
Hello, Nerd Night!
➜ RE Examples r2 hello_nerd_night.o
r_config_set: variable 'asm.cmtright' not found
 -- Analyze socket connections with the socket plugin: 'radare2 socket://www.foo.com:80'. Use 'w' to send data
[0x100000f50]> aa
[x] Analyze all flags starting with sym. and entry0 (aa)
[0x100000f50]> pdf@main
            ;-- main:
            ;-- section.0.__TEXT.__text:
            ;-- func.100000f50:
            ;-- rip:
┌ (fcn) entry0 40
   entry0 (int arg4);
            ; var int local_8h @ rbp-0x8
            ; var int local_4h @ rbp-0x4
            ; arg int arg4 @ rcx
            0x100000f50      55             push rbp                    ; [00] -r-x section size 40 named 0.__TEXT.__text
            0x100000f51      4889e5         rbp = rsp
            0x100000f54      4883ec10       rsp -= 0x10
            0x100000f58      c745fc000000.  dword [local_4h] = 0
            0x100000f5f      488d3d340000.  rdi = str.Hello__Nerd_Night ; section.3.__TEXT.__cstring ; 0x100000f9a ; "Hello, Nerd Night!"
            0x100000f66      e80d000000     sym.imp.puts ()             ; int puts(const char *s)
            0x100000f6b      31c9           ecx = 0
            0x100000f6d      8945f8         dword [local_8h] = eax
            0x100000f70      89c8           eax = ecx                   ; arg4
            0x100000f72      4883c410       rsp += 0x10
            0x100000f76      5d
            0x100000f77      c3             return
[0x100000f50]>
```

Offensive RE

- Just because it’s offensive, doesn’t mean it’s illegal or even unethical
- Company pay analysts to RE software to discover vulnerabilities
- Analysts RE software looking for bugs for bug bounty ($$!)
- Requires background in exploitation development
  - How can I abuse this bug in order to take control?

Defensive RE

- Very different than Offensive RE
- Look for unique signatures for future detection
- 5 W's+H:
  - **How** did this malware arrive in our environment?
  - **Who** sent this malware?
  - **What** is the objective for the malware? (Exfiltration, Botnet, C2 connection, System damage)
  - **When** did the malware arrive? **When** was it made?
  - **Where** did this malware come from? (Infrastructure, country, etc.)
  - **Why** was this malware targeted towards us? (Financial gain, Intellectual Property, Hacktivism)

GHIDRA

- National Security Agency created a disassembler and decompiler
  - Makes it easier to read and understand the code
  - Still impossible to retrieve original code, so still difficult
- Completely free and Open Source
  - Anyone can look at the source code for the program and make improvements
- The only (decent) free disassembler for Mac, Linux, and Windows
  - Supports 25+ instruction sets (32bit, 64bit, ARM, CHIP-8, etc.)

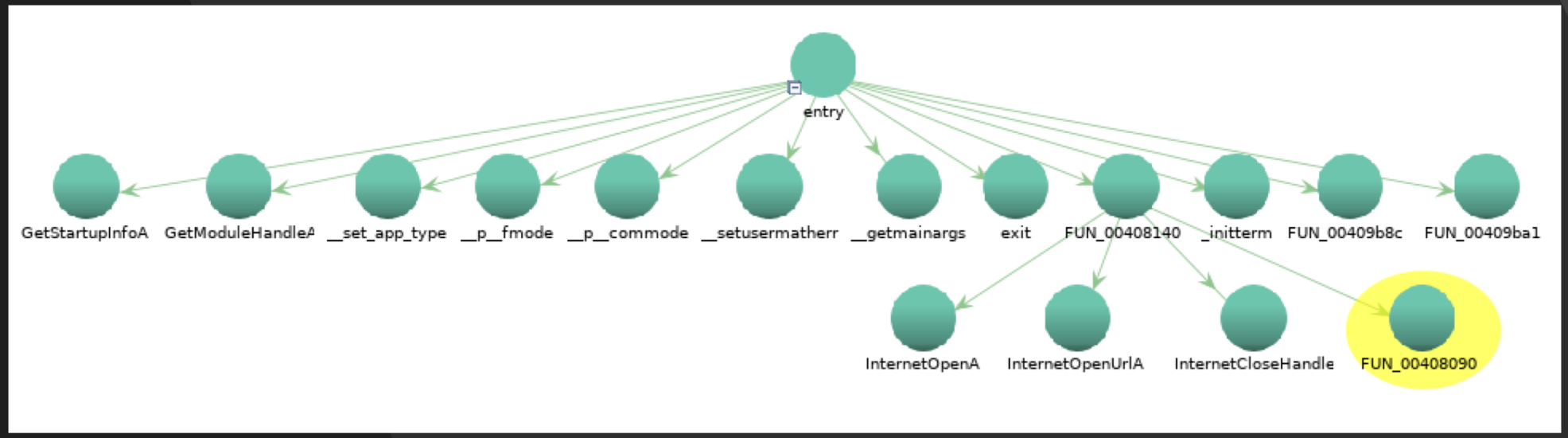
Reverse Engineering Walkthrough w/ GHIDRA

- WannaCry
  - Ransomware attack ~ May 2017
  - 200k computers across 150 countries
  - Billions of dollars in damage
  - Attributed to DPRK actors
  - Used EternalBlue Exploit developed by NSA (ironic)
- Infamous kill switch
  - Hardcoded domain to check if it should proceed
  - Smart move, bad execution
  - Marcus Hutchins first discovered it
  - Let's find it!

https://www.ghidra.ninja/posts/03-wannacry-1/

Walkthrough (cont.)

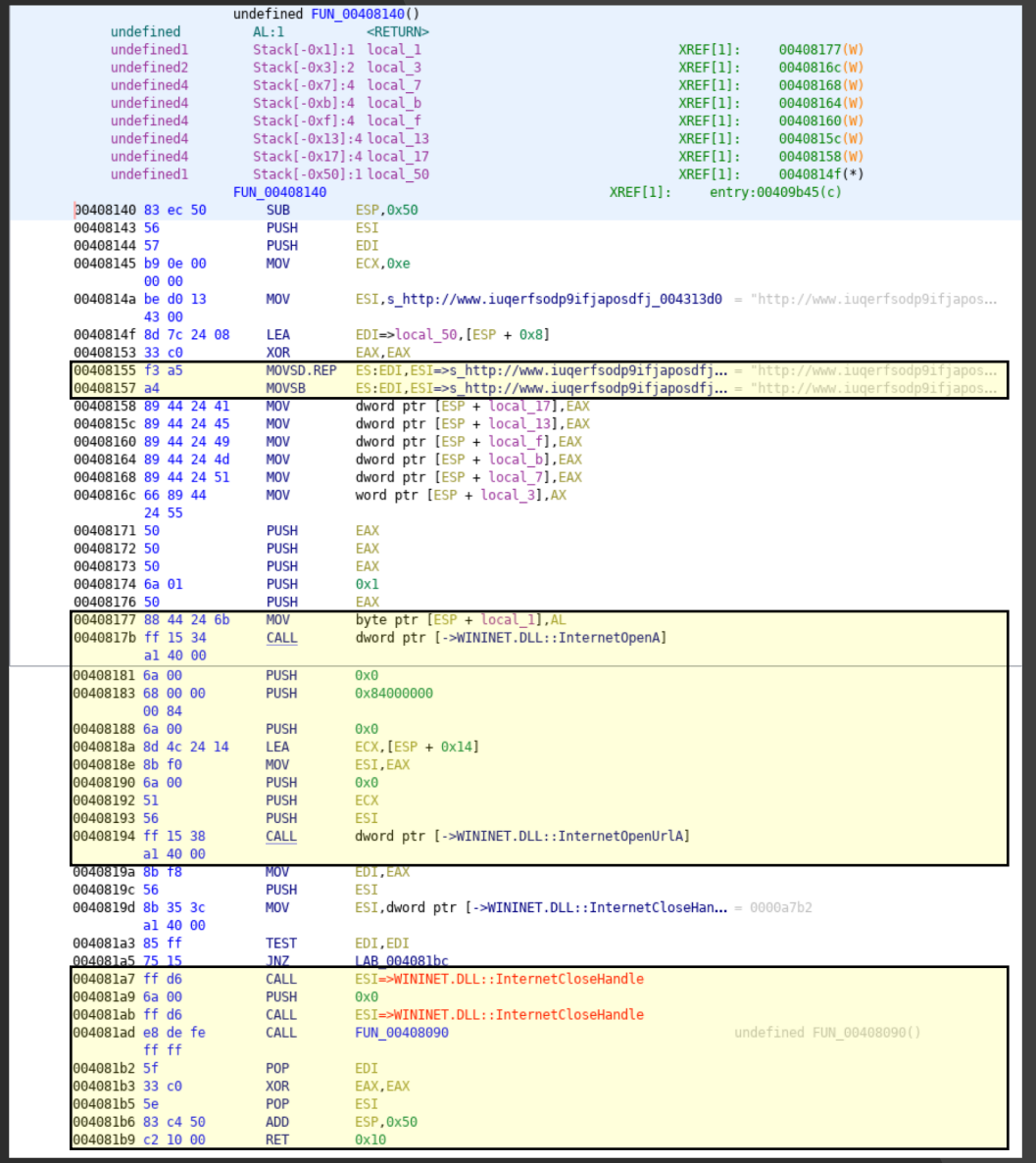
- Function Call Graphs (a feature of GHIDRA) shows the parent-child relationship of function calls
- Highlighted function starts the actual malicious code (encryption of files)
- **How** does the code decide whether or not to call this function?



Walkthrough (cont.)

- Interesting that there are functions referring to the internet so early in the code execution… Let's look into that.
- Section One shows a URL
- Section Two shows the program connecting to that URL
- Section Three shows that the program will quit without malicious behavior if the connection fails
- But wait… That isn't a valid URL…
  - `s_http://www.iuqerfsodp9ifjaposdfj_004313d0`

http://www[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com



Walkthrough (cont.)

- GHIDRA has a built-in decompiler which has a very naïve code converter
- It's still difficult to understand
- Unlike other software, GHIDRA makes it easy to decode

Decompile: killswitch_logic - (wannacry)

```
undefined4 killswitch_logic(void)

{
  HINTERNET hInternet;
  HINTERNET hInternet_rtr;
  int i;
  char *strange_url;
  char *strange_url_copy;
  char strange_url_buffer [57];

  i = 14;
  strange_url = s_http://www.iuqerfsodp9ifjaposdfj_004313d0;
  strange_url_copy = strange_url_buffer;
  while (i != 0
                    /* strncpy(strange_url_copy, strange_url, 14); */) {
    i = i + -1;
    *(undefined4 *)strange_url_copy = *(undefined4 *)strange_url;
    strange_url = strange_url + 4;
    strange_url_copy = strange_url_copy + 4;
  }
  *strange_url_copy = *strange_url;
  InternetOpenA((LPCSTR)0x0,1,(LPCSTR)0x0,(LPCSTR)0x0,0);
  hInternet_rtr = InternetOpenUrlA(hInternet,strange_url_buffer,(LPCSTR)0x0,0,0x84000000,0);
  if (hInternet_rtr == (HINTERNET)0x0) {
    InternetCloseHandle(hInternet);
    InternetCloseHandle(0);
    malicious_entry();
    return 0;
  }
  InternetCloseHandle(hInternet);
  InternetCloseHandle(hInternet_rtr);
  return 0;
}
```

Decompile: FUN_00408140 - (wannacry)

```
undefined4 FUN_00408140(void)

{
  undefined4 uVar1;
  int iVar2;
  undefined4 *puVar3;
  undefined4 *puVar4;
  undefined4 uStack100;
  undefined4 uStack96;
  undefined4 uStack92;
  undefined4 local_50 [14];
  undefined4 local_17;
  undefined4 local_13;
  undefined4 local_f;
  undefined4 local_b;
  undefined4 local_7;
  undefined2 local_3;
  undefined local_1;

  iVar2 = 0xe;
  puVar3 = (undefined4 *)s_http://www.iuqerfsodp9ifjaposdfj_004313d0;
  puVar4 = local_50;
  while (iVar2 != 0) {
    iVar2 = iVar2 + -1;
    *puVar4 = *puVar3;
    puVar3 = puVar3 + 1;
    puVar4 = puVar4 + 1;
  }
  *(undefined *)puVar4 = *(undefined *)puVar3;
  local_17 = 0;
  local_13 = 0;
  local_f = 0;
  local_b = 0;
  local_7 = 0;
  local_3 = 0;
  uStack92 = 0;
  uStack96 = 0;
  uStack100 = 0;
  local_1 = 0;
  uVar1 = InternetOpenA(0,1);
  iVar2 = InternetOpenUrlA(uVar1,&uStack100,0,0,0x84000000,0);
  if (iVar2 == 0) {
    InternetCloseHandle(uVar1);
    InternetCloseHandle(0);
    FUN_00408090();
    return 0;
  }
  InternetCloseHandle(uVar1);
  InternetCloseHandle(iVar2);
  return 0;
}
```

Walkthrough (cont.)

- Now we know there is a killswitch:
  - If the software can reach that odd domain it won’t encrypt the files
- Next steps are to:
  - Contact authorities to register domain
  - Identify any hosts in your own network who may have reached out to that domain
    - If there are any, quarantine them

How to Get Started

- Computer Science degree
  - CST/IT/IS
- Tier-One Analyst
- Knowledge of Threat Intelligence
  - Understand the geo-political landscape of who might target your organization/clients
- Exploitation Development
  - How can I abuse this software to gain control of the computer?

DREXEL UNIVERSITY
Pennoni
Honors College

Questions?