Introduction to Reverse Engineering

Ali Ghaffarian

Me?

- Interested in
  - Operating Systems
  - Networking
- CTF
  - Active Member of FlagMotori
  - Network Forensics

FlagMotori

Also known as

- parrot fan club
- fl4gmotori

Website: https://fmC.TF

Twitter: https://twitter.com/FlagMotori

Sign in to join the team.

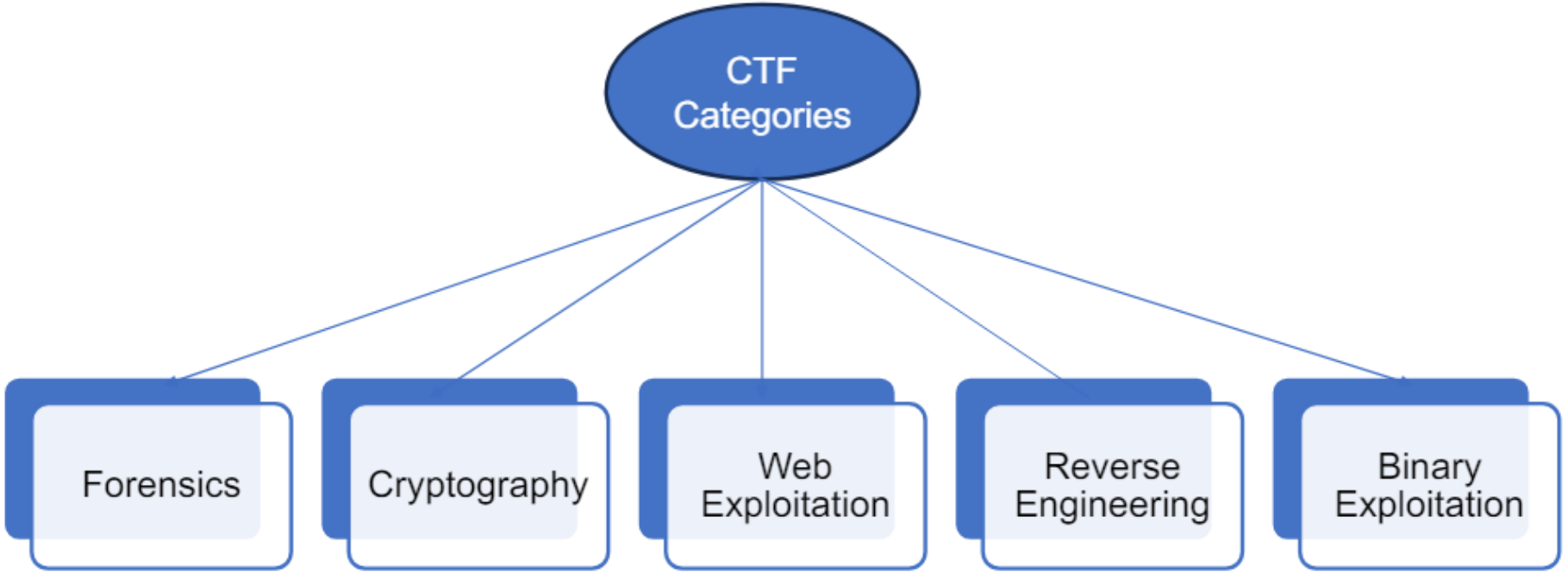
An independent Iranian CTF team composed of players enthusiastic about learning everything related to cybersecurity.

If you are a Persian speaker interested in playing or learning about CTFs, we invite you to join us. Simply fill out this form: https://forms.gle/TsembkjqfQycTm1y8 or send a direct message to mheidari98 on Discord.

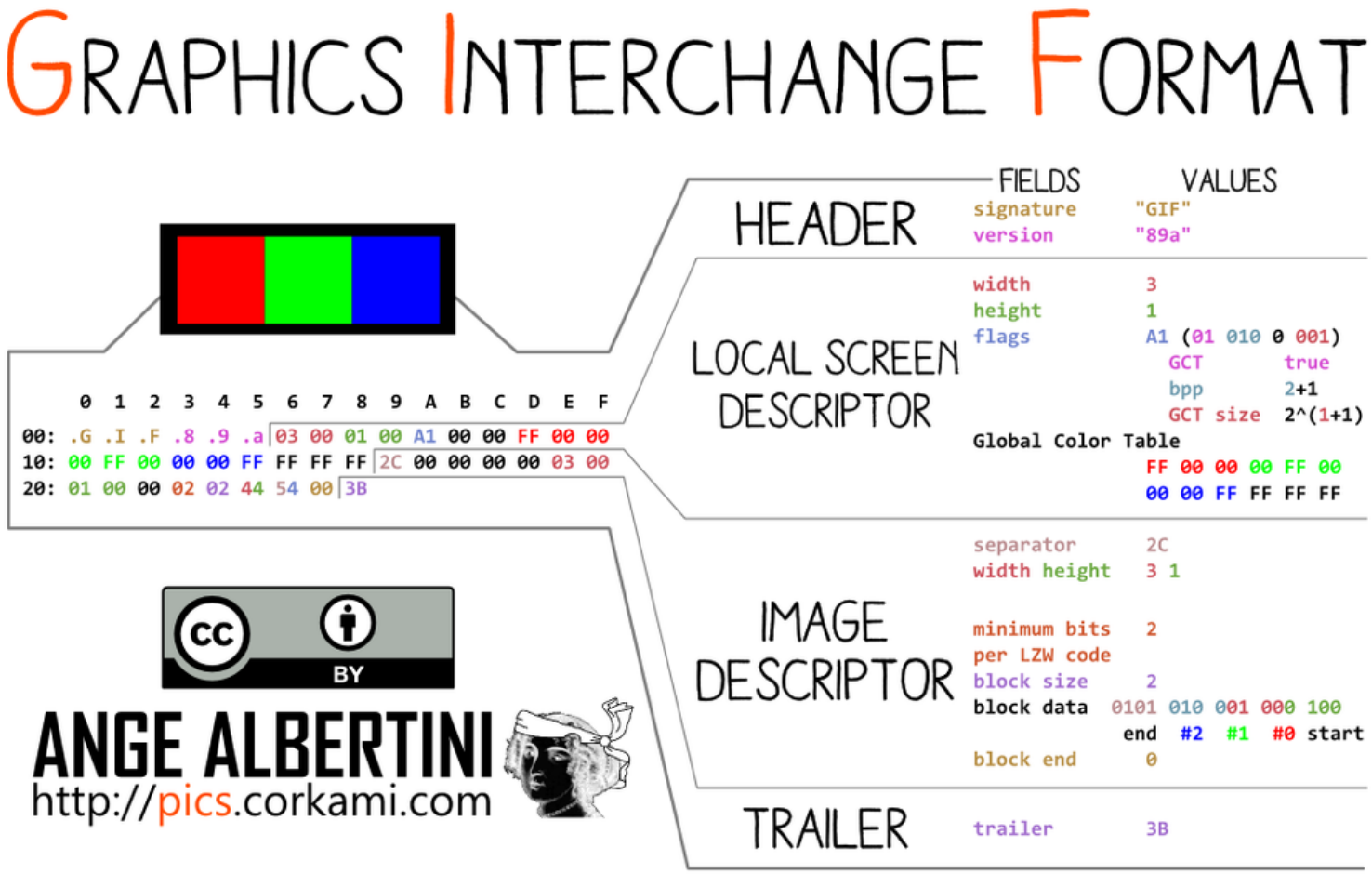
Table of Contents


ctf_
CAPTURE THE FLAG

Categories


CTF Categories
Forensics
Cryptography
Web Exploitation
Reverse Engineering
Binary Exploitation

Forensics



THE GIF WAS CREATED BY COMPUSERVE IN 1987.
IT'S PALETTE BASED: EACH BLOCK IS LIMITED TO 256 COLORS.
IT USES THE LEMPEL-ZIV-WELCH ALGORITHM, WHICH WAS PATENTED UNTIL 2004.

Cryptography

X-net agents have uncovered an encrypted message from the Department of National Security high command. It appears to be using a new cypher called LOLCrypt.

We have also found the tool that DNS agents use to generate LOLCrypt encrypted texts, maybe you can use this in your cracking efforts.

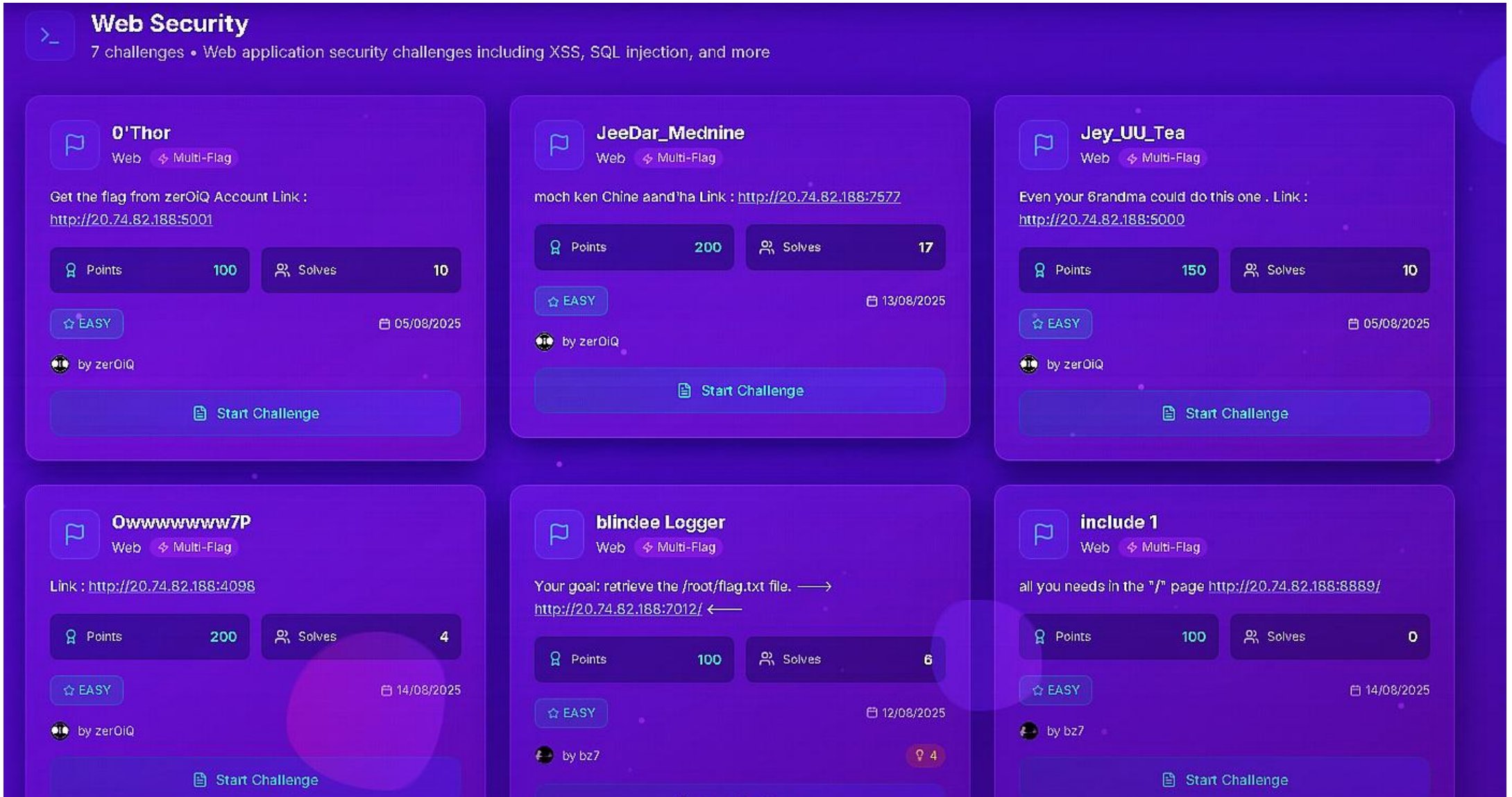
LOLCrypt CipherText:

74 57 14 26 54 54 37 6 45 47 50 8 60 21 49 10 68 64 74 17 23 61 17 35 26 23 61 44 9 118 20 7 17 58 3 70 45 50
45 76 43 34 18 45 8 73 66 76 16 26 62 51 44 47 57 6 50 34 34 38 1 63 23 54 19 52 3 60 51 60 20 12 19 13 12 95
42 43 36 29 24 38 39 43 38 7 24 76 187 17 5 14 15 52 57 19 56 11 43 48 37 42 29 32 37 4 33 66 63 38 23 19 49 36
35 31 0 3 61 77 46 19 39 54 68 50 6 16 40 39 39 36 52 43 96 32 41 37 18 54 15 34 43 48 62 52 17 66 109 11 79 24
28 21 103 35 32 37 48 27 30 51 29 35 17 86 76 37 12 35 55 37 24 21 82 96 75 59 18 37 54 26 70 20 50 55 39 35 33
53 3 90 50 54 14 55 21 68 59 22 11 73 76 17 31 57 63 72 60 48 23 15 60 12 26 56 27 79 86 31 12 21 38 86 26 25
23 67 58 2 60 24 50 46 16 110

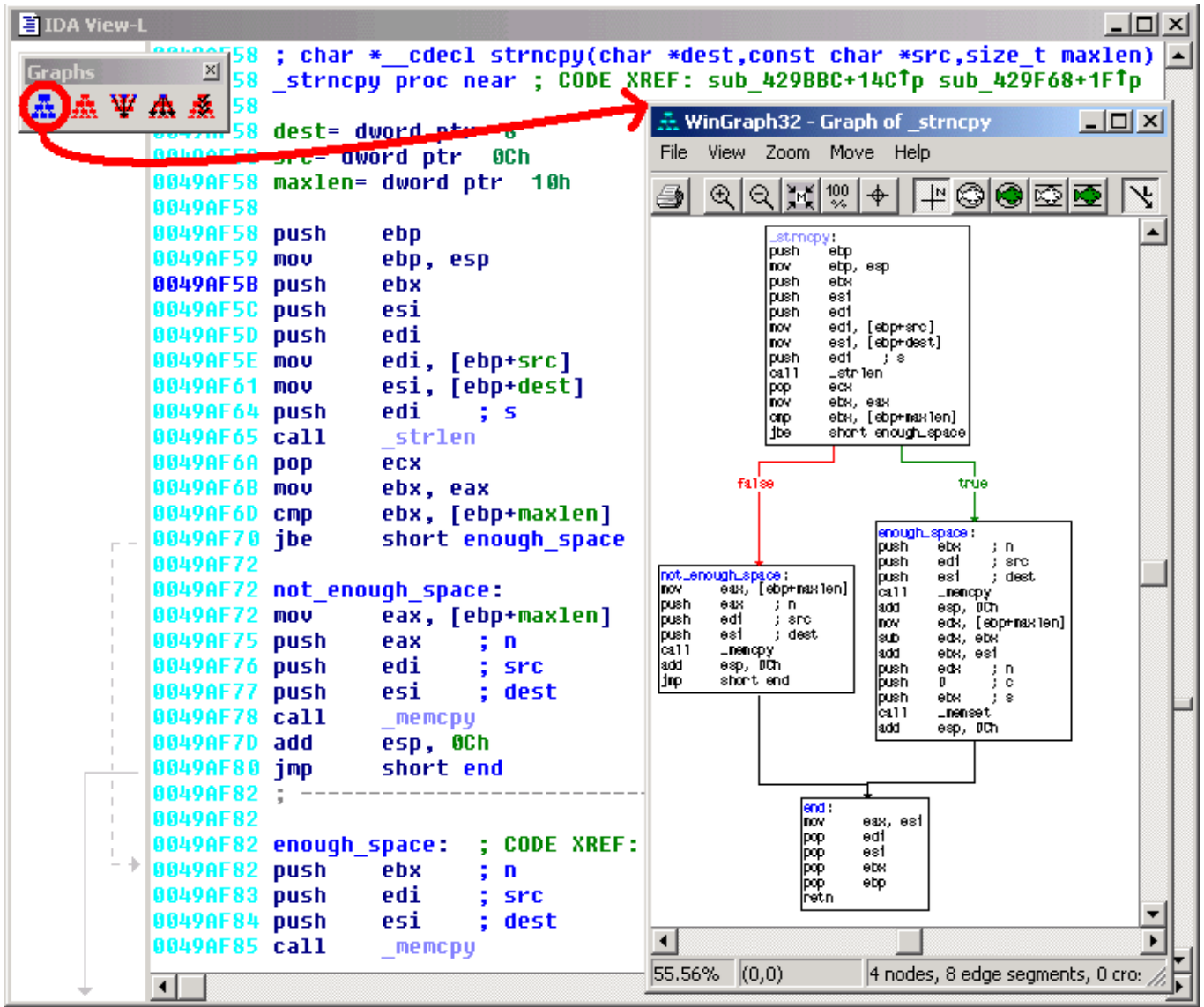
Plain text: [redacted] encrypt

Encrypted output:

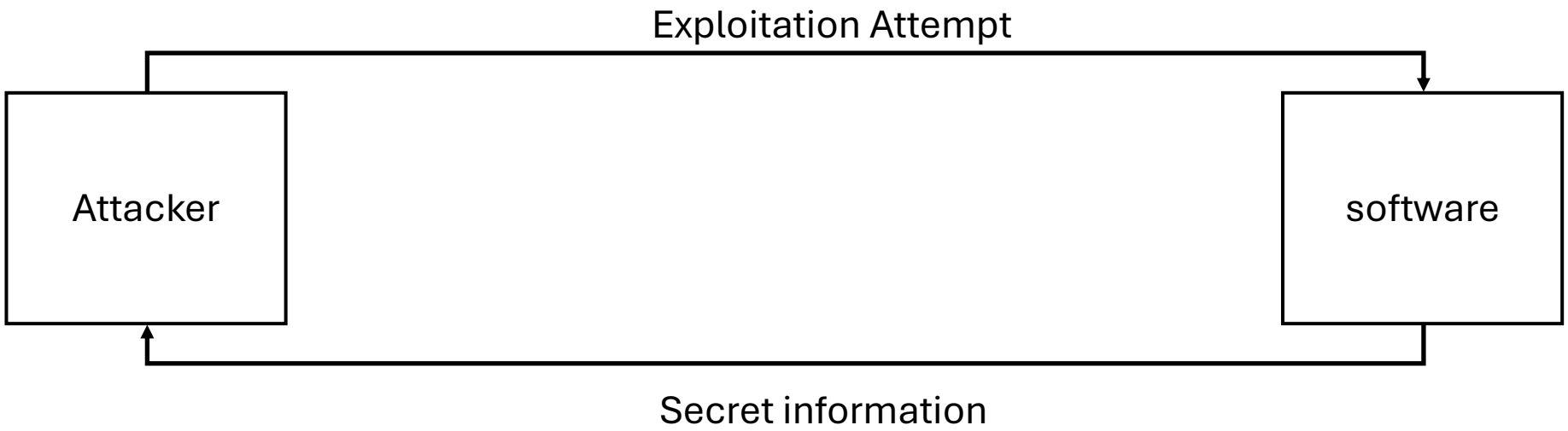
Web Security



Reverse Engineering



Binary Exploitation



Introduction to Binary Analysis

Make it do what you want!

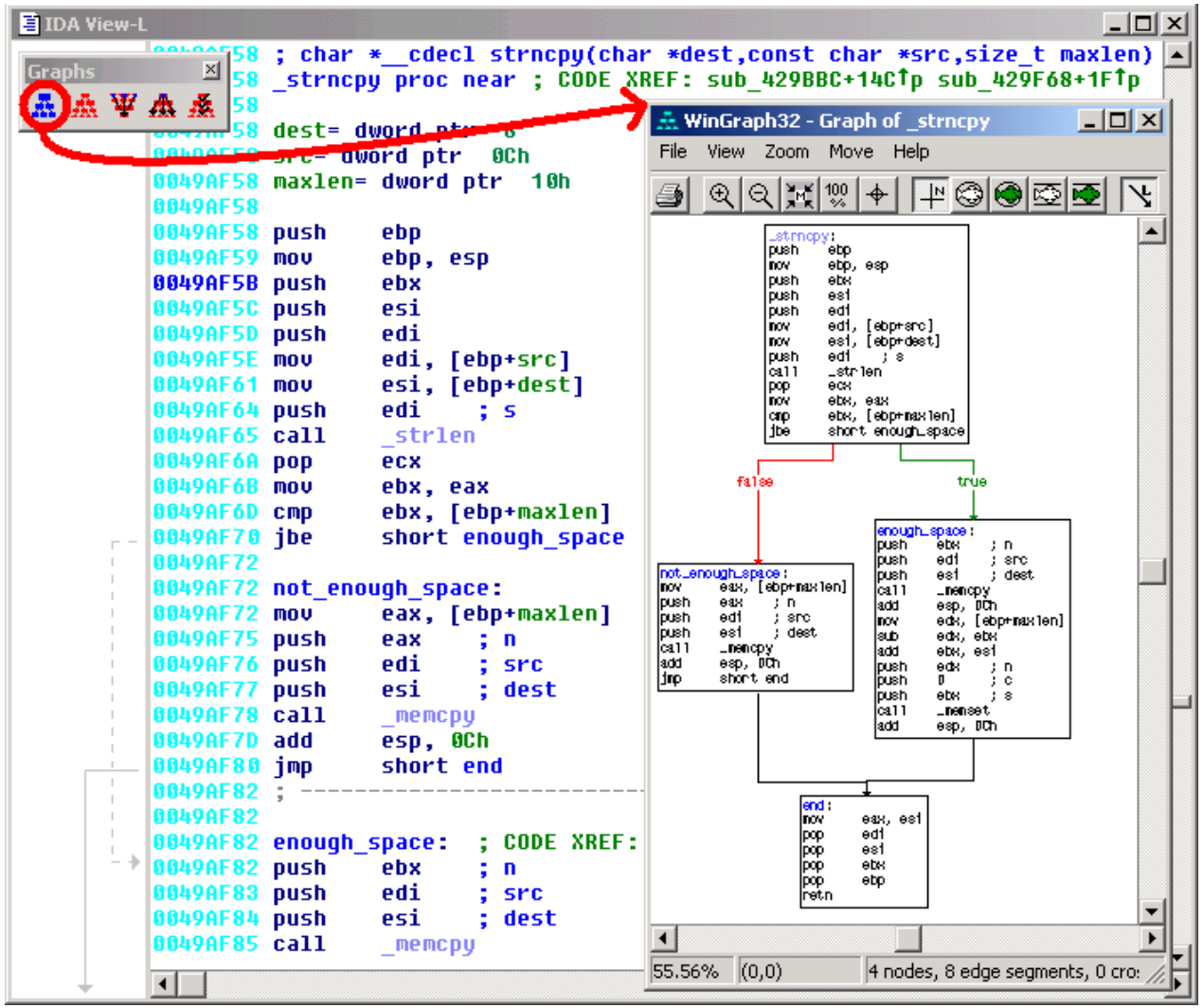
Instances

- Game Hacking
- Malware Analysis
- Cracking

Types of Analysis

Static vs Dynamic

Static Analysis



Dynamic Analysis

- Debugging

```
[0x08049090]> aaa
INFO: Analyze all flags starting with sym. and entry0 (aa)
INFO: Analyze all functions arguments/locals (afva@@@F)
INFO: Analyze function calls (aac)
INFO: Analyze len bytes of instructions for references (aar)
INFO: Finding and parsing C++ vtables (avrr)
INFO: Type matching analysis for all functions (aaft)
INFO: Propagate noreturn information (aanr)
INFO: Integrate dwarf function information
INFO: Use -AA or aaaa to perform additional experimental analysis
[0x08049090]> afl
0x08049090    1     44 entry0
0x080490bd    1      4 fcn.080490bd
0x08049040    1      6 sym.imp.__libc_start_main
0x080490f0    4     40 sym.deregister_tm_clones
0x08049130    4     53 sym.register_tm_clones
0x08049170    3     34 sym.__do_global_dtors_aux
0x080491a0    1      6 sym.frame_dummy
0x080490e0    1      4 sym.__x86.get_pc_thunk.bx
0x08049210    1     24 sym._fini
0x080490d0    1      5 loc..annobin_static_reloc.c
0x080491a6    6    101 dbg.main
0x08049080    1      6 sym.imp.errx
0x08049060    1      6 sym.imp.strcpy
0x08049070    1      6 sym.imp.puts
0x08049050    1      6 sym.imp.printf
0x08049000    3     36 sym._init
```

Tools

- IDA
- Ghidra
- Dogbolt.org

Demo

The Compilation Process

1. Compile
2. Assemble
3. Link

```
#include <stdio.h>

int main(){
        printf("hello world!\n");
}
```

Compile

hello_world.s

```
lea     rax, .LC0[rip]
mov     rcx, rax
call    puts
mov     eax, 0
add     rsp, 32
pop     rbp
ret
```

Assemble

hello_world.o

```
Contents of section .text:
 0000 554889e5 4883ec20 e8000000 00488d05  UH..H.. .....H..
 0010 00000000 4889c1e8 00000000 b8000000  ....H...........
 0020 004883c4 205dc390 90909090 90909090  .H.. ]..........
```

Link

$dir

10/20/2025	05:38 PM	128,829	hello_world.exe
10/17/2025	05:55 PM	66	hello_world.c
10/20/2025	05:36 PM	882	hello_world.o
10/20/2025	05:34 PM	571	hello_world.s

Know Your Compiler!

So Much Added Code!

```
int WinMainCRTStartup(void);
int mainCRTStartup(void);
int32_t atexit(void (*func)(void));
int64_t __gcc_register_frame(void);
void __gcc_deregister_frame(void) __pure;
int64_t main(void);
```

Find the Relevant Code

```
int64_t main() {
        __main();
        puts("hello world!");
        return 0;
}
```

Thanks!

github.com/AliGhaffarian/QUT-reverse-engineering-competition