Lab 4
Rehosting & ROP

HS 2020

The Goals

- Carry out function level rehosting
  - And Instrumentation!
- Challenge your creativity

Assignment 4A

- Get familiar with ARM ISA
- Craft your first ARM ROP exploit

Assignment 4B

Assignment 4A

```
uint32_t modpow(uint32_t b, uint32_t e)
{
   uint32_t base = b % MODULO;
   uint32_t res  = 1;

   for(int i=0; i<e; i++)
   {
       res = (res * base) % MODULO;
       thread_sleep_for(e);    //Sleep "e" ms
   }
   return (char)(res & 0xff);
}

void assignment_rehosting()
{
   printf("Welcome to assignment REHOSTING!\r\n");

   for(int i=0; i<sizeof(exponents)/sizeof(uint32_t); i++)
   {
       putchar(modpow(BASE, exponents[i]));
       fflush(stdout);
   }
}
```

Assignment 4A

Unicorn

The Ultimate CPU emulator

Download Docs Showcase Contact

Unicorn is a lightweight multi-platform, multi-architecture CPU emulator framework.

Highlight features:

- Multi-architectures: Arm, Arm64 (Armv8), M68K, Mips, Sparc, & X86 (include X86_64).
- Clean/simple/lightweight/intuitive architecture-neutral API.
- Implemented in pure C language, with bindings for Pharo, Crystal, Clojure, Visual Basic, Perl, Rust, Haskell, Ruby, Python, Java, Go, .NET, Delphi/Pascal & MSVC available.
- Native support for Windows & *nix (with Mac OSX, Linux, *BSD & Solaris confirmed).
- High performance by using Just-In-Time compiler technique.
- Support fine-grained instrumentation at various levels.
- Thread-safe by design.
- Distributed under free software license GPLv2.

Find in this BlackHat USA 2015 slides more technical details behind Unicorn engine.

Unicorn is based on QEMU, but it goes much further with a lot more to offer.

https://www.unicorn-engine.org/

Assignment 4A

Unicorn

The Ultimate CPU emulator

Download Docs Showcase Contact

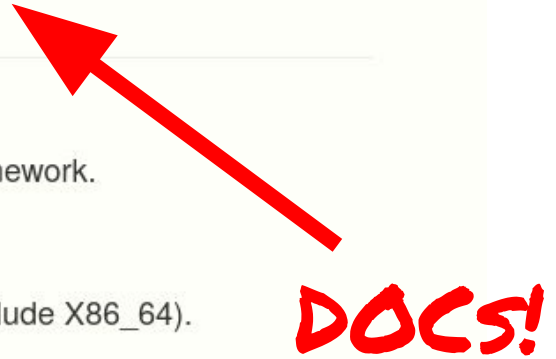
Unicorn is a lightweight multi-platform, multi-architecture CPU emulator framework.

Highlight features:

- Multi-architectures: Arm, Arm64 (Armv8), M68K, Mips, Sparc, & X86 (include X86_64).
- Clean/simple/lightweight/intuitive architecture-neutral API.
- Implemented in pure C language, with bindings for Pharo, Crystal, Clojure, Visual Basic, Perl, Rust, Haskell, Ruby, Python, Java, Go, .NET, Delphi/Pascal & MSVC available.
- Native support for Windows & *nix (with Mac OSX, Linux, *BSD & Solaris confirmed).
- High performance by using Just-In-Time compiler technique.
- Support fine-grained instrumentation at various levels.
- Thread-safe by design.
- Distributed under free software license GPLv2.

Find in this BlackHat USA 2015 slides more technical details behind Unicorn engine.

Unicorn is based on QEMU, but it goes much further with a lot more to offer.



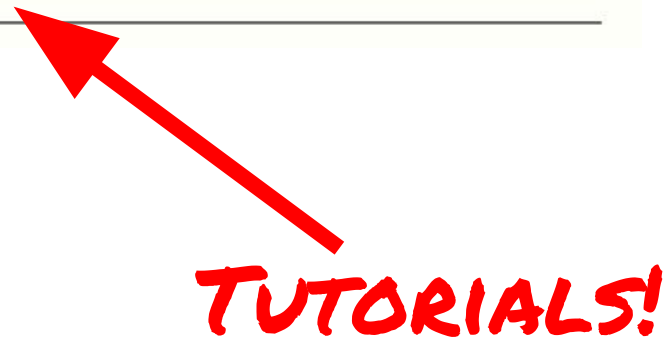
https://www.unicorn-engine.org/

Assignment 4A

Programming

After installation, find in tutorials below how to write your tools based on Unicorn using your favorite programming languages.

- Quick tutorial on programming with Unicorn - with C & Python.



https://www.unicorn-engine.org/docs/

Assignment 4A - The plan

1. Start GDB, set breakpoint at assignment_rehosting, and reach it.
2. Dump the firmware, the ram, and the register values to create a snapshot of the current execution context.
3. Load the snapshot in unicorn engine (c.f. sample_arm.py).
4. Create a hook for putchar
5. Create a hook for thread_sleep_for
6. Create hooks for all the remaining code that you want to skip
7. Start emulation and get the flag!

Assignment 4A - Part II

- You learned rehosting via unicorn
- This challenge can be solved in various other ways
  - Find, implement, and report 2 other ways!
  - Be creative!

Assignment 4B

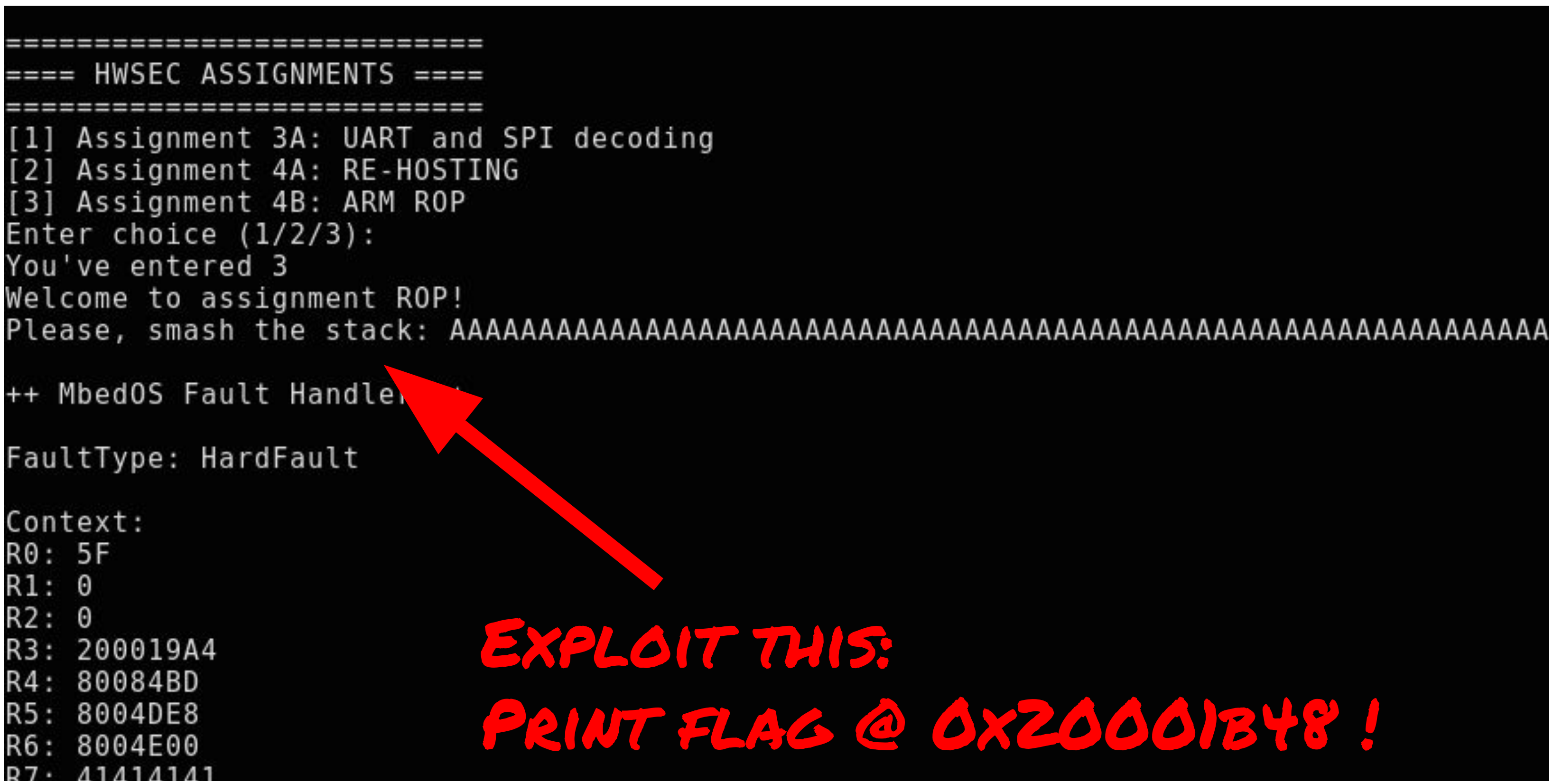
```
============================
==== HWSEC ASSIGNMENTS ====
============================
[1] Assignment 3A: UART and SPI decoding
[2] Assignment 4A: RE-HOSTING
[3] Assignment 4B: ARM ROP
Enter choice (1/2/3):
You've entered 3
Welcome to assignment ROP!
Please, smash the stack: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

++ MbedOS Fault Handler ++

FaultType: HardFault

Context:
R0: 5F
R1: 0
R2: 0
R3: 200019A4
R4: 80084BD
R5: 8004DE8
R6: 8004E00
R7: 41414141
```

Assignment 4B



But how do I find gadgets?

```
$ arm-none-eabi-objdump -d hwsec.elf
```

Tools:

- ROPgadget
- ropper
- radare2
- And others

Point distribution

- **4A Part I** (4 points)
  - Rehosting via Unicorn
- **4A Part 2** (2 points)
  - 1 point per alternative solution
- **Part 4B** (4 points)
  - ROP Exploit for the Firmware

Questions?