s0fT7: Revealing the Secrets of the Siemens S7 PLCs

Sara Bitan | Alon Dankner

Joint work with **Professor Eli Biham**, **Maxim Barsky** and **Idan Raz**

Faculty of Computer Science, Technion – Israel Institute of Technology

Who are we?

Sara Bitan

Founder and CEO of CyCloak: Secure system design and audit

Senior researcher at the Technion Hiroshi Fujiwara Cyber Security Research Center

Alon Dankner

Security researcher
M.Sc. graduate from the Technion

Advisors: **Prof. Eli Biham, Dr. Sara Bitan**

PLCs turn rogue…

Stuxnet
(Anonymous author)

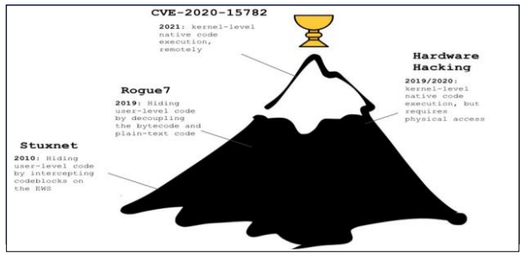
- Exploit a vulnerable Siemens Step7 engineering station/ WinCC HMI client
- Inject a rogue control program, and tamper with HMI outputs

Rogue7: Rogue Engineering-Station attacks on S7 Simatic PLCs
(Biham, et al)

- A phyton script impersonating an engineering WS
- All S7 PLCs from the same model and firmware version share the same key

Doors of Durin: The Veiled Gate to Siemens S7 Silicon
(Abbasi, et al)

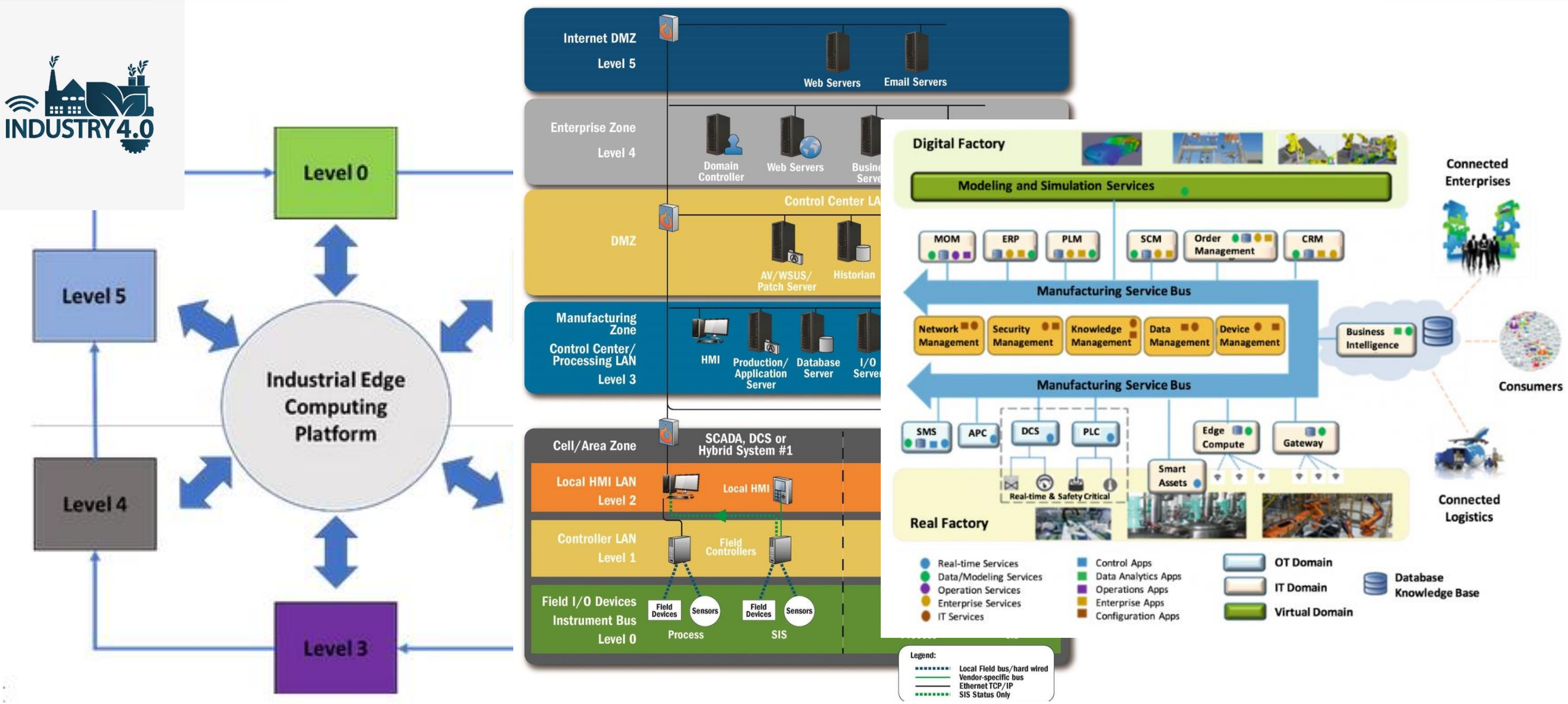
- Siemens S7-1200 PLC Bootloader Arbitrary Code Execution
- Siemens S7 firmware is using Adonis kernel



The Race to Native Code Execution in PLCs
(Keren)

- Remote arbitrary code execution on Siemens S7-1500
- Exploiting memory protection vulnerability to escape the control program sandbox

ICS architectures are evolving



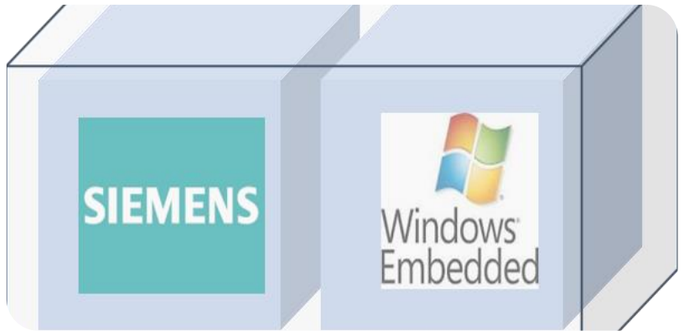
Software PLCs

Smart Manufacturing

- New requirements from PLC vendors
- New features: IDEs, new protocols, extensive cloud communication

Vendor Requirements

- Agility and flexibility
- Preserve existing IP and technology
- The solution: software PLCs



New PLC architecture

- Generic functions: GP OS – updatable, flexible→ Standard hardware
- Legacy functions : proprietary OS - closed and hardened
- Virtualization: isolation and separation

Siemens ET 200SP open controller

- The PC-based version of SIMATIC S7-1500
  - Introduced in January 2015
- Combines PLC functionality with a PC-based platform using virtualization
- Isolation between Windows and control logic
  - Supports Windows updates and reboot without interruption to the control logic
  - The controller continues to work even if Windows crashes
- DUT: CPU 1515SP PC2

The quest...

The boot process


```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ lsblk /dev/sda
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 119.2G  0 disk
├─sda1   8:1    0  58.6G  0 part
├─sda2   8:2    0  15.6G  0 part
├─sda3   8:3    0  44.6G  0 part
├─sda4   8:4    0     1K  0 part
└─sda5   8:5    0   400M  0 part
ubuntu@ubuntu:~$
```

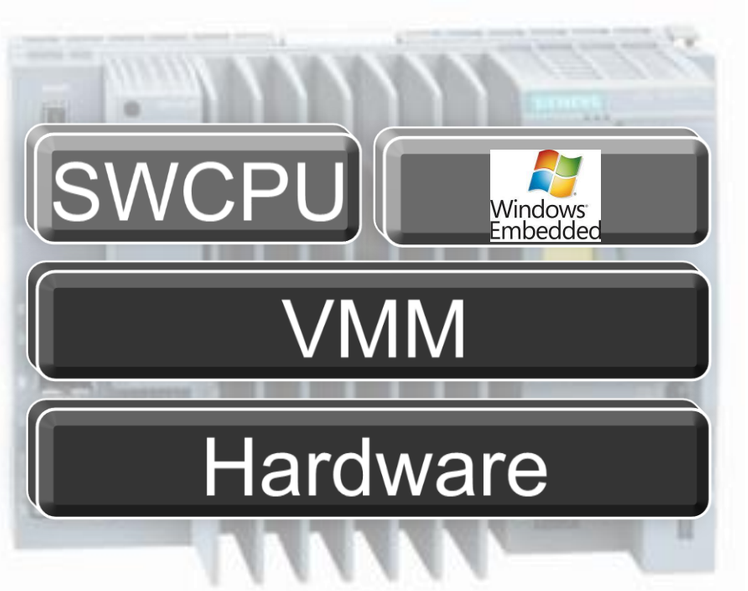
Windows Embedded

```
ubuntu@ubuntu: /mnt/TIA_project_files
ubuntu@ubuntu:/mnt/TIA_project_files$ ll
total 34
drwxr-xr-x 4 root root   512 Jan  1  1970 ./
drwxr-xr-x 1 root root    60 Oct  5 01:02 ../
-r-xr-xr-x 1 root root 32768 Jan  1  1980 __LOG__*
drwxr-xr-x 2 root root   512 Jan 11  2021 ODK1500S/
-rwxr-xr-x 1 root root     7 Jan 11  2021 S7_JOB.S7S*
drwxr-xr-x 3 root root   512 Jan 11  2021 SIMATIC.S7S/
ubuntu@ubuntu:/mnt/TIA_project_files$
```

The boot process

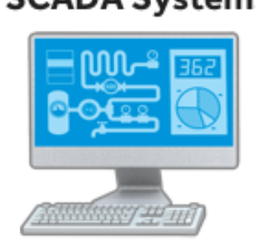
- The GRUB configuration file

```
26  menuentry   'Windows and S7-1500 Software Controller'   --class matches   --class icon-swcpu  {
27     set vmm_dir=/Boot/Siemens/SIMATIC_RT_VMM
28     set boot_partition_file=vmm_boot.000
29     set system_partition_file=vmm_system.000
30     set vmdid=1
31     set swcpu_dir=/Boot/Siemens/SWCPU
32     set swcpu_file=CPU.elf
33     set swcpu_configuration_file=vmm_cpu.cfg
34     getpartition  file  $vmm_dir/$boot_partition_file
35     vmm_multiboot ($root)$vmm_dir/VMM_1st_stage.elf
36     if [ $? = 0 ] ; then
37        vmm_module ($root)/$swcpu_dir/$swcpu_configuration_file
38        vmm_module ($root)$vmm_dir/VMM_2nd_stage.elf
39        getpartition file $vmm_system_dir/$system_partition_file
40        vmm_module ($root)$winfile
41        getpartition file $vmm_dir/$boot_partition_file
42        vmm_module ($root)$swcpu_dir/$swcpu_file :p pagedir_mem_reg_id=0 vmdid=$vmdid
43        workaround_for_scrambled_screen      boot
44     else
45        echo Hypervisor not found!
46     fi
47  }
```



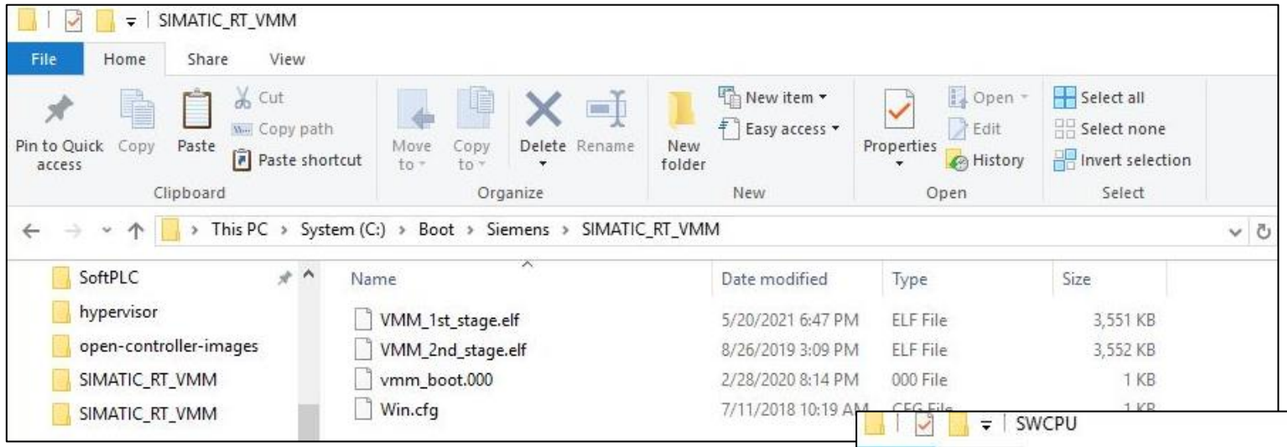
Open controller boot sequence

BIOS → GRUB Boot-loader → 1st Stage VMM → 2nd Stage VMM → Windows Embedded / SWCPU

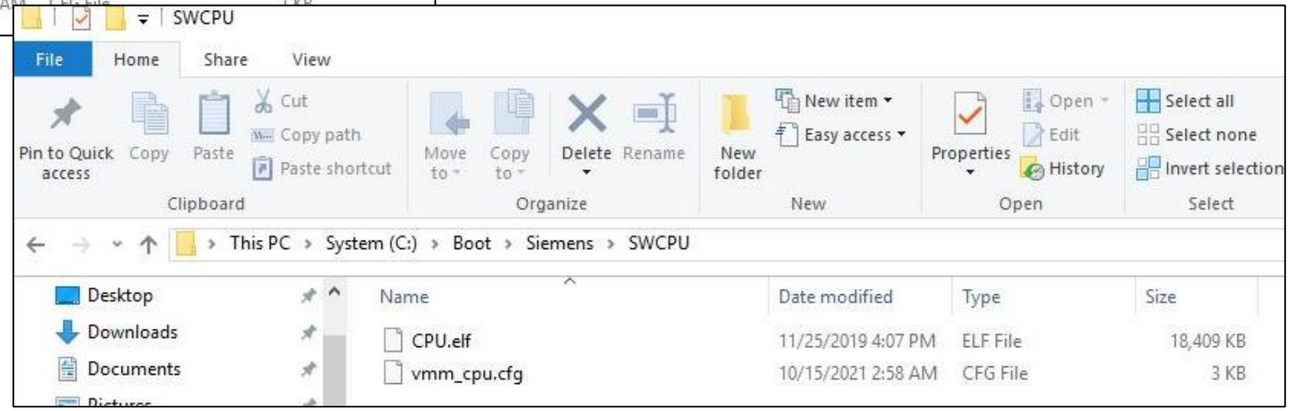


The boot process

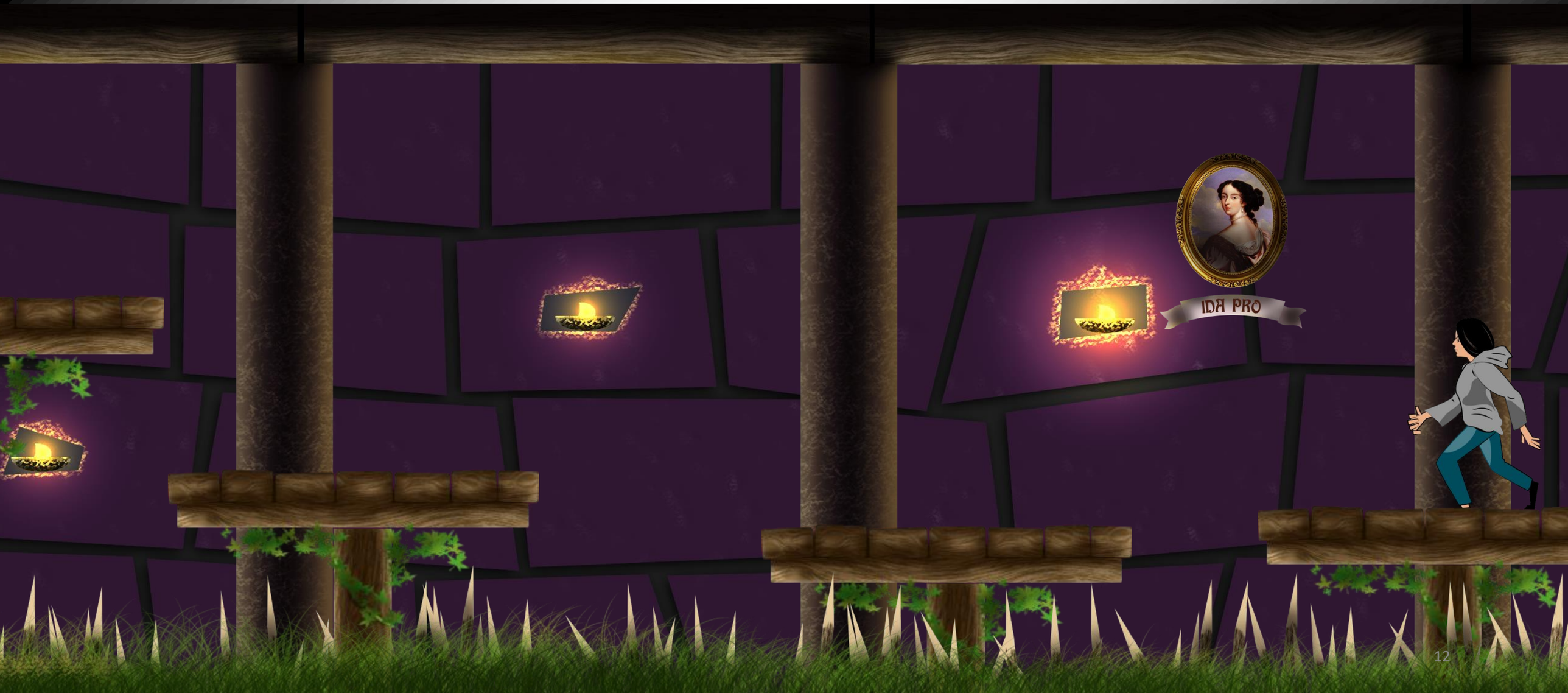
- Surprisingly, the VMM binary, grub configuration and CPU.elf files are also accessible from Windows



- RW by admin



The quest...



Finding the ELF decryption code

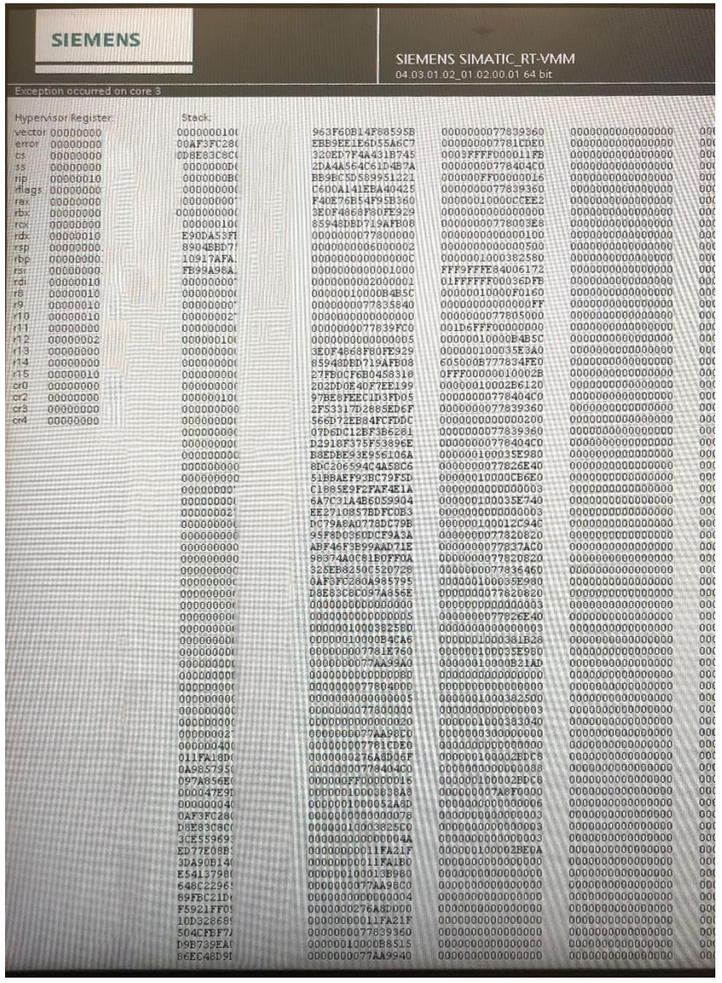
```
.text:00000010          call    sub_1000
.text:00000010          mov     r15, rax
.text:00000010
.text:00000010  loc_10  :                       ; CODE XREF: check_header_and_decompress_elf+26↑j
.text:00000010          mov     qword ptr [rbx+20h], 0
.text:00000010          movzx   eax, byte ptr [r12]
.text:00000010          cmp     al, 7Fh
.text:00000010          jz      short loc_10
.text:00000010          cmp     al, 53h ; 'S'
.text:00000010          jnz     short loc_10
.text:00000010          cmp     byte ptr [r12+1], 33h ; '3'
.text:00000010          jz      short loc_10
.text:00000010          nop     dword ptr [rax]
.text:00000010
.text:00000010  loc_10  :                       ; CODE XREF: check_header_and_decompress_elf+53↑j
.text:00000010                                  ; check_header_and_decompress_elf+7A↓j ...
.text:00000010          lea     rdi, aErrorLoadingEl_0 ; "Error loading elf file (%s): invalid ma"...
.text:00000010          mov     rsi, r15
.text:00000010          xor     eax, eax
.text:00000010          call    print
.text:00000010          int     3               ; Trap to Debugger
.text:00000010  ; ---------------------------------------------------------------------------
.text:00000010  ; ---------------------------------------------------------------------------
.text:00000010
.text:00000010  loc_10  :                       ; CODE XREF: check_header_and_decompress_elf+5B↑j
.text:00000010          cmp     byte ptr [r12+2], 5Eh ; '^'
.text:00000010          jnz     short loc_10
.text:00000010          cmp     byte ptr [r12+3], 9Fh
.text:00000010          jnz     short loc_10
.text:00000010          movzx   eax, byte ptr [r12+4]
.text:00000010          sub     dword ptr [rsp+98h+code_size], 4
.text:00000010          movzx   edx, al
.text:00000010          lea     r12, [r12+rax+4]
.text:00000010          mov     rax, cs:off_10
.text:00000010          sub     dword ptr [rsp+98h+code_size], edx
.text:00000010          mov     rdi, [rax]
.text:00000010          cmp     qword ptr [rdi+110h], 2
.text:00000010          jbe     short loc_10
.text:00000010          lea     rdi, aInvalidConfigu ; "Invalid configuration for elf file (%s)"
.text:00000010          mov     rsi, r15
.text:00000010          xor     eax, eax
```

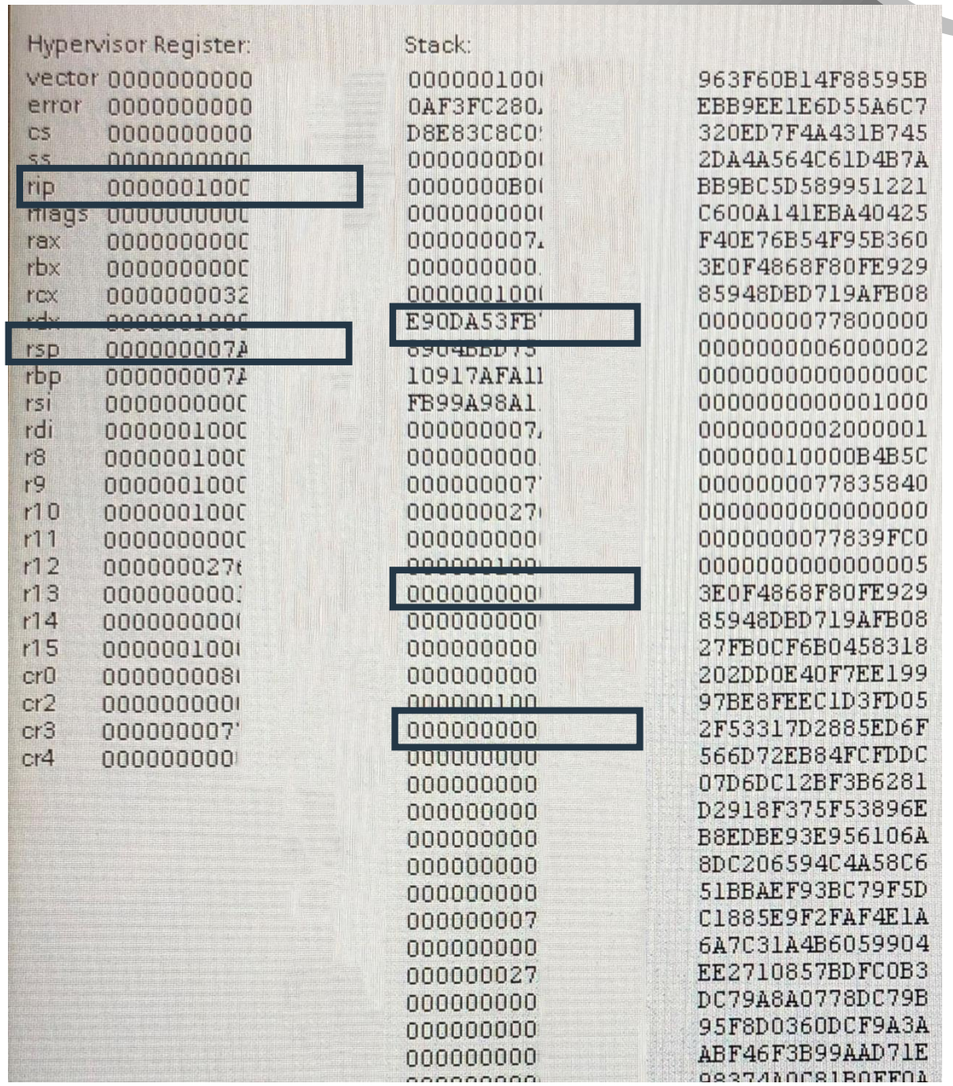
```
usr@VM: ~
usr@VM:~$ xxd CPU.elf | head -20
00000000: 5333 5e9f 6b9d ce3d 0a80 300c 05e0 1c25  S3^.k..=..0....%
00000010: [illegible]  041d a583 ab50 9c9d aa16 fc43  ....P.....C
00000020: 1df4 485e c393 5975 7012 c16f 0819 1ebc  ..H^..Yup..o....
00000030: a7ad 694d 3781 ca72 e490 160d 0f49 8222  ..iM7..r.....I."
00000040: 41ee 5211 0482 f8fa 25bc 91c4 b1cf eccb  A.R.....%.......
00000050: 1040 f5c3 3ada aa9e 71df 50df 2598 a41e  .@..:...q.P.%...
00000060: 9e21 f8e6 f608 d394 f0d7 017e b217 fc8d  .!.........~....
00000070: bf14 bd17 59fe e385 9c36 5e5b 87ec 51a0  ....Y....6^[..Q.
00000080: 31fc fda0 5f80 f656 4f00 2714 b2f9 6fc4  1..._..VO.'...o.
```

The quest...

The quest...

The “Int 3” debugger





Intermediary status check

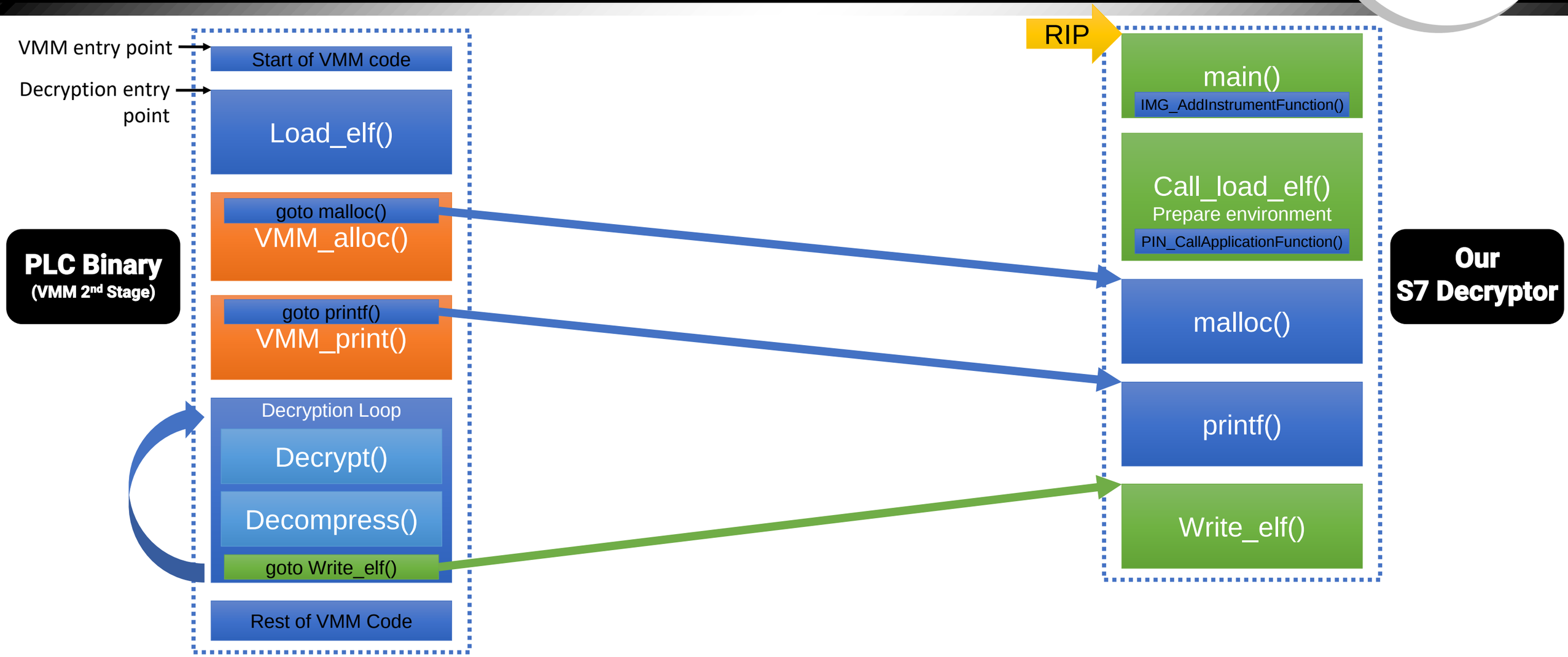
- Found the decompression/decryption function
- Static analysis – very complex
- Debugging
  - Using Int3 debugger
    - We have the decrypted swcpu in memory
    - But cannot export it from the PLC, for analysis

Stealing the vault

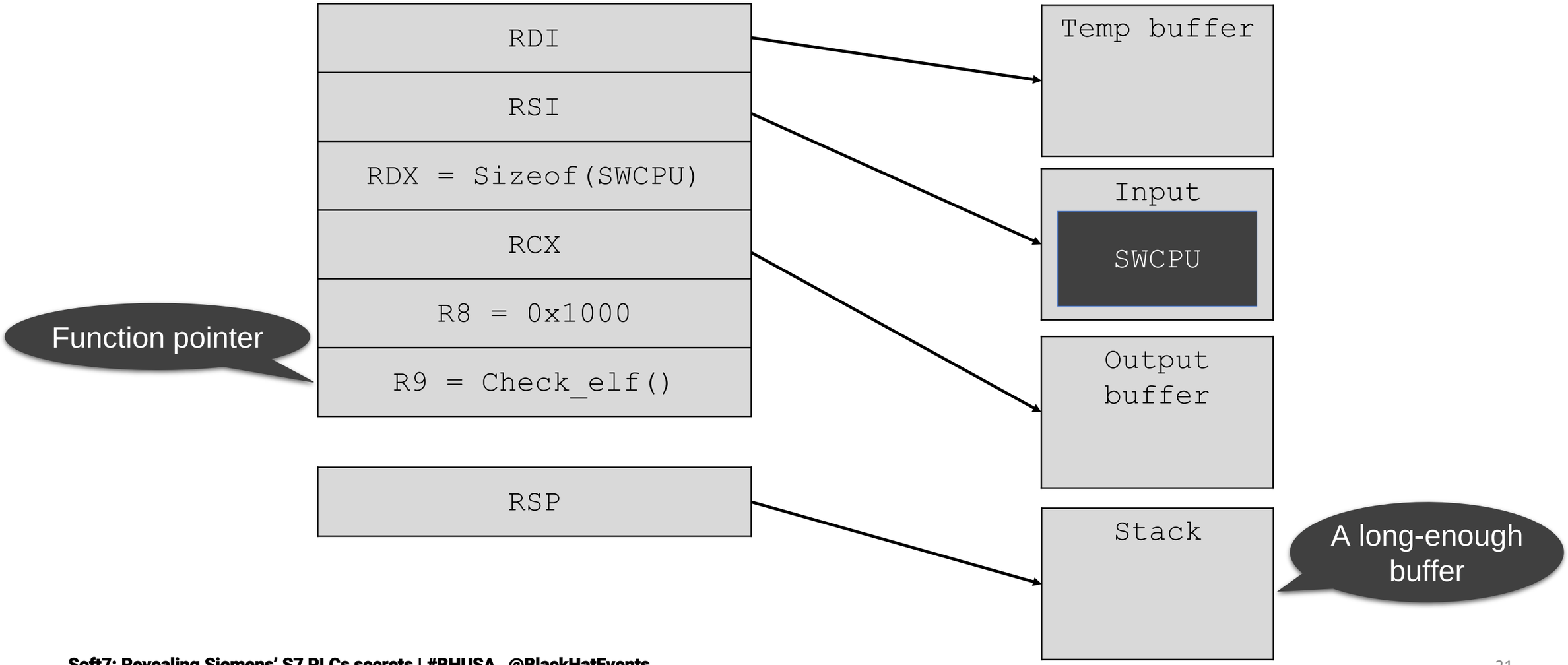
Unlocking the vault

- The VMM is an x86 binary → We run it on standard Linux (Ubuntu)
- Challenge: different execution environment
  - VMM runs in hypervisor mode, we run it in user mode
  - Siemens proprietary VMM run time library vs. standard CRT
- Solution: dynamic binary instrumentation
  - Start from a specific instruction
  - Replace VMM functions
  - Add our code
- We used Intel Pin to run the VMM decryption

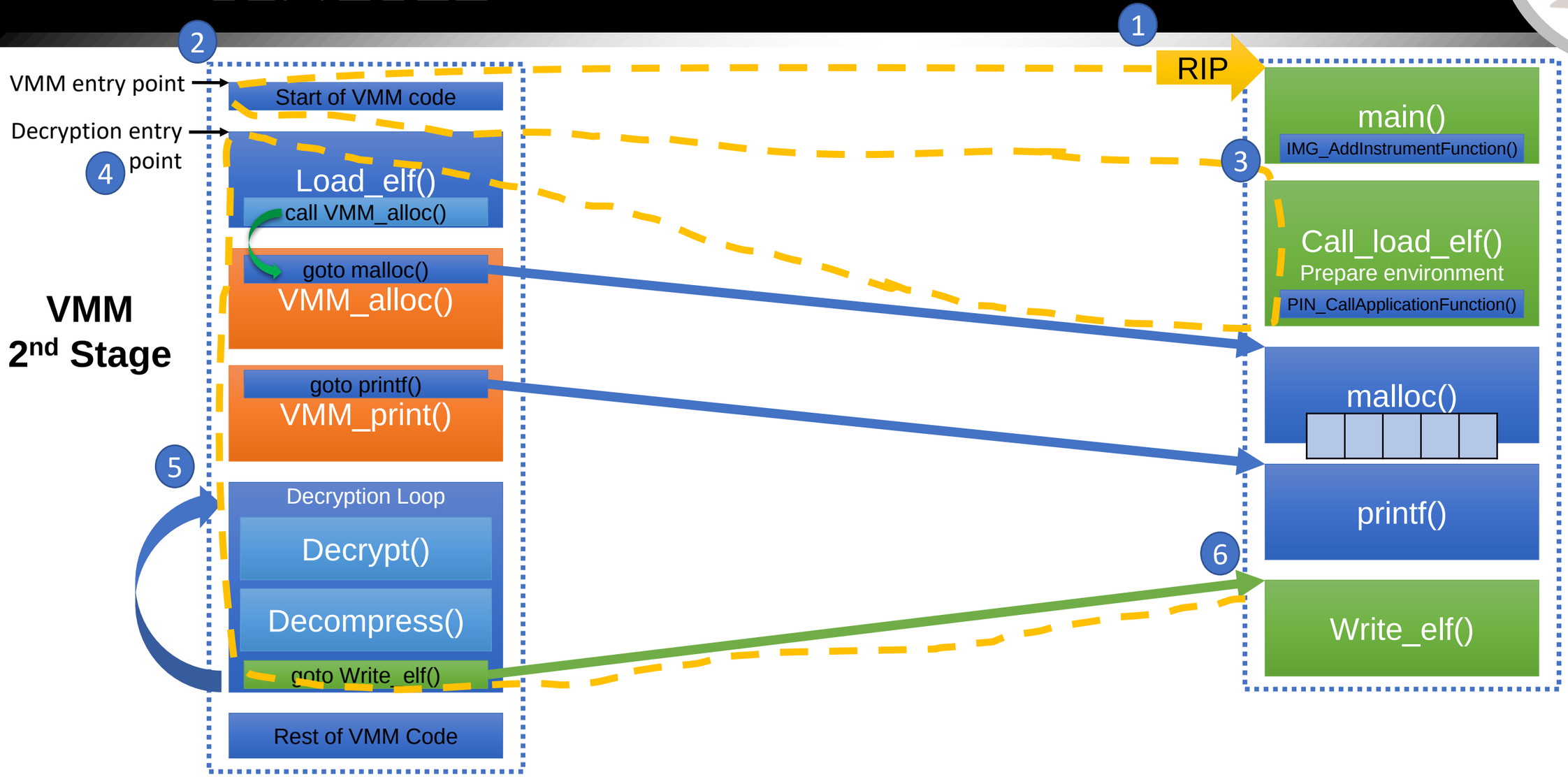
PLC binary instrumentation



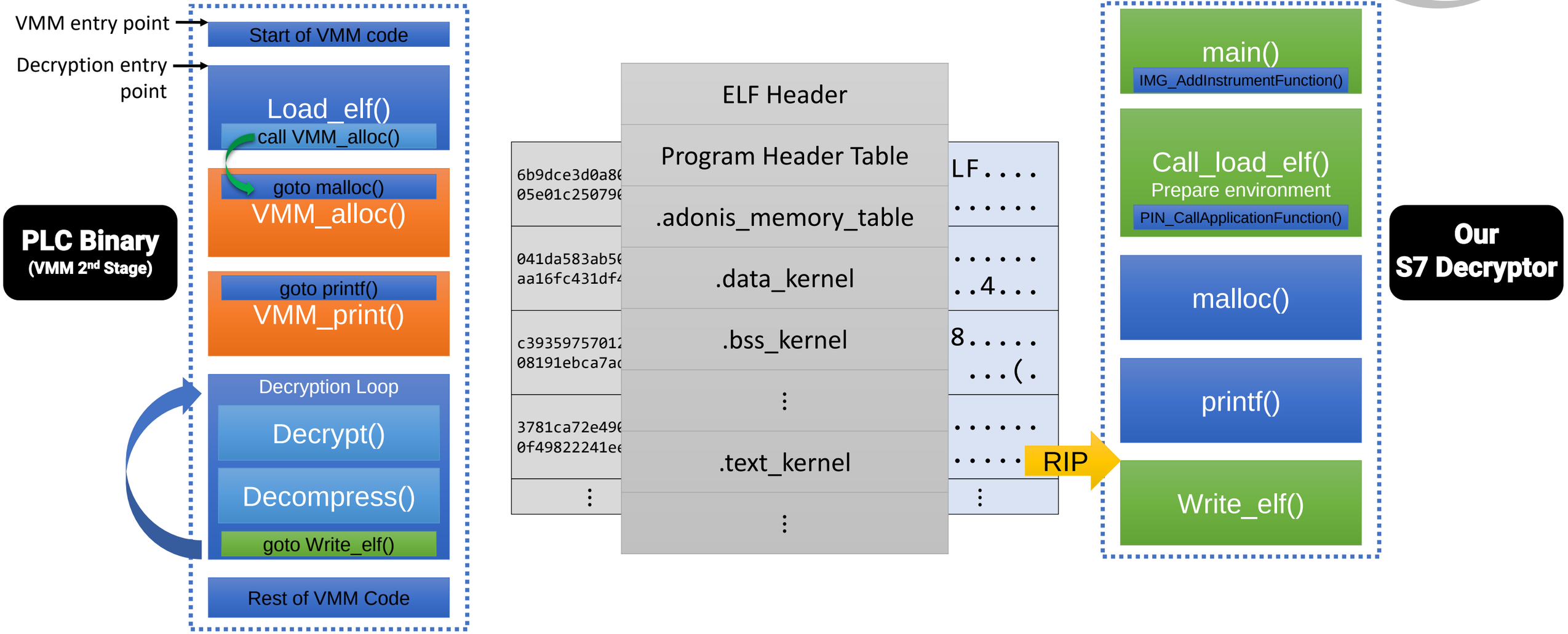
Environment



VMM binary instrumentation

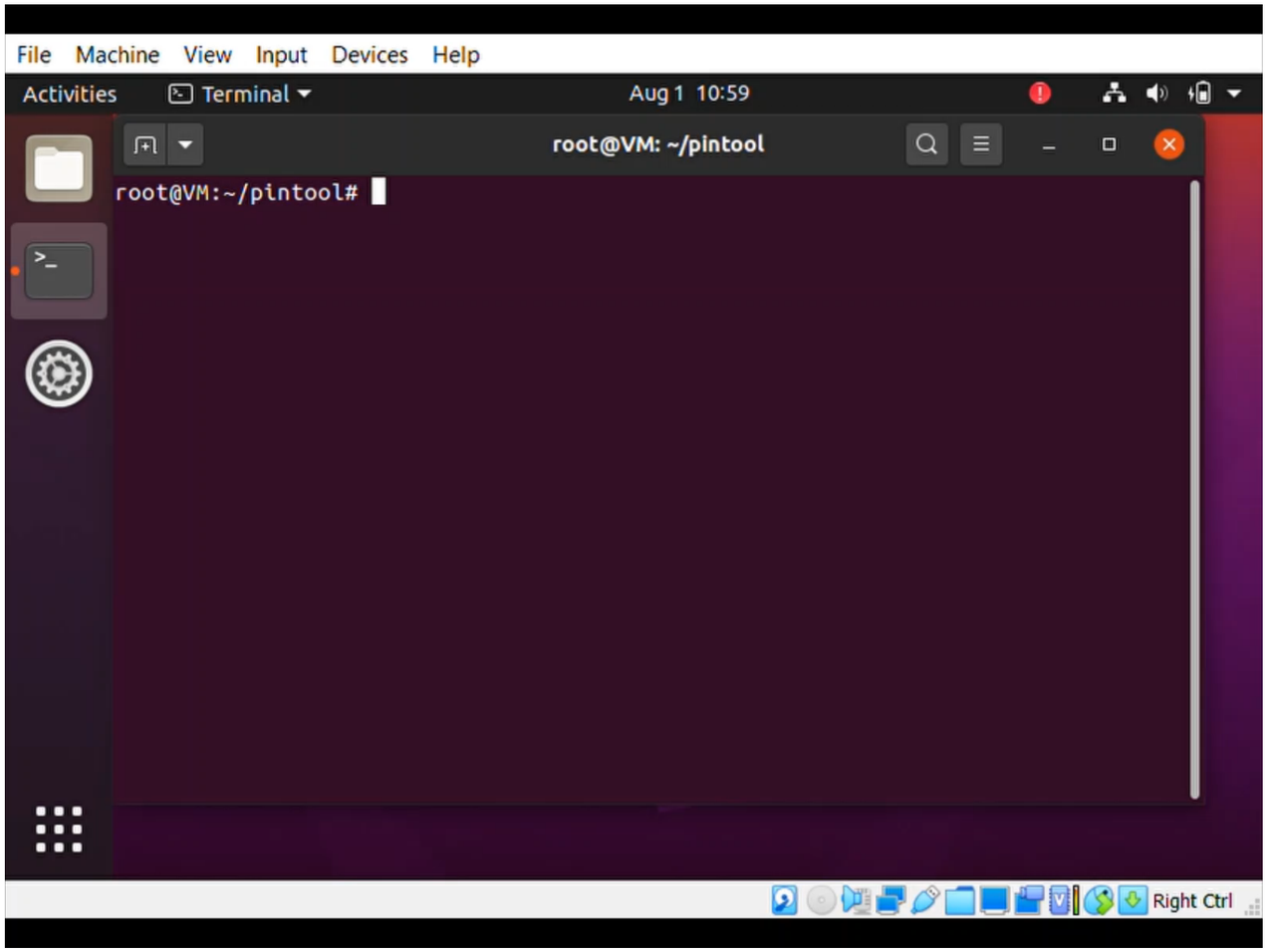


PLC binary instrumentation



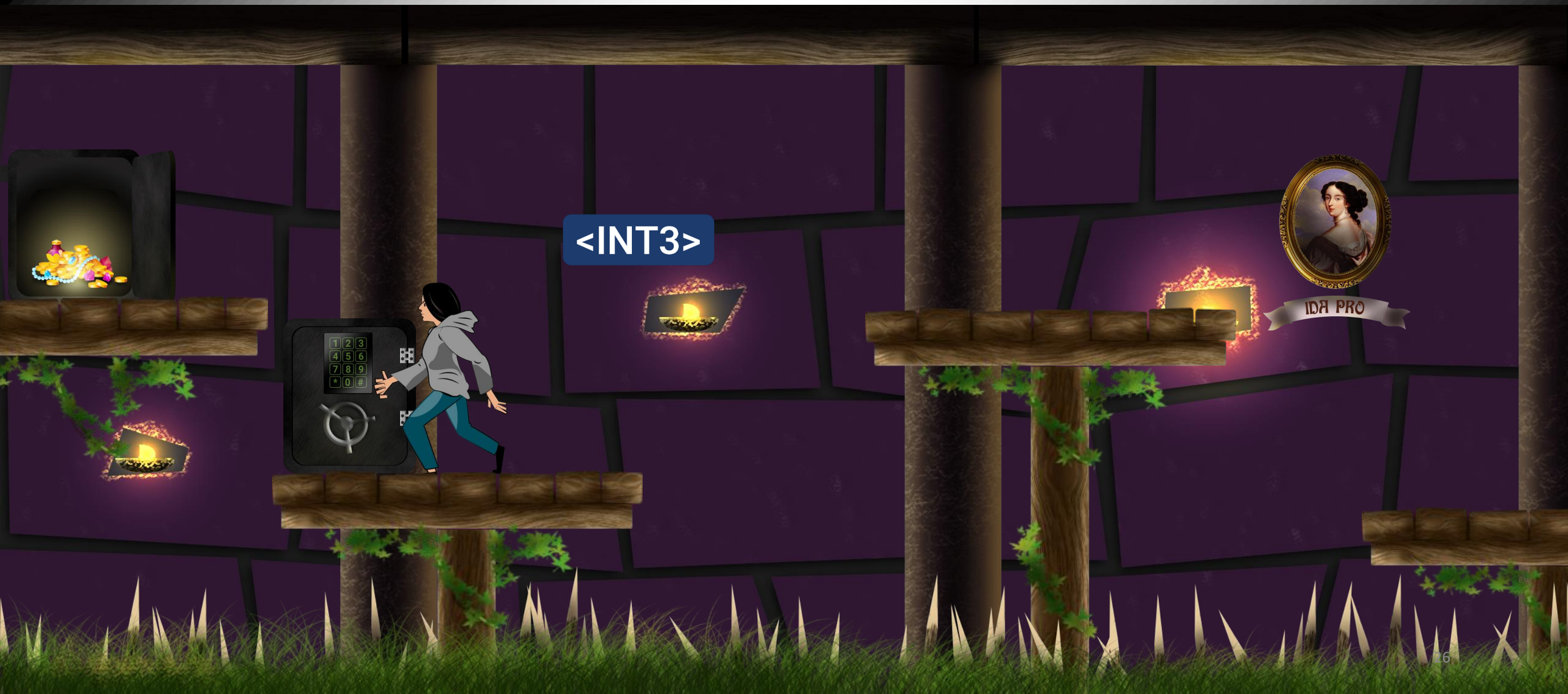
Demo

- Running PLC binary (VMM 2nd stage) on our Ubuntu machine

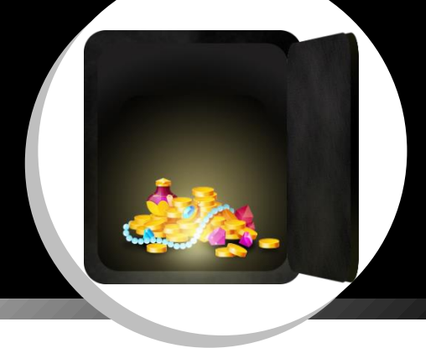


The quest...

The quest...



The Decrypted firmware



- Our initial research shows that SWCPU is based on the Adonis Linux
- Contains far more than the basic kernel + PLC code:
  - Stand-alone libc.so
  - openSSL
  - tar archive called “winac_bb_soc1” with a MIPS ELF inside
  - Strings from other S7 Simatic PLCs

```
42  FUN_10c09ec0("Booting ADONIS x86_64\n\n");
43  *(undefined8 *)(uVar2 - 8) = 0x10c02faa;
44  FUN_10c09ec0("Using ... \n");
45  *(undefined8 *)(uVar2 - 8) = 0x10c02fb9;
46  FUN_10c09ec0("... 64-bit mode\n");
```

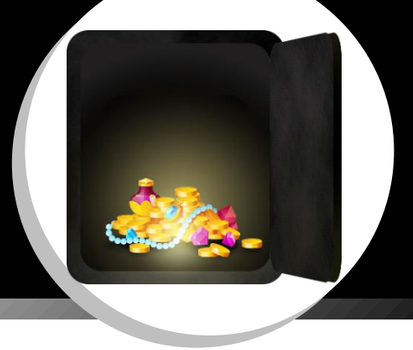
```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             ELF, 32-bit LSB executable, Intel 80386, version 1 (SYSV)
56954477      0x3650E6D       ELF, 32-bit LSB shared object, AMD x86-64, version 1 (SYSV)
57156624      0x3682410       POSIX tar archive, owner user name: "_soc1/"
```

The quest...

The quest...

The quest...

Mitigations



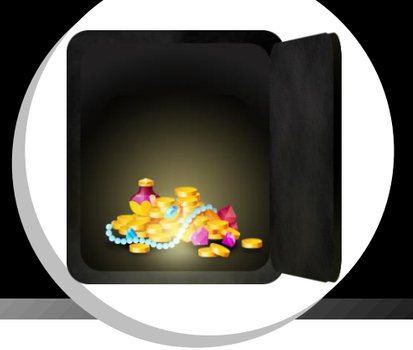
Separating the key from the code:
prevents decryption with PIN

Prevents INT3 debugging

Prevents Ubuntu booting

Prevents static code reversing

Customer impact



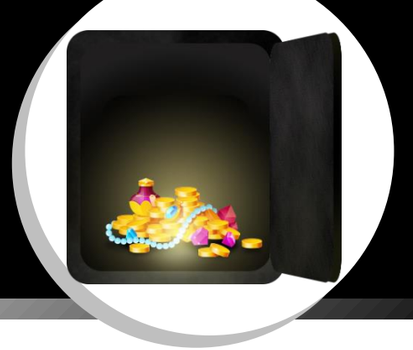
! PLC firmware leakage exposes the full Simatic S7 product line

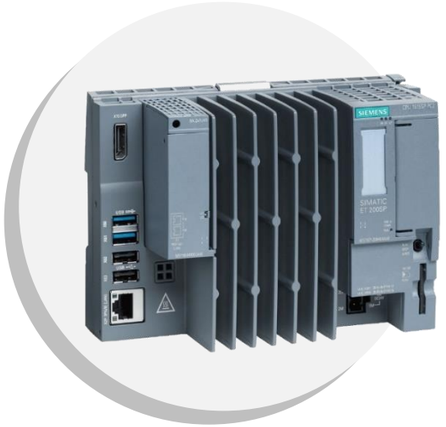
- Via exploitation of known vulnerabilities
- The horses may have already left the stable…

! Recent finding (future publication)

- An attacker who gains admin rights on the Windows VM can replace the PLC firmware with his own crafted rogue PLC firmware
- We shared the full details with Siemens

Soft7 Summary





31%

Siemens PLC market share (2019)

Deployment

Power plants, water facilities, transportation systems, nuclear reactors

Firmware leakage

↓

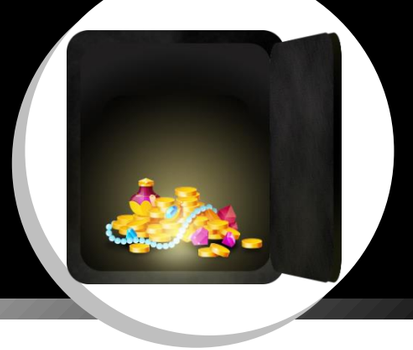
Exposure to known unpatched vulnerabilities

A design flaw

↓

No easy solution

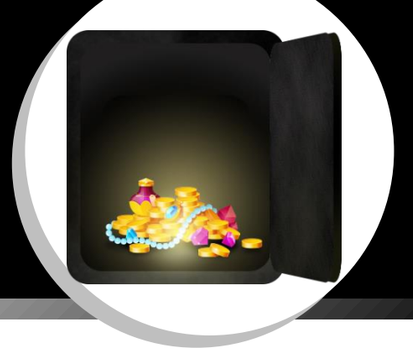
Message to the community



Message to the security & research community

- Secure binding to hardware and large-scale key management are tough operational problems
- This is challenge to the security & research community
  - Especially important since ICS architecture currently shifting from walled garden to open and cloud-oriented environments
- A solution is crucial!!!

Message to customers



Message to the customers of all ICS vendors

- You are the assets owners!
- You will suffer from the impact!
- Demand the security you need from the ICS vendors!
  - Otherwise, you get “generic” security features that do not fit your full requirements

Thank you!

Sara Bitan | Alon Dankner

Sarab@cycloak.com | Dankner@cs.technion.ac.il