HITBSECCONF2018

DUBAI


HITBSECCONF2018
OFFENSIVE MEMORY FORENSICS
DUBAI

MAIN MENU

TESO

STORY MODE

TRAINING MODE

VS MODE

GOD MODE

TUOMINEN

TESO

MAIN MENU

STORY MODE

TRAINING MODE

VS MODE

GOD MODE

TUOMINEN

TESO

VS

TUOMINEN

MAIN MENU

STORY MODE

TRAINING MODE

VS MODE

GOD MODE

TESO

TUOMINEN


TESO
KO
TRAINING
TUOMINEN
Know yourself...
Know your enemy...

TRAINING

1P

2P

TESO

KO

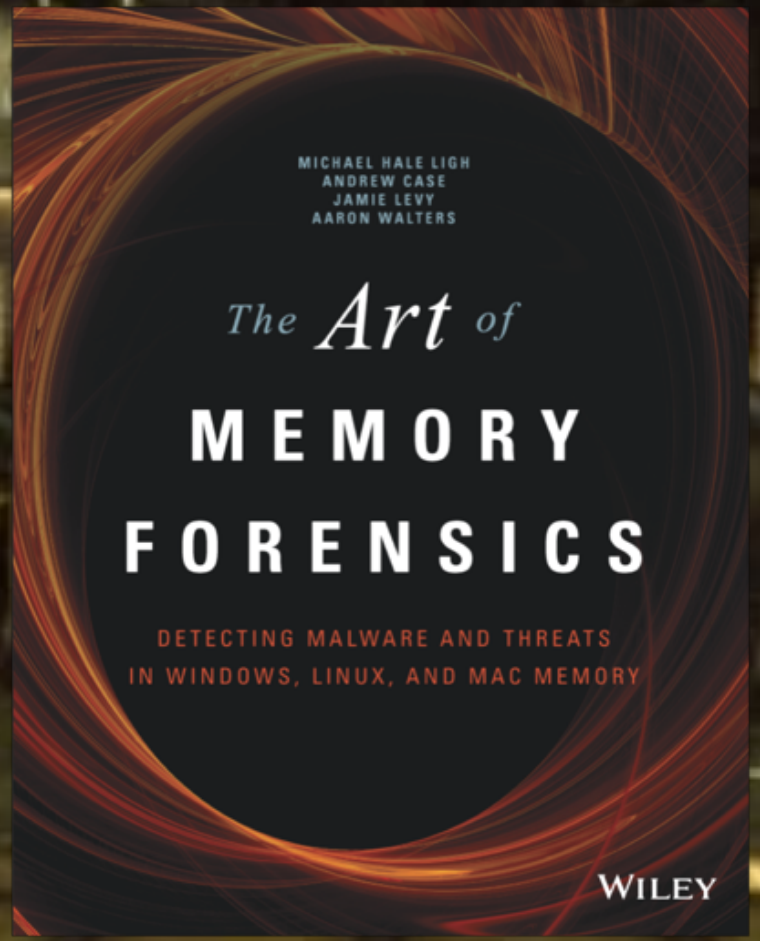
TRAINING

TUOMINEN

No you didn't

I wrote that book

No I didn't...



TESO

KO

TRAINING

TUOMINEN

Memory Forensics 101

- Memsics: Memory + Forensics
- One part of DIGITAL FORENSICS
- Analysis of VOLATILE DATA

TESO

KO

TRAINING

TUOMINEN

MEMORY - FORENSICS

TESO

KO

TRAINING

TUOMINEN

MEMORY - FORENSICS

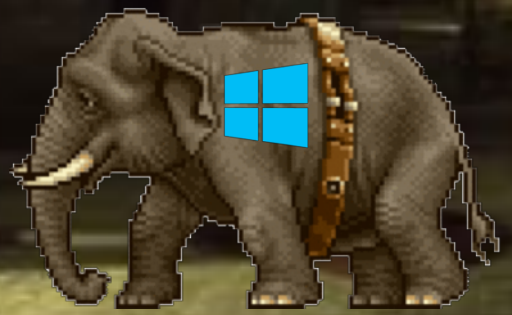
TESO

KO

TRAINING

TUOMINEN

WHICH MEMORY?



TESO

KO

TRAINING

TUOMINEN

Meh...

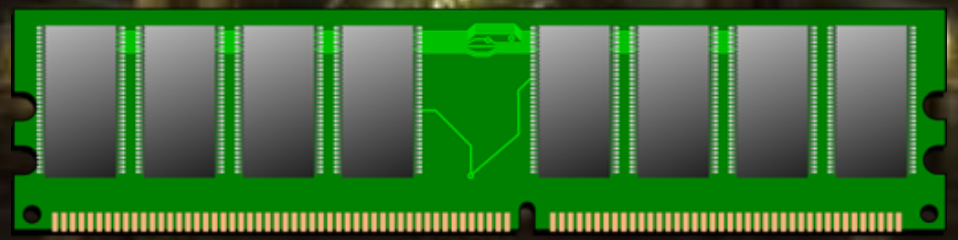
TESO

KO

TRAINING

TUOMINEN

In the beginning was...



RAM

TESO

KO

TRAINING

TUOMINEN

OS (Memsics) doesn't work with "RAM"

VIRTUAL MEMORY

TESO

KO

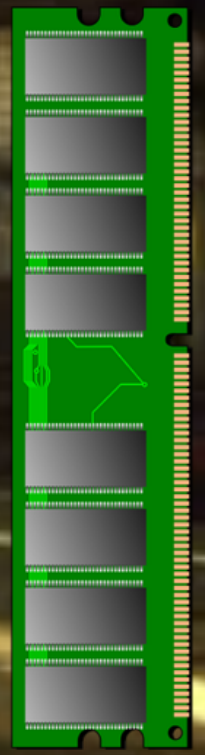
TRAINING

TUOMINEN

OS (Memsics) doesn't work with "RAM"

VIRTUAL MEMORY

=



RAM

+ ...

TESO

KO

TRAINING

TUOMINEN

Wanna know more?

Virtual Memory, in Windows ,is actually a polymorphic term.

- VM = Physical memory + Page file
- VM = the collection of Pages (4KB segments) scattered in memory of a process working set

TESO

KO

TRAINING

TUOMINEN

Virtual Memory, in Windows ,is actually a **polymorphic** term.

- VM = Physical memory + Page file

TESO

Old fart bad jokes

KO

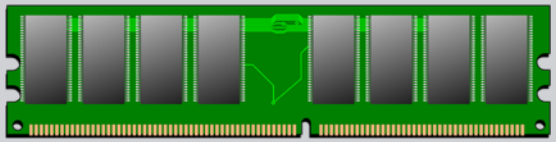
TRAINING

TUOMINEN

Brings back memories

Virtual Memory

Physical Memory



Page File

TESO

KO

TRAINING

TUOMINEN

Virtual Memory

Physical Memory

TESO

KO

TRAINING

TUOMINEN

Virtual Memory

- Page 1
- Page 2
- Page 3
- Page n

Physical Memory

- Frame 1
- Frame 2
- Frame 3
- Frame n

KO

TRAINING

Virtual Memory

- Page 1
- Page 2
- Page 3
- Page n

Physical Memory

- Frame 1
- Frame 2
- Frame 3
- Frame n

Virtual Memory

- Page 1
- Page 2
- Page 3
- Page n

MMU

Physical Memory

- Frame 1
- Frame 2
- Frame 3
- Frame n

TESO

KO

TRAINING

TUOMINEN

Virtual Memory

Physical Memory

Page 1

Page 2

Page 3

Page n

MMU

Frame 1

Frame 2

Frame 3

Frame n

Consider Pages and Frames like another unit of memory, it will be easier

TESO

KO

TRAINING

TUOMINEN

Virtual Memory, in Windows ,is actually a polymorphic term.

- VM = the collection of Pages (4KB segments) scattered in memory of a process working set

Virtual Memory

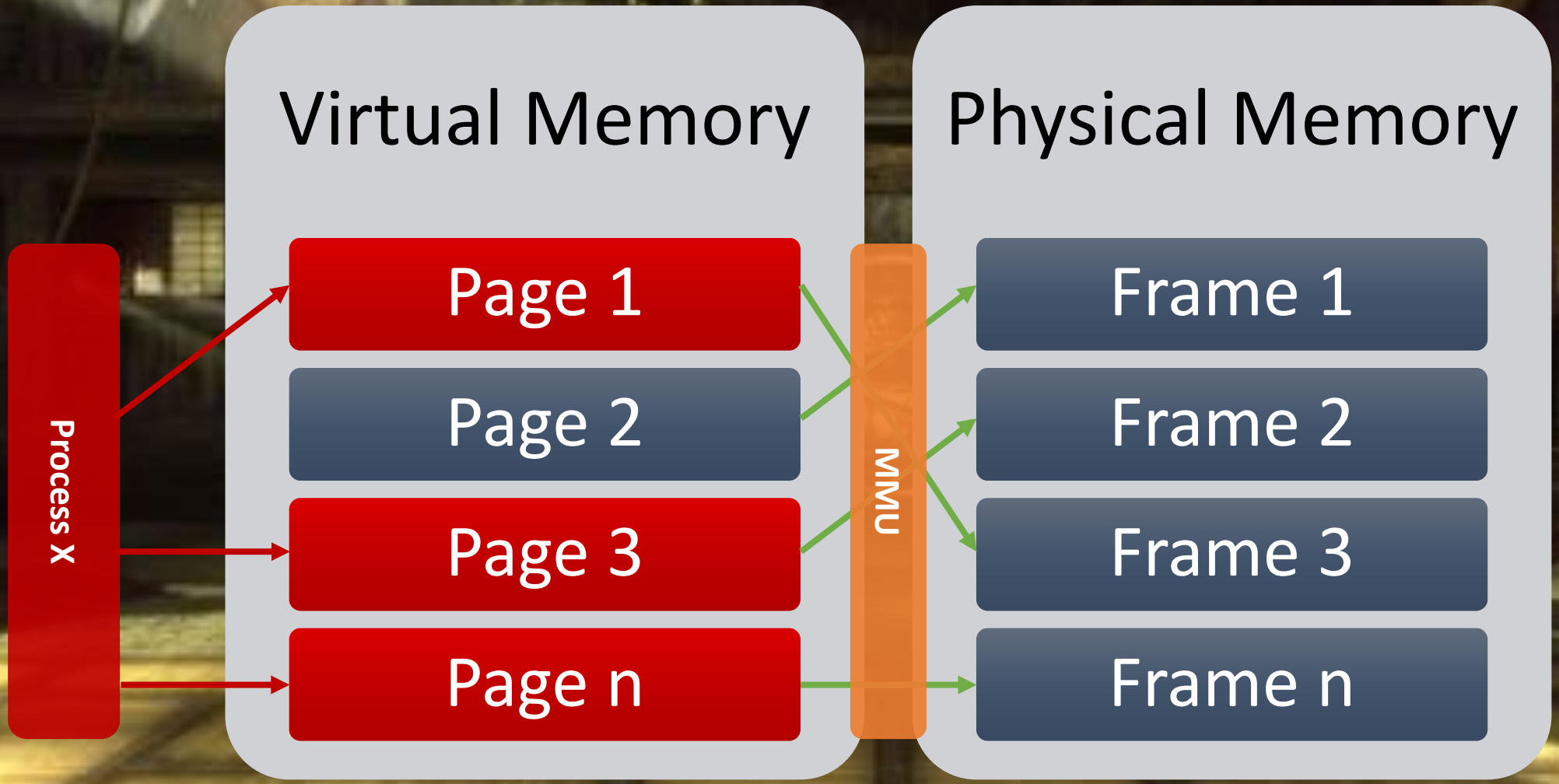
- Page 1
- Page 2
- Page 3
- Page n

MMU

Physical Memory

- Frame 1
- Frame 2
- Frame 3
- Frame n


TESO
KO
TRAINING
TUOMINEN

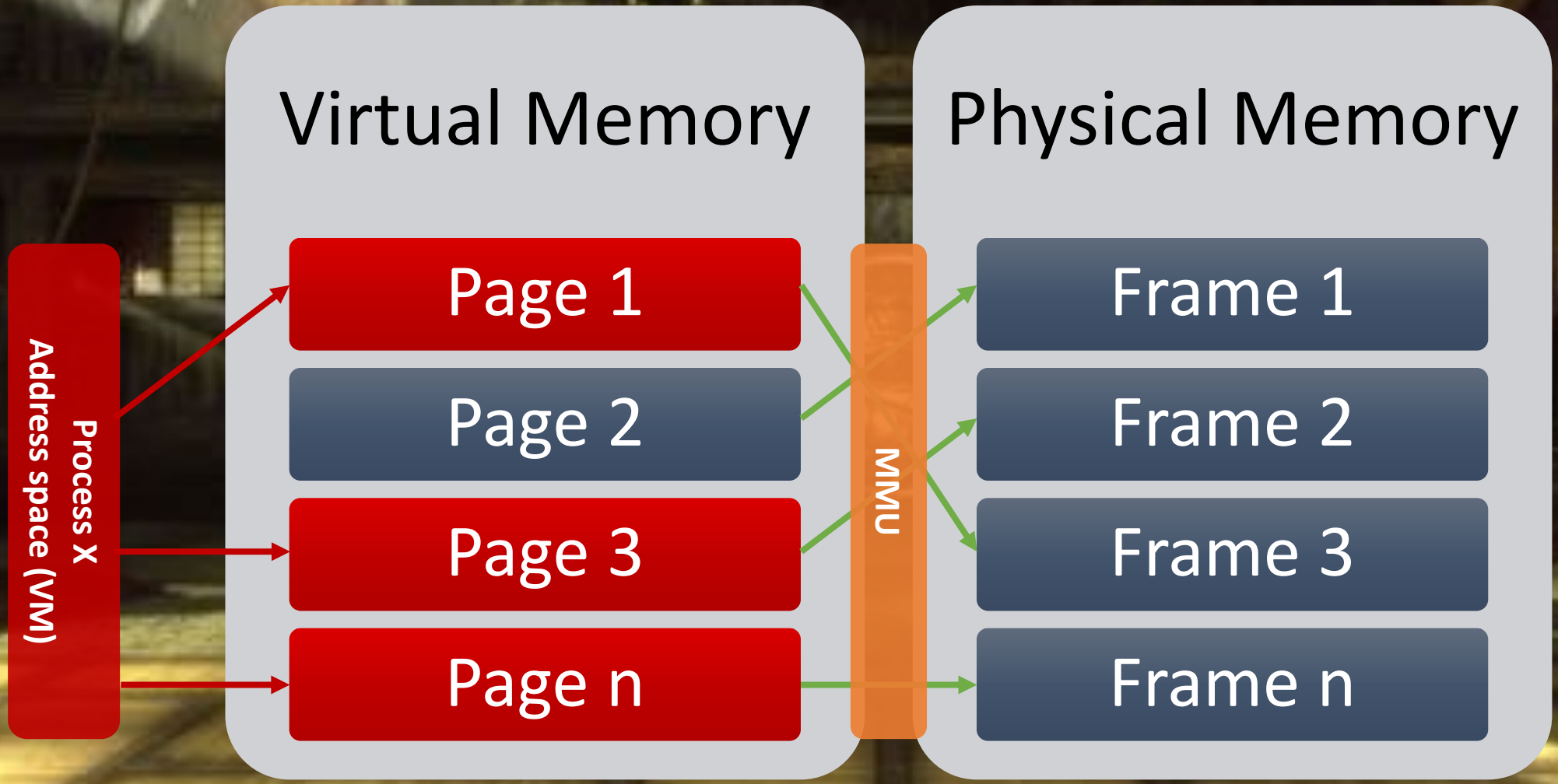


Virtual Memory
Physical Memory
Process X
Page 1
Page 2
Page 3
Page n
MMU
Frame 1
Frame 2
Frame 3
Frame n

TESO

KO

TRAINING

TUOMINEN

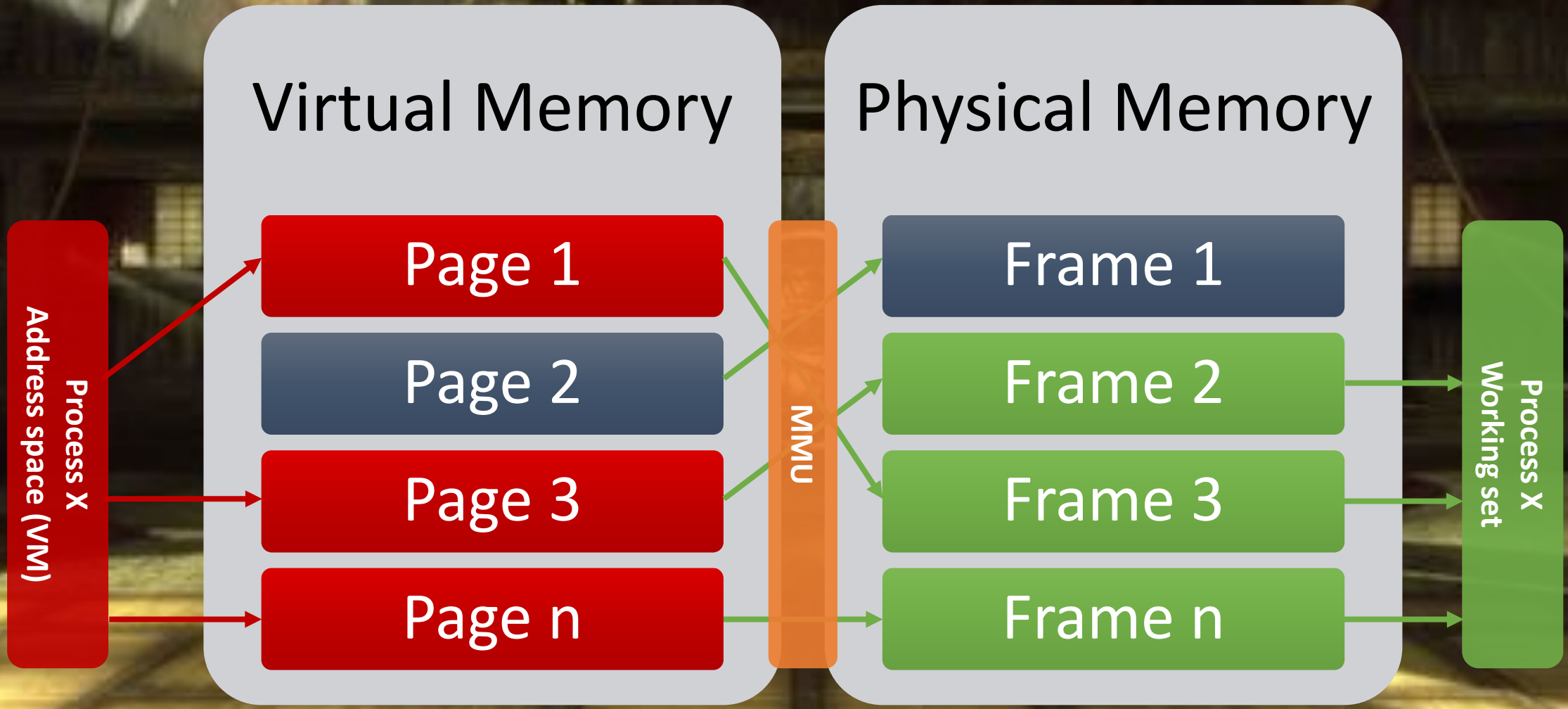



TESO

KO

TRAINING


TUOMINEN

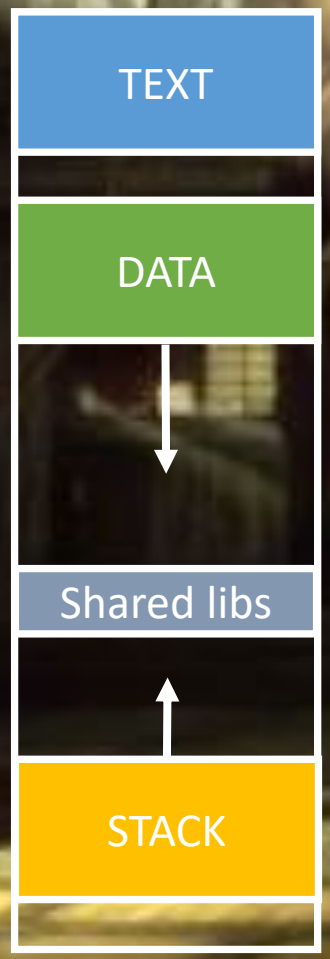

Virtual Memory
Physical Memory
Process X
Address space (VM)
Page 1
Page 2
Page 3
Page n
MMU
Frame 1
Frame 2
Frame 3
Frame n
Process X
Working set

TESO

KO

TRAINING

TUOMINEN



=

Process X
Address space (VM)

Virtual Memory

Page 1
Page 2
Page 3
Page n

MMU

Physical Memory

Frame 1
Frame 2
Frame 3
Frame n

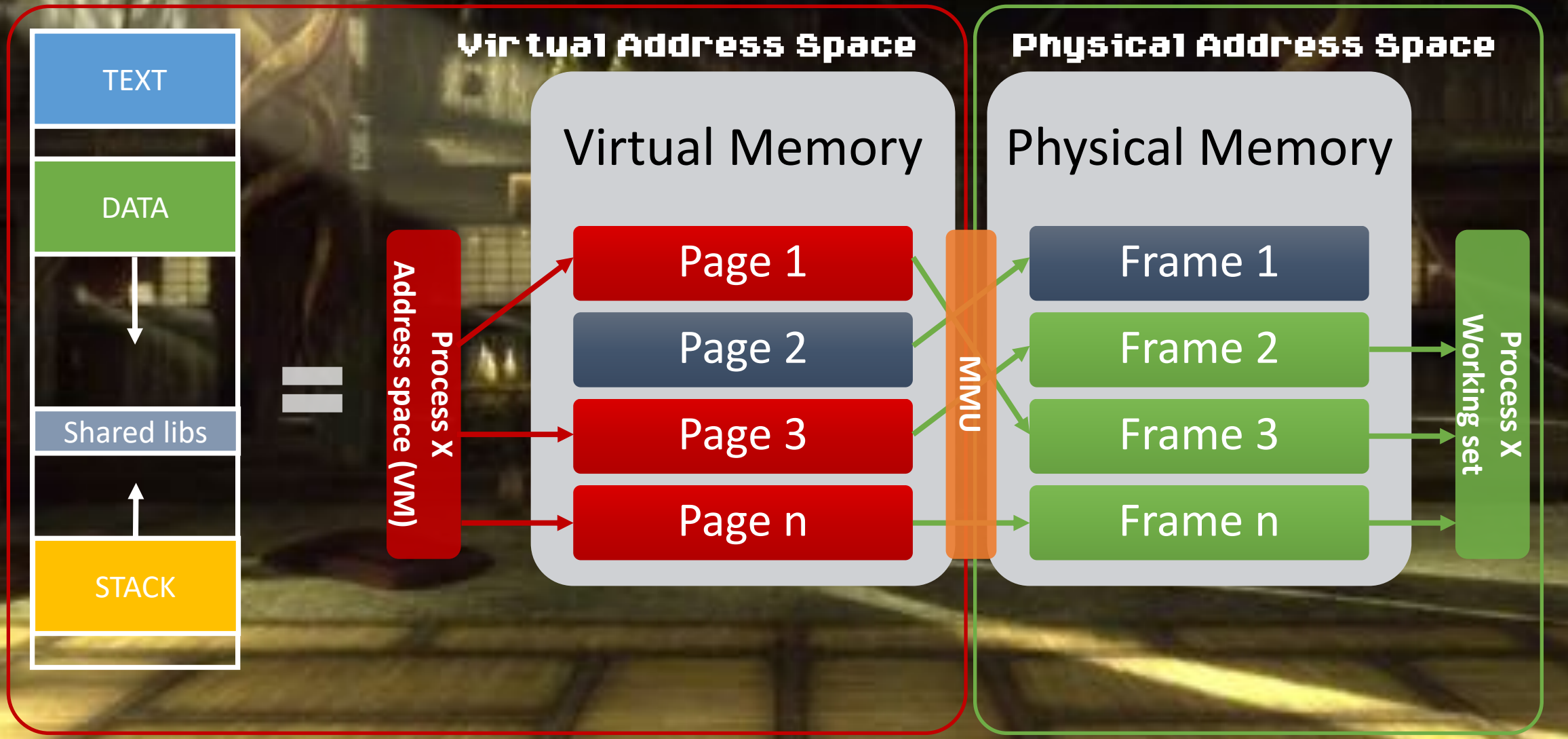
Process X
Working set

TESO

KO

TRAINING

TUOMINEN

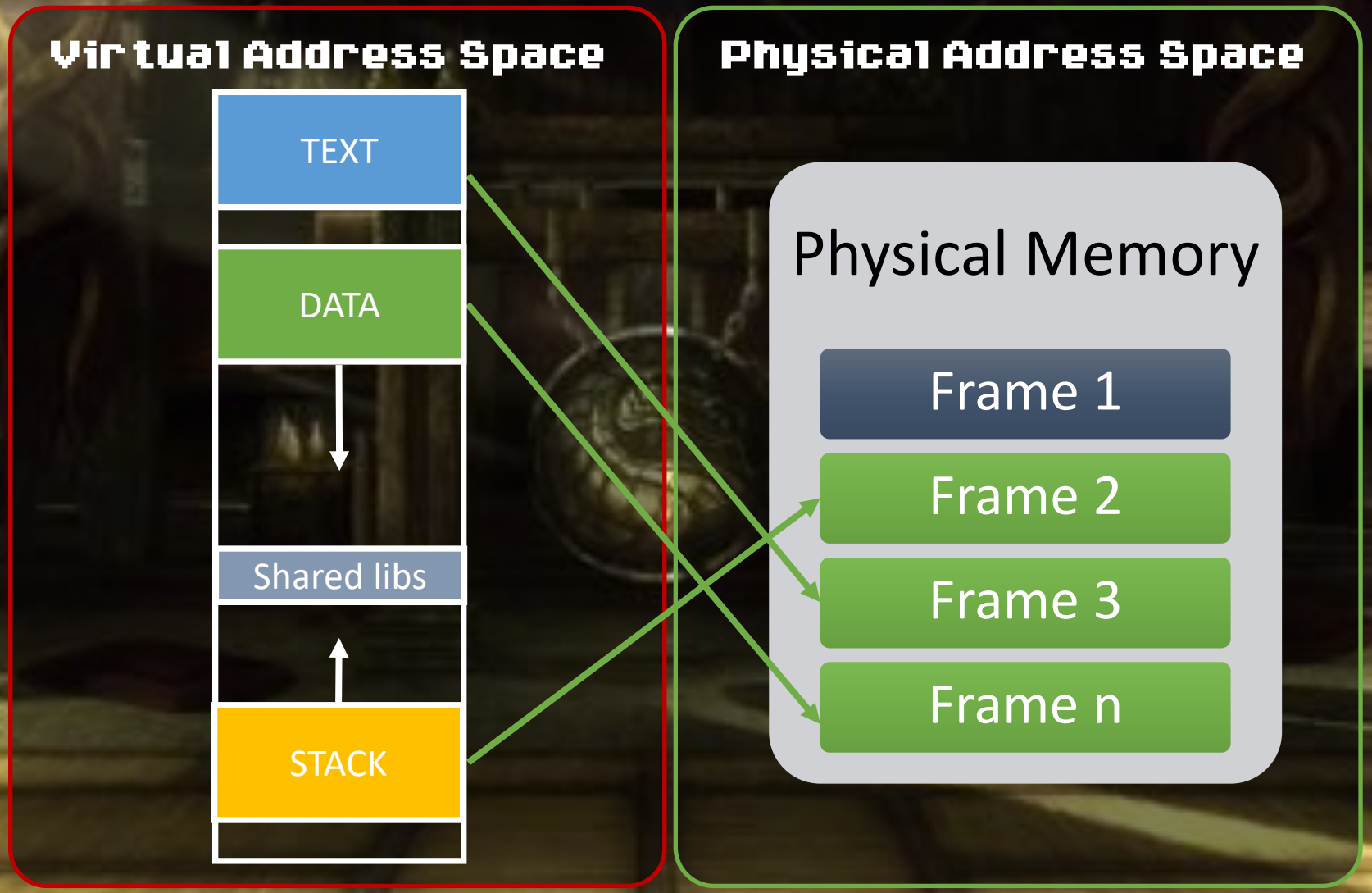


TESO

KO

TRAINING

TUOMINEN



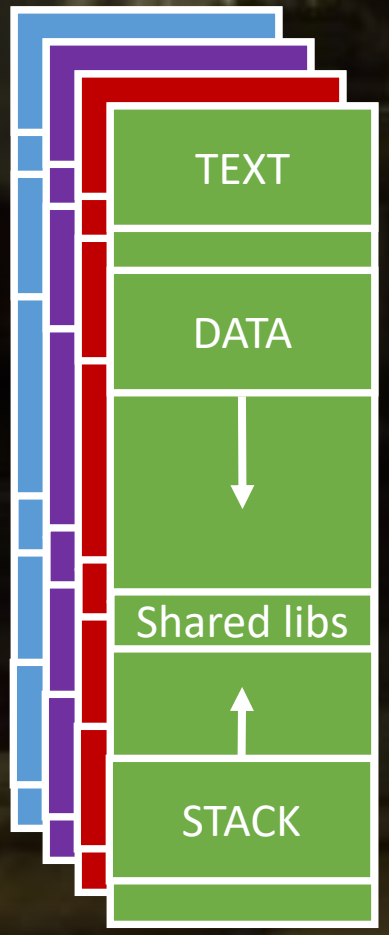
TESO

KO

TRAINING

TUOMINEN

Virtual Address Space



Physical Address Space

Physical Memory

- Frame 1
- Frame 2
- Frame 3
- Frame 4
- Frame 5
- Frame 6
- Frame n

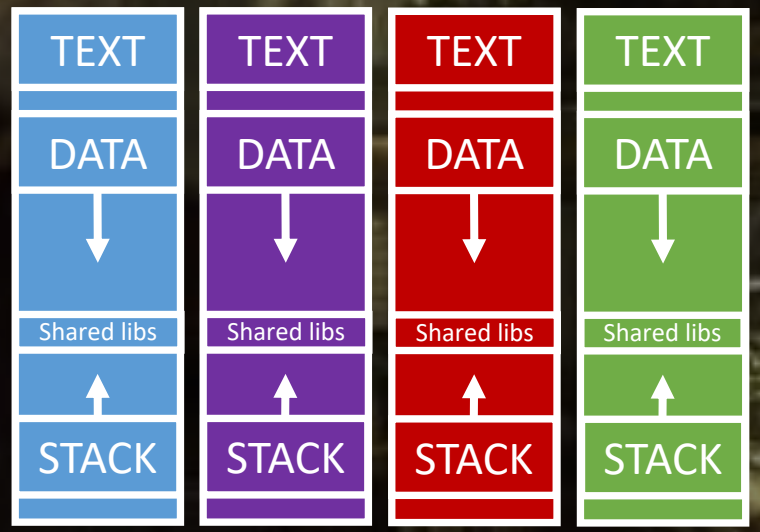
TESO

KO

TRAINING

TUOMINEN

User Space (2Gb)



Kernel Space (2Gb)

Physical Address Space

Physical Memory

- Frame 1
- Frame 2
- Frame 3
- Frame 4
- Frame 5
- ...
- Frame n

TESO

KO

TRAINING

TUOMINEN

WAIT! Where are my files? And registry? And network? And...?

TESO

KO

TRAINING

TUOMINEN

Of course, where you put them...

In the Hard Drive!

TESO

KO

TRAINING

TUOMINEN

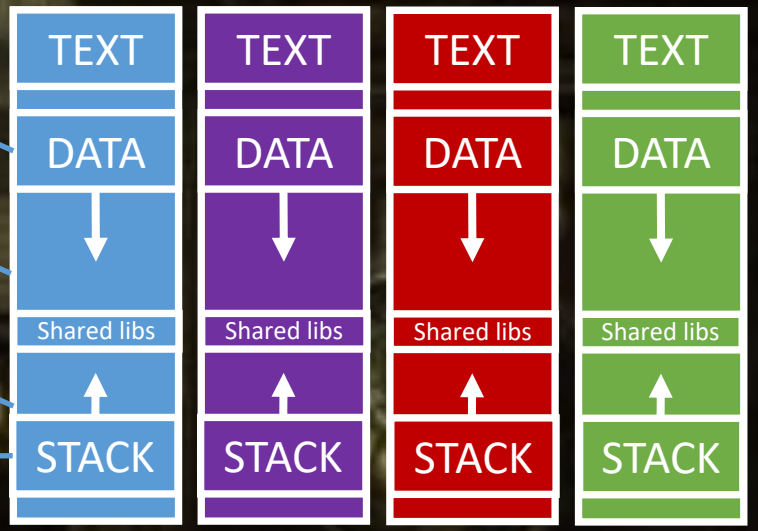
Documents

Registry

Network

[Resources]

User Space (2Gb)



Kernel Space (2Gb)

Physical Address Space

Physical Memory

Frame 1

Frame 2

Frame 3

Frame 4

Frame 5

...

Frame n

TESO

KO

TRAINING

TUOMINEN

MEMORY - FORENSICS

TESO

KO

TRAINING

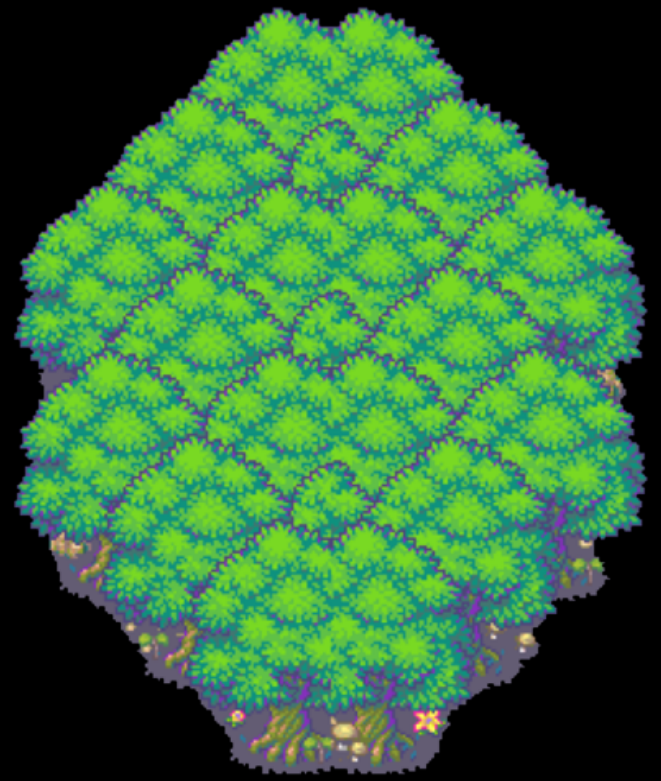
TUOMINEN

TESO

KO

TRAINING

TUOMINEN

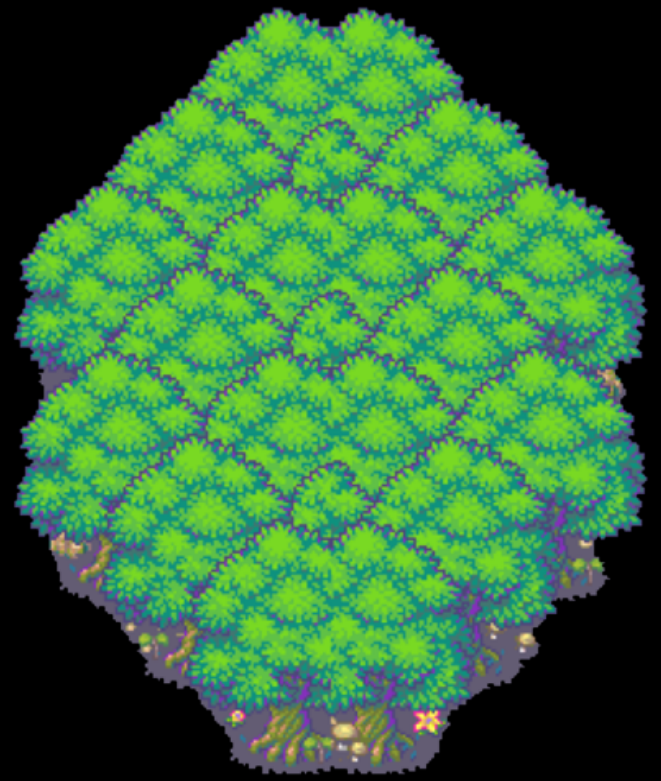


TESO

KO

TRAINING

TUOMINEN




GAME OVER
THANK YOU FOR PLAYING!

MAIN MENU

STORY MODE

TRAINING MODE

VS MODE

GOD MODE

TESO

TUOMINEN

TESO

VS

TUOMINEN

TESO

KO

VS

TUOMINEN

The Contenders

- 2 representatives:
  - Spanish team: Offensive
  - Finnish team: Defensive

TESO

KO

VS

TUOMINEN

Mad? LOL

The GOAL

SPA: Avoid implant detection by FI team

FI: Detect SPA implant with mad memsics skills

TESO

KO

VS

TUOMINEN

The TARGET

TESO

KO

VS

TUOMINEN

The TARGET

TESO

KO

VS

TUOMINEN

The REFEREE

"You're absolutely one brilliant lunatic :D"

TESO

KO

VS

TUOMINEN

The RULES

TESO

KO

VS

TUOMINEN

YOU MUST DEFEAT MY DRAGON
PUNCH TO STAND A CHANCE!

TESO

KO

VS

TUOMINEN

ATTACK ME IF YOU DARE,
I WILL CRUSH YOU.

MAIN MENU

STORY MODE

TRAINING MODE

VS MODE

GOD MODE

TESO

TUOMINEN

TESO

KO

GOD MODE

TUOMINEN

TESO

KO

GOD MODE

TUOMINEN

The Requirements

- No deep "OS XYZ" memory skills
- No deep memsics skills
- Multiplatform — 1 solution to rule them all

TESO

KO

GOD MODE

TUOMINEN

The approach

- Avoid presence detection...?
- Avoid acquisition...?
- Avoid analysis detection...?

TESO

KO

GOD MODE

TUOMINEN

Option 1

TESO

KO

GOD MODE

TUOMINEN

Option 2

TESO

KO

GOD MODE

TUOMINEN

Option 3

TESO

KO

GOD MODE

TUOMINEN

F**k yeah, that's the offensive way!

Option 4

TESO

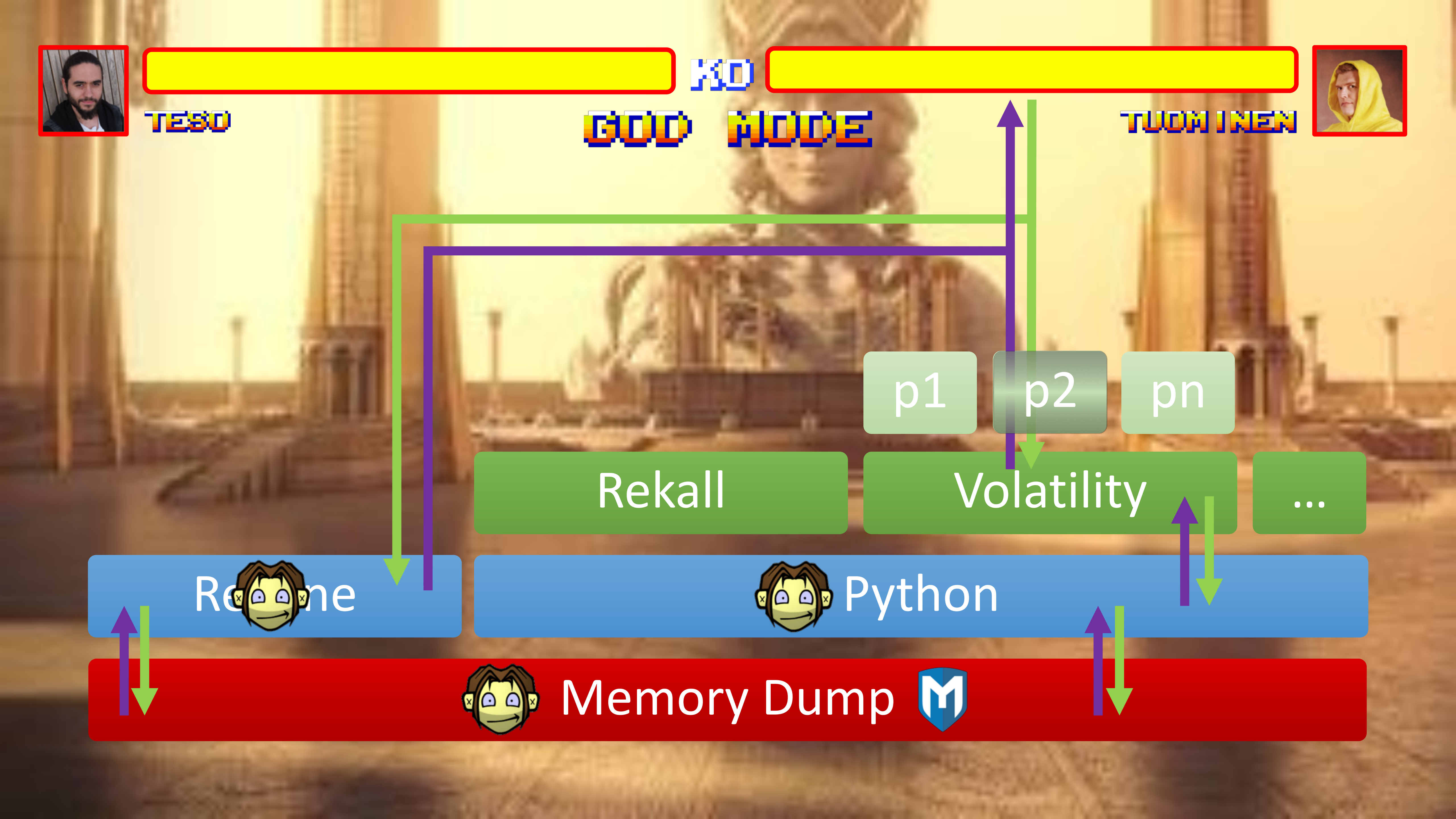
KO

GOD MODE

TUOMINEN

The offensive approach :D

FIGHT BACK


KO
TESO
GOD MODE
TUOMINEN
p1
p2
pn
Rekall
Volatility
...
Python
Memory Dump

TESO

KO

GOD MODE

TUOMINEN

Volatility

TESO

KO

GOD MODE

TUOMINEN

Vulnerabilities

Volatility → Volatility plugin → Python → Python modules → Vulnerability

TESO

KO

GOD MODE

TUOMINEN

The vulnerabilities. Fuzzing?

Human Fuzzing

TESO

KO

GOD MODE

TUOMINEN

Memory Dump

Volatility (Python)

Memory rootkit

Weaponized memory dump

TESO

KO

GOD MODE

TUOMINEN

DLL

TESO

KO

GOD MODE

TUOMINEN

Trigger

DLL

TESO

KO

GOD MODE

TUOMINEN

Trigger

Exploit

DLL

TESO

KO

GOD MODE

TUOMINEN

Trigger

Exploit

Rootkit

DLL

TESO

KO

GOD MODE

TUOMINEN

Trigger

Exploit

Rootkit

Win DLL

Trigger

Exploit

Rootkit

OS X DLL

Trigger

Exploit

Rootkit

Linux DLL

TESO

KO

GOD MODE

TUOMINEN

Trigger

Exploit

Win

OS X

Linux

Rootkit

Win

OS X

Linux

32

DLL

64

TESO

KO

GOD MODE

TUOMINEN

Trigger

Exploit

Win

OS X

Linux

Rootkit

Win

OS X

Linux

32

DLL

64

Volatility

Rekall

Redline

Radare2

EnCase

...

TESO

KO

GOD MODE

TUOMINEN

Approach

Detect Architecture

Detect 32 OS

Detect 64 OS

TESO

KO

GOD MODE

TUOMINEN

Determine architecture

Detect Architecture

\x40\x90

```
* rasm2 -d -b 32 4090
inc eax
nop
```

```
* rasm2 -d -b 64 4090
nop
```

TESO

KO

GOD MODE

TUOMINEN

Determine Architecture

```
arch_detect:
 xor eax, eax
 inc eax
 nop
 jnz x86_code
```

```
x86_code:
 bits 32
 ...
```

```
64_code:
 bits 64
 ...
```

TESO

KO

GOD MODE

TUOMINEN

Determine OS

```
arch_detect:
 xorl %eax, %eax
 rex
 nop
 jnz determine_32_os
```

```
determine_32_os:
 mov eax, fs
 test eax, eax
 jz lin32_code
```

```
determine_64_os:
 mov eax, ds
 test  eax, eax
 jnz win64_code
 jmp lin64_code
```

TESO

KO

GOD MODE

TUOMINEN

Disassembly

\x31\xc0\x40\x90\x75\x08\x8c\xd8\x85\xc0\x75\x0a\xeb\x07\x8c\xe0\x85\xc0\x74\x03\x90\x90\x90\x90

```
[0x00000000]> e asm.bits
64
[0x00000000]> pdf
┌ (fcn) fcn.00000000 24
│   fcn.00000000 ();
│           0x00000000      31c0           xor eax, eax
│           0x00000002      4090           nop
│       ┌─< 0x00000004      7508           jne 0xe
│       │   0x00000006      8cd8           mov eax, ds
│       │   0x00000008      85c0           test eax, eax
│     ┌─< 0x0000000a      750a           jne 0x16
│     │ ┌─< 0x0000000c      eb07           jmp 0x15
│     │ │   0x0000000e      8ce0           mov eax, fs
│     │ │   0x00000010      85c0           test eax, eax
│     │ ┌─< 0x00000012      7403           je 0x17
│     │ │   0x00000014      90             nop
│     │ │       ; JMP XREF from 0x0000000c (fcn.00000000)
│     └─└─> 0x00000015      90             nop
│       └─> 0x00000016      90             nop
└         └─> 0x00000017      90             nop
```

```
[0x00000000]> e asm.bits
32
[0x00000000]> pdf
┌ (fcn) fcn.00000000 (64 bits) 24
│   fcn.00000000 ();
│           0x00000000      31c0           xor eax, eax
│           0x00000002      40             inc eax
│           0x00000003      90             nop
│       ┌─< 0x00000004      7508           jne 0xe
│       │   0x00000006      8cd8           mov eax, ds
│       │   0x00000008      85c0           test eax, eax
│     ┌─< 0x0000000a      750a           jne 0x16
│     │ ┌─< 0x0000000c      eb07           jmp 0x15
│     │ │   0x0000000e      8ce0           mov eax, fs
│     │ │   0x00000010      85c0           test eax, eax
│     │ ┌─< 0x00000012      7403           je 0x17
│     │ │   0x00000014      90             nop
│     │ │       ; JMP XREF from 0x0000000c (fcn.00000000)
│     └─└─> 0x00000015      90             nop
│       └─> 0x00000016      90             nop
└         └─> 0x00000017      90             nop
```

TESO

KO

GOD MODE

TUOMINEN

But in real world...

ASLR/PIE

...

TESO

KO

GOD MODE

TUOMINEN

But in real world...

ROP gadgets!

TESO

KO

GOD MODE

TUOMINEN

So, for real world...

I need help!

TESO

KO

GOD MODE

TUOMINEN

Assuming...

OS X

Vulnerable Buffer

Saved RIP

Windows

Vulnerable Buffer

Saved RIP

Linux

Vulnerable Buffer

Saved RIP

TESO

KO

GOD MODE

TUOMINEN

Then...

OS X

Vulnerable Buffer

ROP Chain

Linux

Vulnerable Buffer

ROP Chain

Windows

Vulnerable Buffer

ROP Chain

TESO

KO

GOD MODE

TUOMINEN

Finally...

OFFSET 2

Vulnerable Buffer	Linux: add rsp, OFF1 ret	OS X: add rsp, OFF2 ret	ROP Chain Windows	ROP Chain Linux	ROP Chain OS X

OFFSET 1

TESO

KO

GOD MODE

TUOMINEN

And what now?

Post-exploitation time!

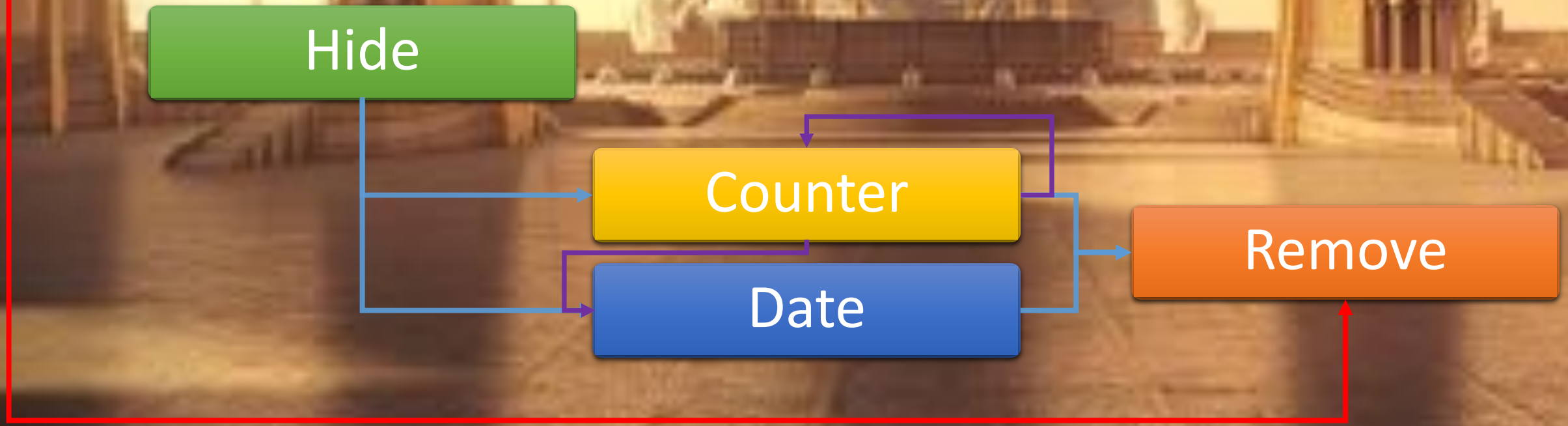
TESO

KO

GOD MODE

TUOMINEN

Post-exploitation time!




So Long, and Thanks for All the Fish
Hugo Teso (@hteso)