Windows Memory Forensics: Identification of (Malicious) Modifications in Memory-Mapped Image Files

Frank Block (a,b)

a: ERNW Research GmbH, Heidelberg, Germany

b: Friedrich-Alexander University Erlangen-Nuremberg (FAU), Germany

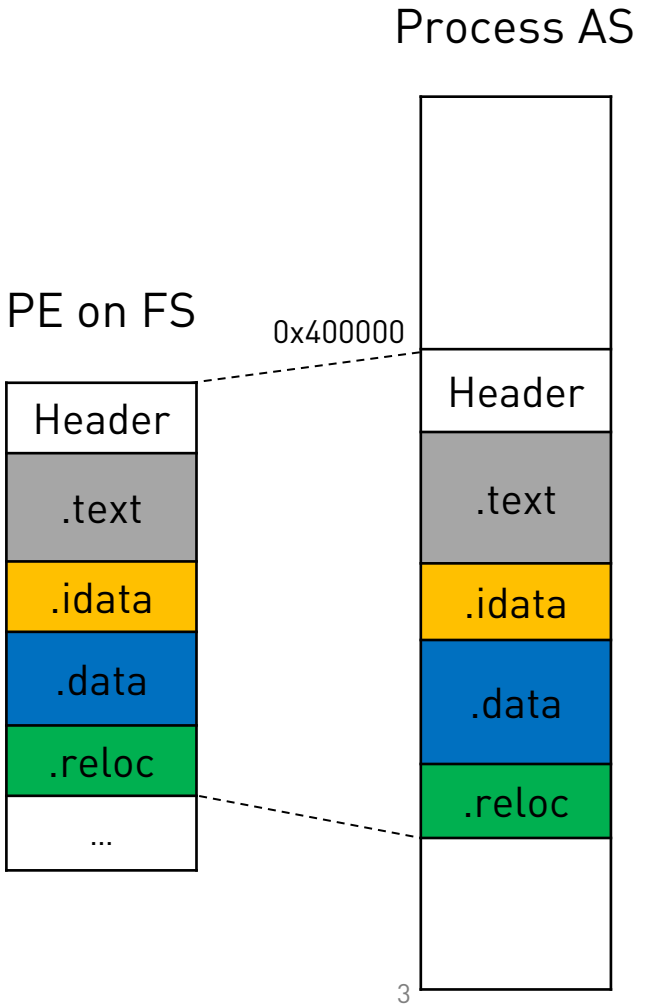
Agenda

- **Introduction**
- Detecting Modified Pages
- Identifying Malicious Modifications
- Evaluation
- Conclusion

ERNW RESEARCH
pursuing knowledge.

Memory-Mapped Image Files

- A Memory-Mapped Image File (**MMIF**) is in essence a Portable Executable, which gets mapped from the file system into the memory space.
  - Mainly DLLs and the main executable.
- Our Focus: Executable memory

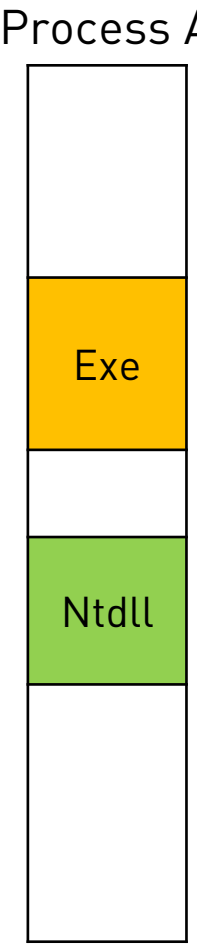


ERNW RESEARCH pursuing knowledge.

Memory-Mapped Image Files

- Attackers may manipulate MMIFs in order to:
  - Spy on users/processes: **Hooking**
  - Disable OS or third-party protection/detection mechanisms: e.g., **AMSI/ETW Patches**
  - Hide their presence by manipulating API arguments/return values: **Hooking**
  - Hide their malicious code: e.g., **Module Stomping, Process Hollowing**

ERNW RESEARCH
pursuing knowledge.



ERNW RESEARCH
pursuing knowledge.

Process A

ERNW RESEARCH
pursuing knowledge.

```
...
1f2f d9c4 722b 1023 3693 35a9 6cb6 e007 07c2 6f00 defa b928 1c6a 5039 3bda 76d8
9706 c787 66b1 ae86 e3bc 49ee 6e84 1094 c9dd d92c a988 4f83 62d9 2a9b 10bd 4be9
33e6 bb67 d1c1 3534 f7db 7460 eacd 7e7f 52ae 639f 9d35 aab7 b11e 983f 05cc 7b69
10c9 9f70 a0de 7759 cfa4 1757 b42f c1e8 2382 c145 177a bfd2 3dd2 bdab 437d 7608
588a bd02 e5b5 dd49 3dfd 20b4 17d9 54b1 27c7 b115 7b9b 990a e5c5 2c3e fea0 80ad
c11a fe2a 2a4e bbe8 2b72 7dc7 d79c 1284 86f8 436d 935b 3520 da63 bb39 4789 fa32
989d 03bc fe90 c27d 09d3 ab95 8b98 f23e f8e9 26d5 768c 4bfe 2917 33bd bc0a af7d
aa8f a4b7 a65d 3020 4e4b 5709 9d7e 071b 420a ff12 0ea8 0d67 ef90 4145 bf46 d713
974c 62b6 8882 03a2 38ef b57c e6b6 ab01 925f d1a9 858e 6cdb d436 7650 8708 e369
9e91 f6f7 f370 c2e3 bc49 5a0c 2039 1243 c1ed 3ab2 fadc bd70 7f90 a9d4 9876 6019
74c8 e999 2638 257a 47b4 5641 8239 e575 6311 f4bf f34e 2014 e18f 5adf 2312 8ddd
2f52 2914 d443 0343 6b29 a402 e84e 7e88 c4e9 9866 48f2 21f9 1f48 ca17 862c 0545
07c8 dc12 b39e 6237 358d bede 64f2 5278 7d1f a907 bd10 2f8c b2ca 7cec a690 28b8
f376 40a8 36e7 922e 336f 09f5 6348 3fcd eed4 3f1c 2d09 2205 2ac5 ebad 1bfb 6c86
d58c 66cf 3a95 95b5 b35e 4818 fa49 fdb5 2d0c bee7 7819 0f83 30e3 1cec 28af ddf2
dd2b 6ba2 4505 ec4d 9ab6 ca20 c9ff 0937 deb6 9b24 512a d1c5 8794 c0ef b13e 1654
6ae5 b4e4 8b2f e38a b7a0 5a42 7cef 4935 ba22 e8e9 cbea f151 8710 71c4 3bcd 0b86
51f7 c552 0e6f b807 51fa df2a f3aa 529d 2e58 7c1c e342 812a 17a3 20e1 0c2b c10c
9d46 1e20 9fe8 8d40 2efd bb5a b956 7186 ae1c e730 ce02 e492 2164 09b4 b54e c586
f425 ce5e ccd7 d260 f795 87bf c14e 1106 cdcf 9c51 1b39 654d 4320 3383 0875 8170
f58a 5faf 901d d24f cb06 fe2d 497a fd6b b8a8 6937 5b9b e50b cbd1 f4f9 7ece 8930
f678 98b4 e2c3 642e 9f0b cc5c 2155 cca4 f103 66c9 1e63 dba7 94a4 2a8d 8eac b8fb
82e7 6c5f 61eb ab1c 1cc7 5874 a4c9 a71c 881b c1de aa74 d5ca 6c50 f457 5221 2a04
bf32 643c 6c2a 1942 8e27 e8d5 2c88 8255 e84d 1db7 6d79 5b8a 59b2 3c2c 76cc 49ed
67e6 37cb 4f91 2080 05b5 9b5d b9f4 2df2 469c 7b96 11e6 73ce 7be4 2099 1628 fd2e
429d 29b1 d1fa 794f 3cec b825 9f5a 392d ab67 645f 3887 6dec feb4 c86c 40dd 0fe6
bd2c db6c 5fee 34ed c4f0 ddcb d184 80e2 6940 715c f775 b97c 03d1 46f4 7c36 bdb4
c9e1 4231 88f9 9343 7954 2456 56ae 781e ef7b 94df 1bbb 8327 b32e e55f 84a4 b9d5
1ca3 b088 7d50 e993 c813 8ea4 6bf6 c0ee fb60 3ef9 fb03 ac8c db0c 4b52 49d8 5b82
619a 252b 1ce3 ce02 f36d 93f1 3169 29c1 031e 79ab 1b84 cea4 004f 43cd 9b27 56ce
64d8 bab2 2851 8511 d63b e78c 8586 6323 9d7a dad4 fab3 02a8 bfad 8987 2917 bc47
df5a 14f7 98ce 1163 4679 dfd2 51c6 69a9 236e 809b 87b1 81f7 838e f0c1 728b 71c4
22c9 302f f2f6 e379 b3f0 9e7a 1014 cbc0 60c3 6075 2c35 5b48 dc95 a45b 86a1 997c
10d4 4067 fcb0 3651 031e 5b7c 078c 9898 aef9 197f 3559 2623 2c61 fa77 34ff 4caf
155d 53fc 5fe7 9baf 1388 e561 3e21 d0a6 b8e0 e410 a6bc 7e3e 8fe2 be99 b9fb fd92
7640 7dbd 3d47 436a 2a99 6007 e75a 7093 9707 5a7f aee2 9fa4 e1c7 4d62 260c 0f7a
...
```

ERNW RESEARCH
pursuing knowledge.

```
55 48 89 e5 53 56 57 50

0x12340: push    rbp
0x12341: mov     rbp, rsp
0x12344: push    rbx
0x12345: push    rsi
0x12346: push    rdi
0x12347: push    rax
```

```
e9 6b 33 01 00 56 57 50

0x12340: jmp     0x256b0
0x12345: push    rsi
0x12346: push    rdi
0x12347: push    rax
```

ERNW RESEARCH pursuing knowledge.

Goals/Contributions

- Reliable detection of modified MMIF pages.
- Compare it with a ground truth available from memory, in order to pinpoint on the exact modified bytes.
- Identification of benign modifications.
- A publicly available Volatility plugin: *imgmalfind*

- (Focus: Executable memory)

ERNW RESEARCH pursuing knowledge.

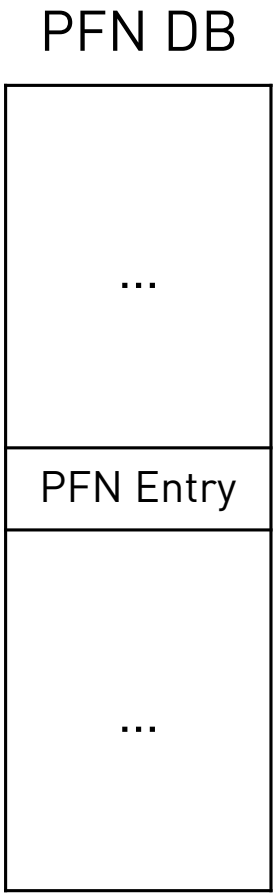
Related Work

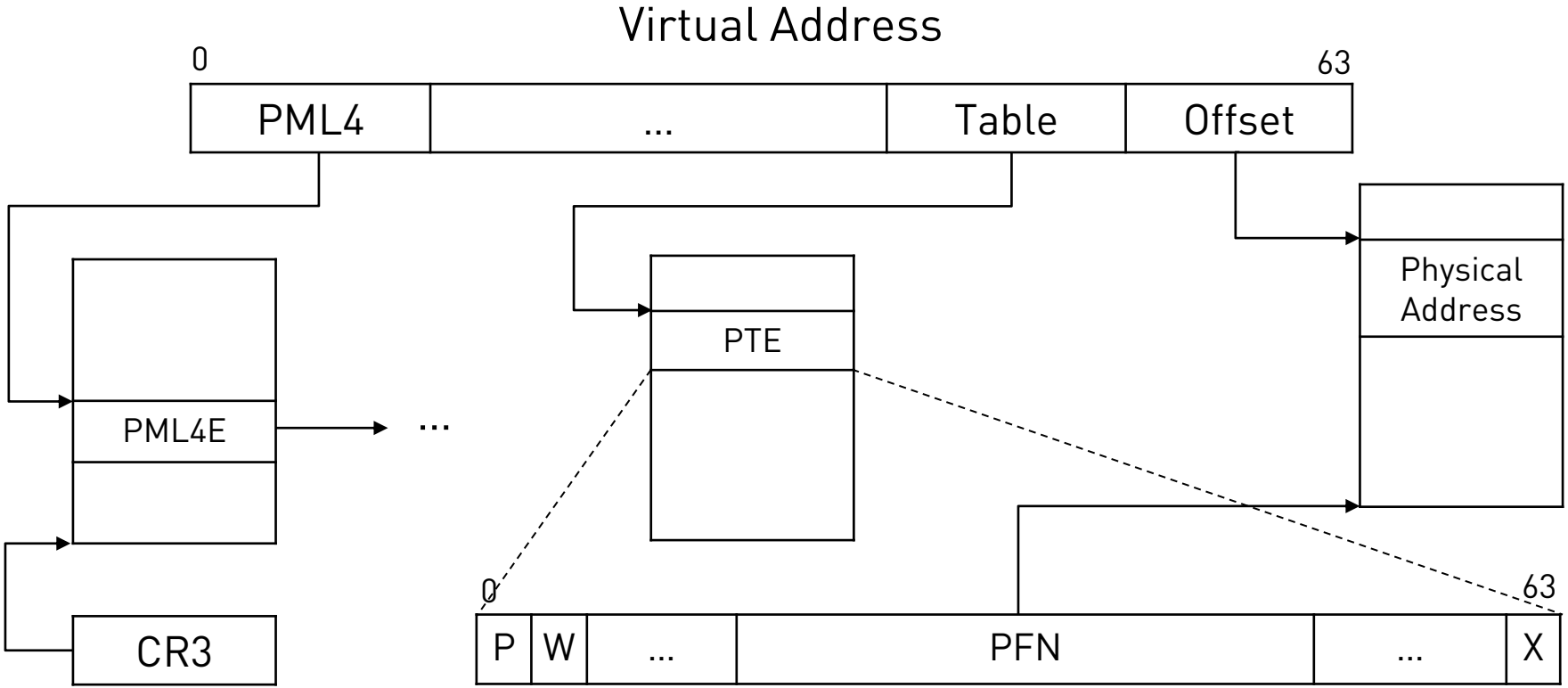
- Detecting modified MMIF memory in general
  - PrototypePte [1]
  - QueryWorkingSetEx, used by Moneta [2]
  - ModifiedList [3]
  - Hashtest [4]
  - hollows_hunter [5]
- Detecting particular attack techniques
  - apihooks, Hooktracer [6]
  - dotnet_amsi [7]

ERNW RESEARCH
pursuing knowledge.

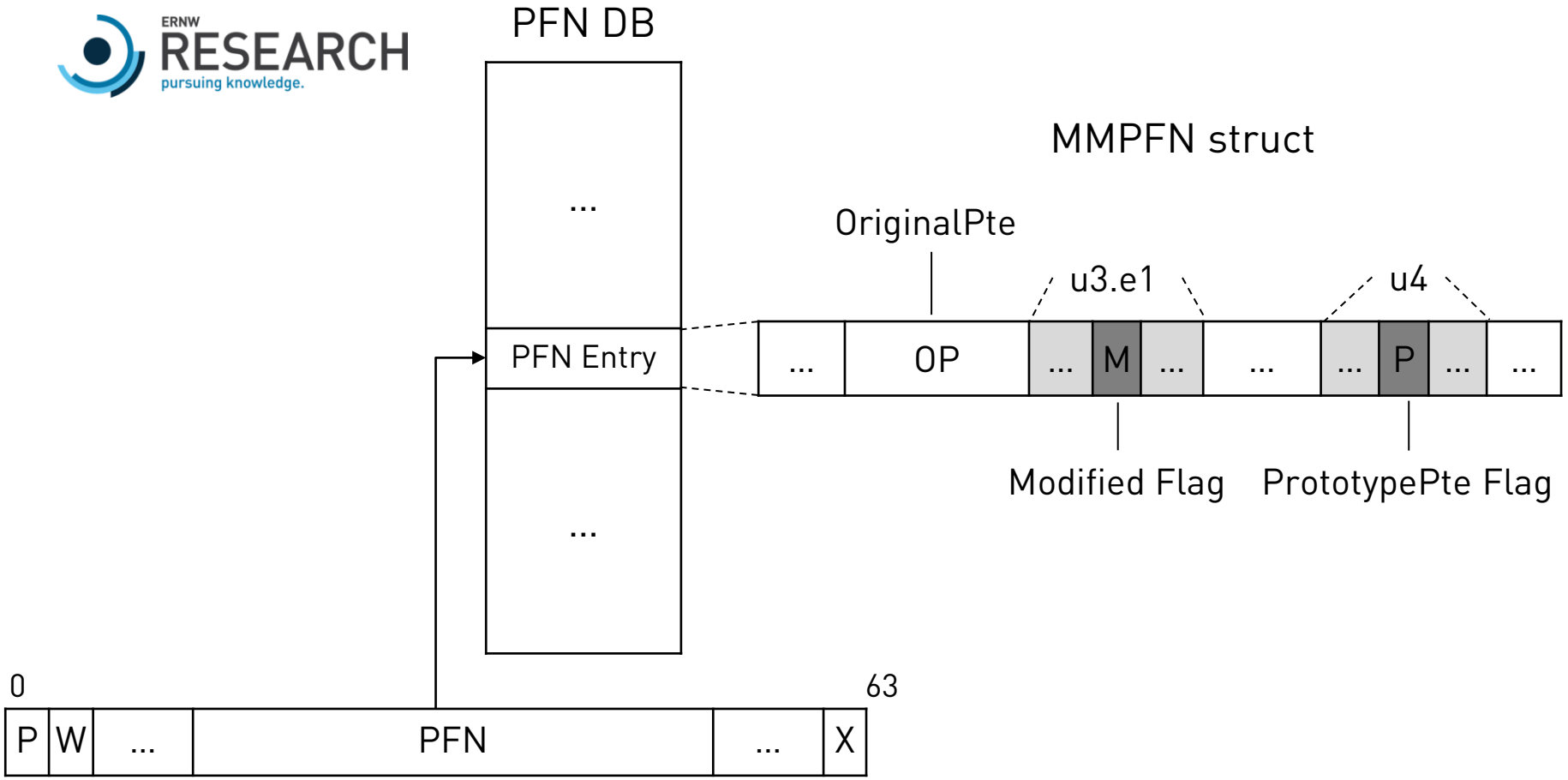
PFN Database

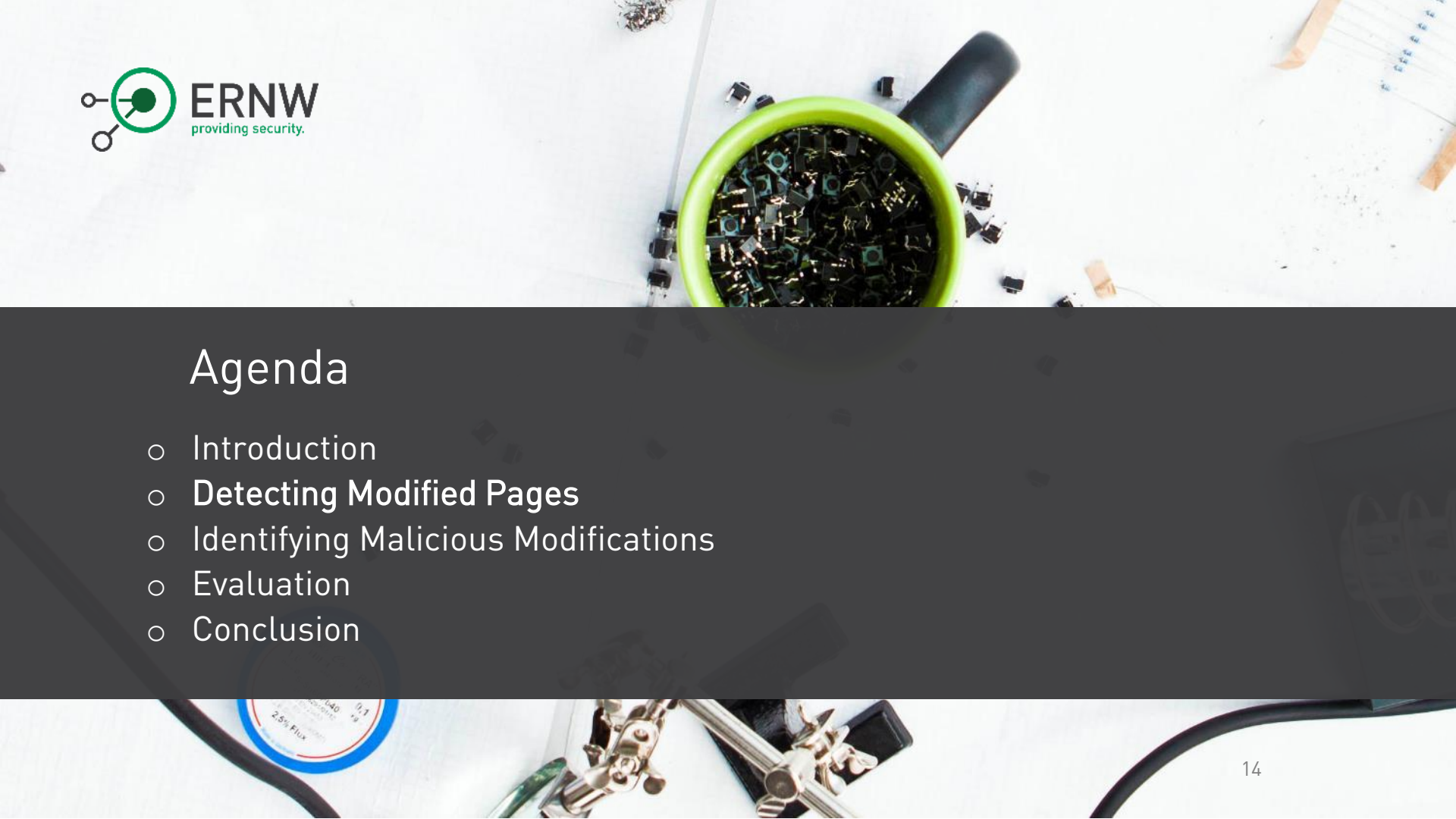
- The PFN Database can be seen as the inventory of all physical pages.
- For each physical memory page, there is one PFN entry.
- Each entry is an instance of the **MMPFN** structure, which describes the physical memory page.





Source: Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A


PFN DB
...
PFN Entry
...
MMPFN struct
OriginalPte
u3.e1
u4
...
OP
...
M
...
...
...
P
...
...
Modified Flag
PrototypePte Flag
0
63
P
W
...
PFN
...
X



Agenda

- Introduction
- **Detecting Modified Pages**
- Identifying Malicious Modifications
- Evaluation
- Conclusion

ERNW RESEARCH pursuing knowledge.

Identification of Modified Image Pages

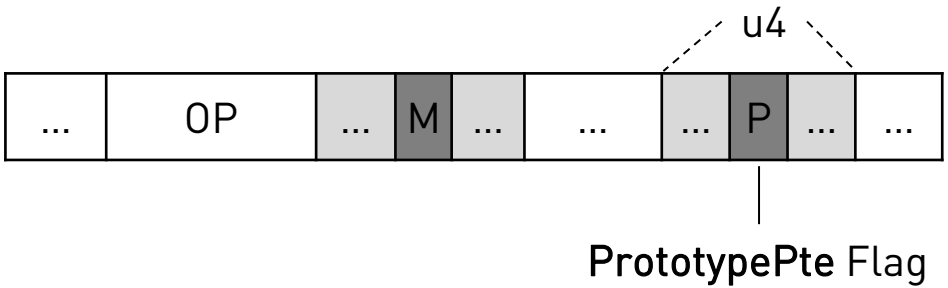
- Currently, these 3 methods are known:
  - PrototypePte [1]
  - QueryWorkingSetEx [2]
  - ModifiedList [3]

- But they all fail in reliably detecting modified pages.

ERNW RESEARCH
pursuing knowledge.

PrototypePte

- Uses the MMPFN's *PrototypePte* field in order to identify modified pages for memory-mapped image files.



ERNW RESEARCH pursuing knowledge.

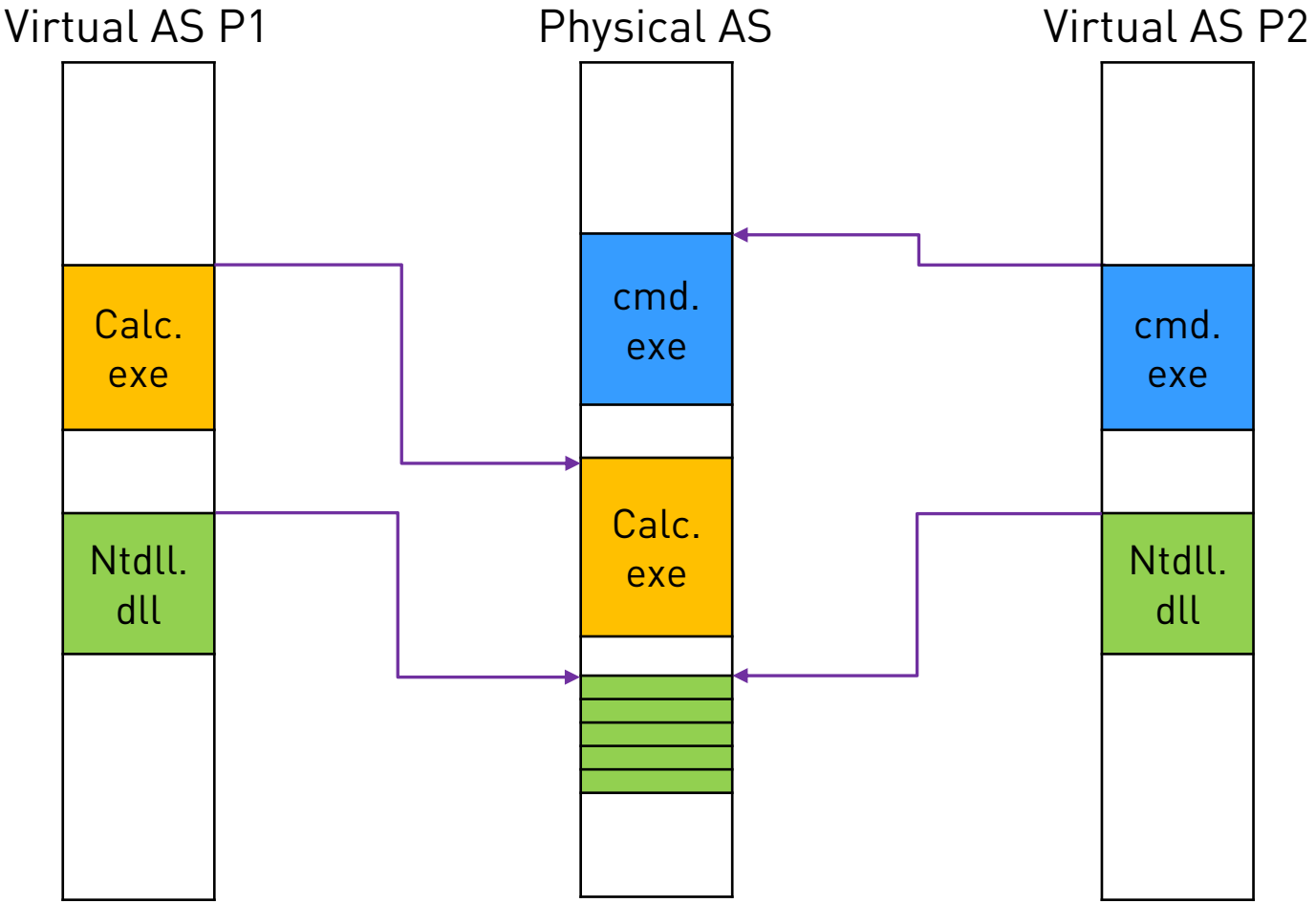
PrototypePte

- This approach can be translated to the question:

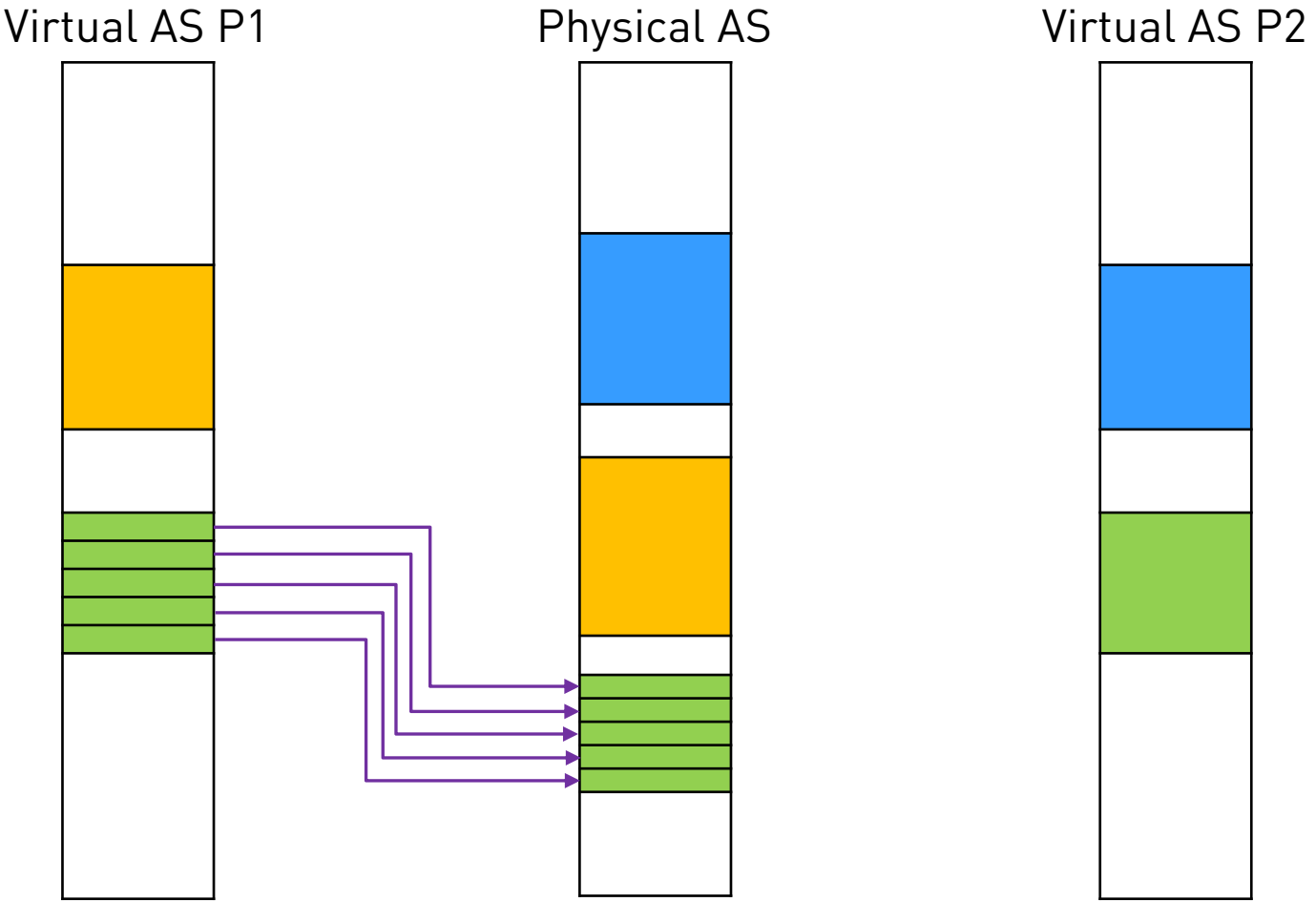
Does the current page belong to a memory-mapped image file and if so, **is it shareable**?

- In the past, the answer to this question was only true for **unmodified** image-file pages.
- The Problem: Along comes **Memory Combining**.

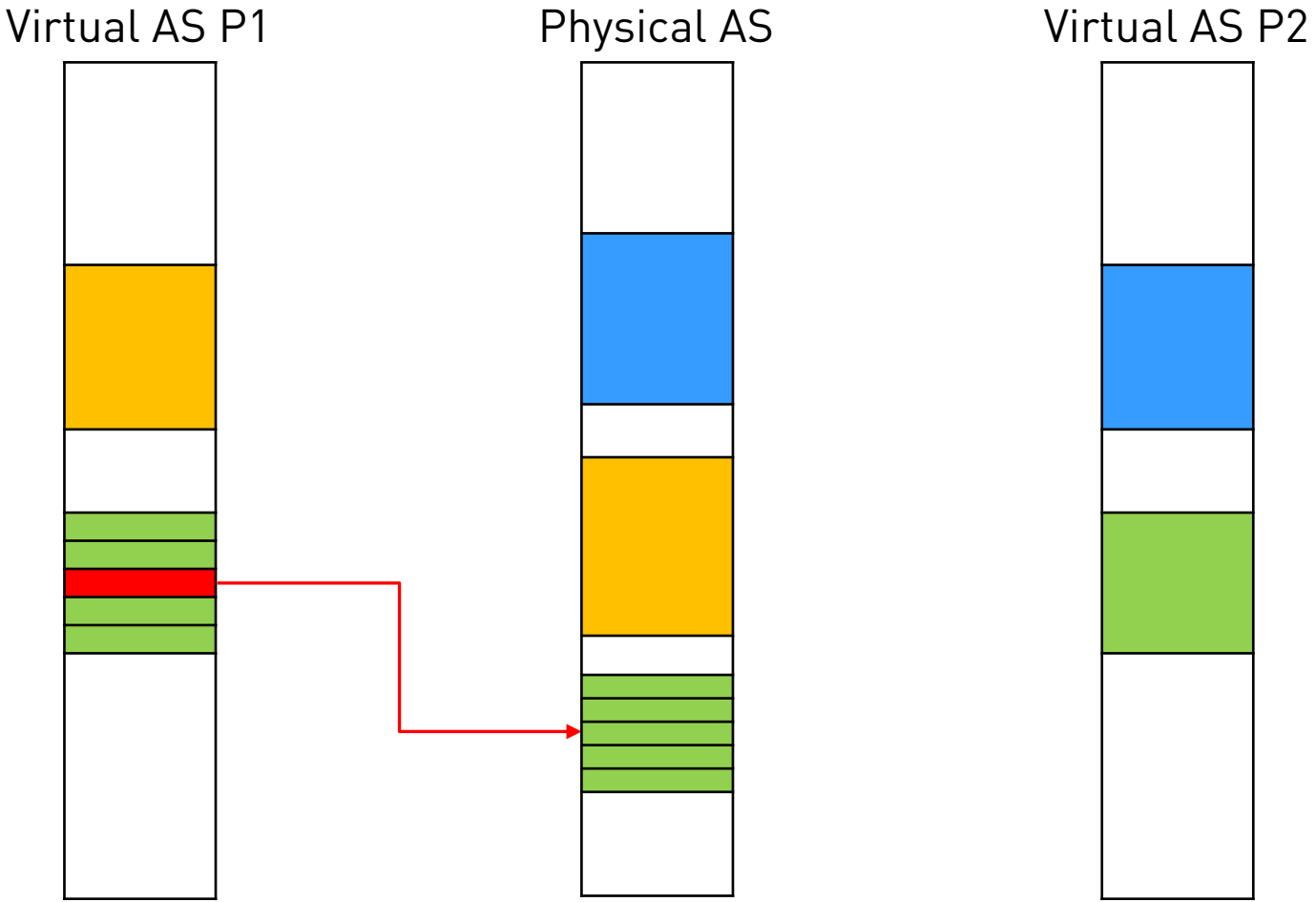
ERNW RESEARCH
pursuing knowledge.



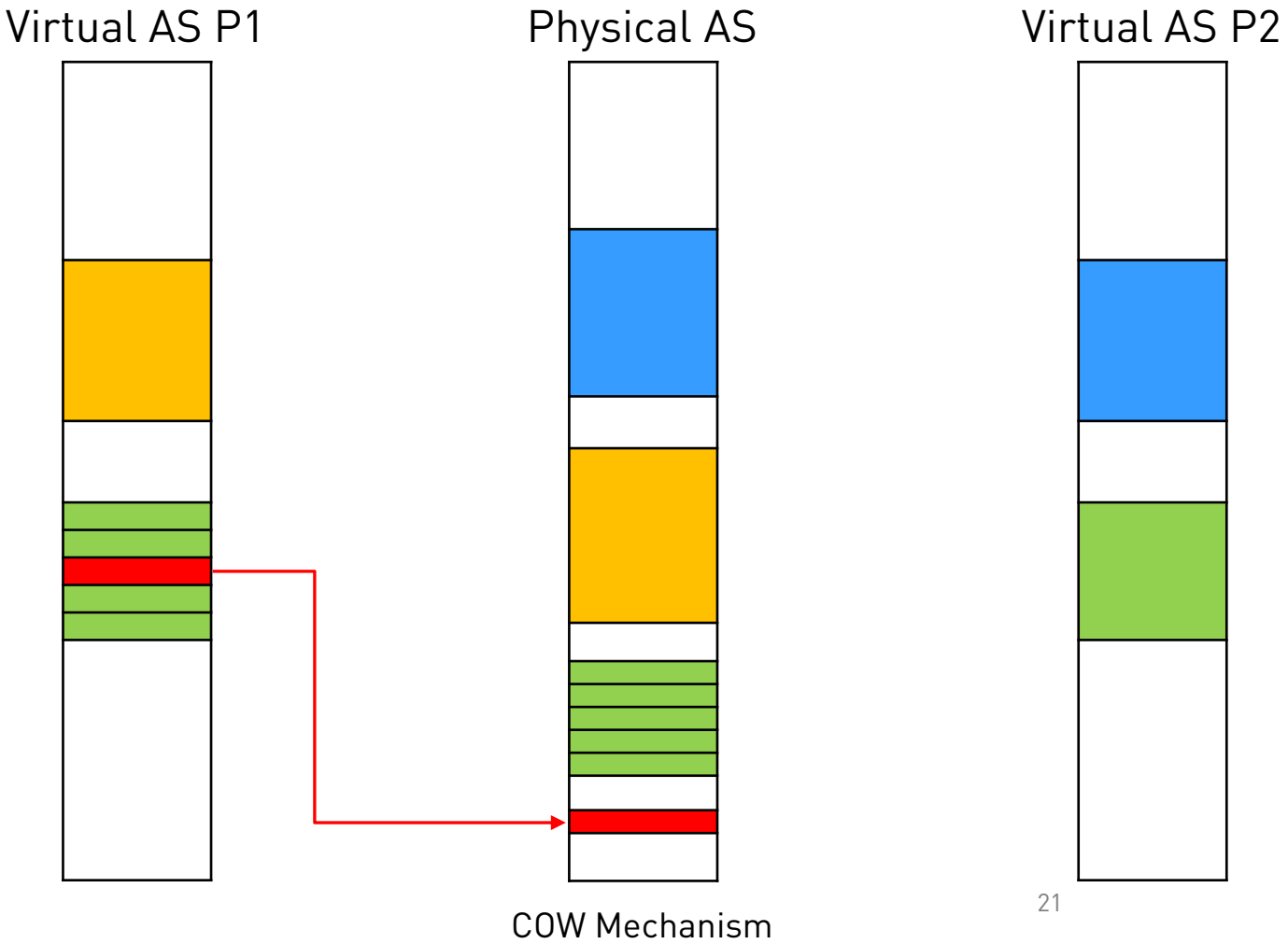
ERNW RESEARCH
pursuing knowledge.



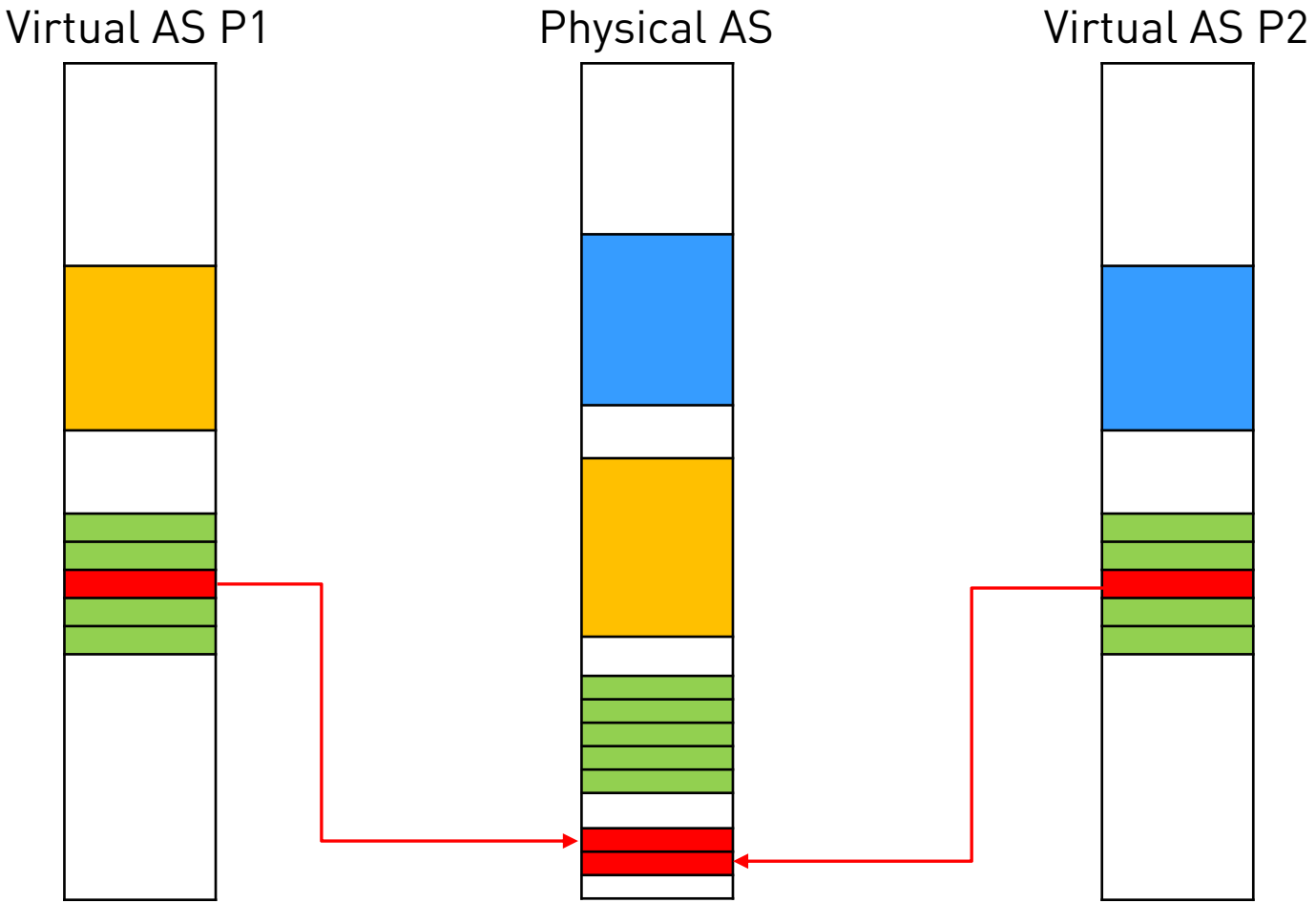
ERNW RESEARCH
pursuing knowledge.



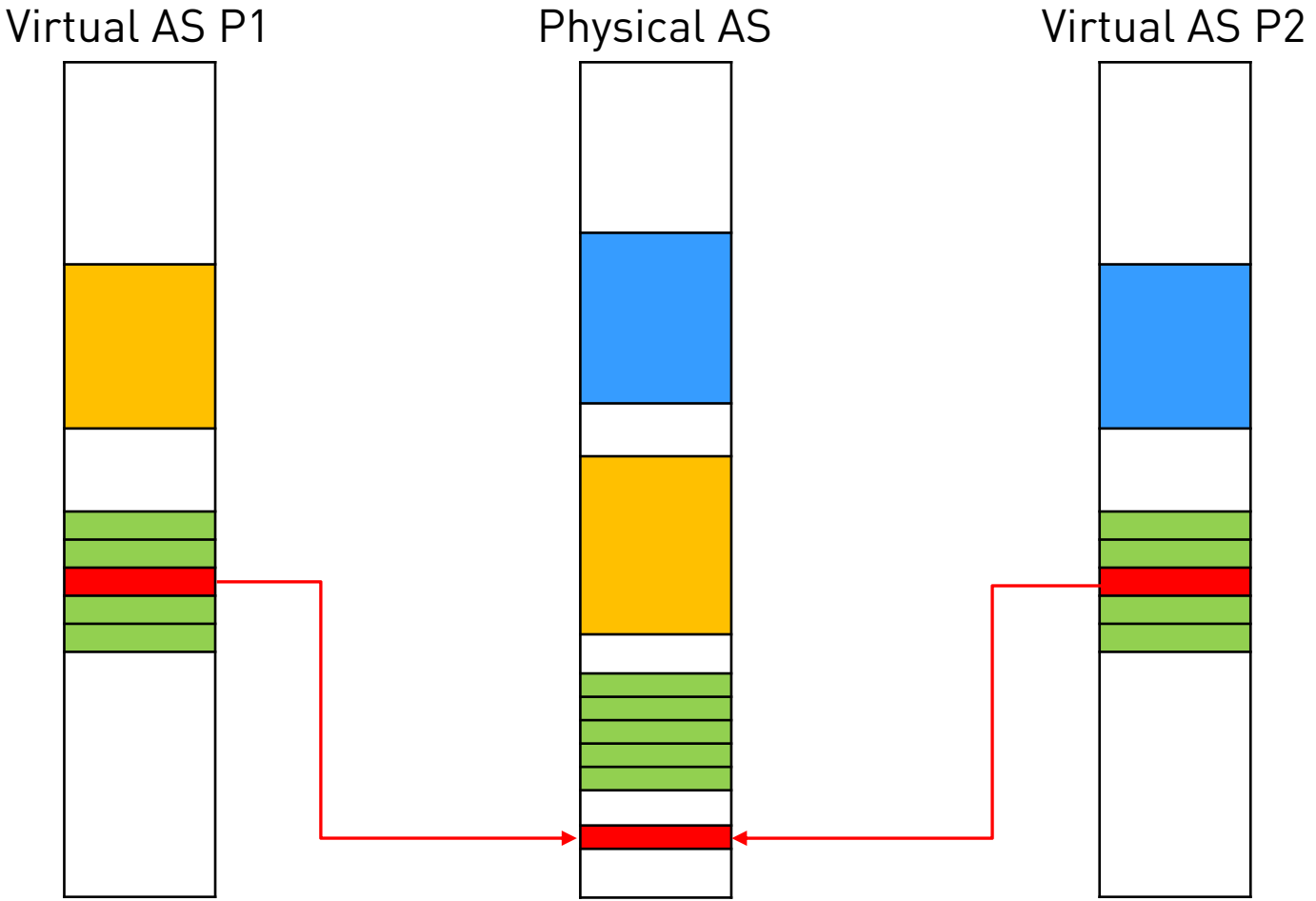
ERNW RESEARCH
pursuing knowledge.



ERNW RESEARCH
pursuing knowledge.



ERNW RESEARCH
pursuing knowledge.



After Memory Combining

ERNW RESEARCH
pursuing knowledge.

QueryWorkingSetEx

- The same approach as PrototypePte, but called live from user-space via the *QueryWorkingSetEx* API.

```
PSAPI_WORKING_SET_EX_INFORMATION WorkingSets = { 0 };
WorkingSets.VirtualAddress = 0x12340;
QueryWorkingSetEx(ProcHandle, &WorkingSets, sizeof(PSAPI_WORKING_SET_EX_INFORMATION);
if (!WorkingSets.VirtualAttributes.Shared) {
     # Page is considered modified
}
```

ERNW RESEARCH pursuing knowledge.

QueryWorkingSetEx

- This approach can be translated to the question (the same as with the PrototypePte approach):

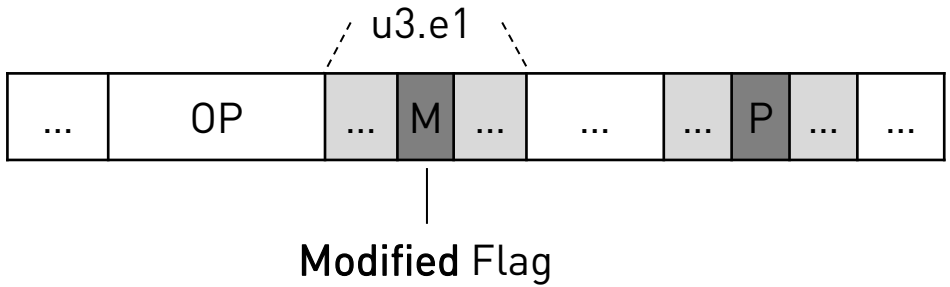
Does the current page belong to a memory-mapped image file and if so, **is it shareable**?

- The Problem: Again, **Memory Combining**.

ERNW RESEARCH
pursuing knowledge.

ModifiedList

- Uses the MMPFN's *Modified* field in order to identify modified pages for memory-mapped image files.



ERNW RESEARCH pursuing knowledge.

ModifiedList

- This approach can be translated to the question:

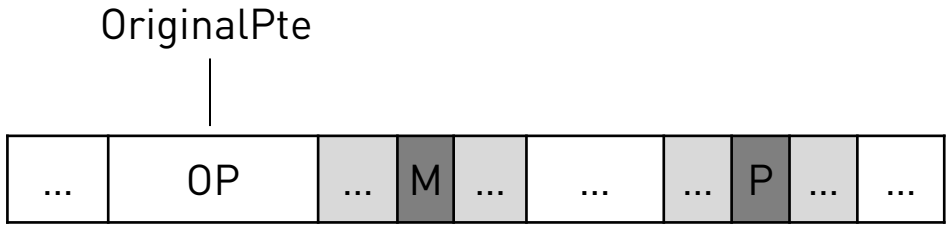
Does the current page belong to a memory-mapped image file and if so, **is it backed by a file**?

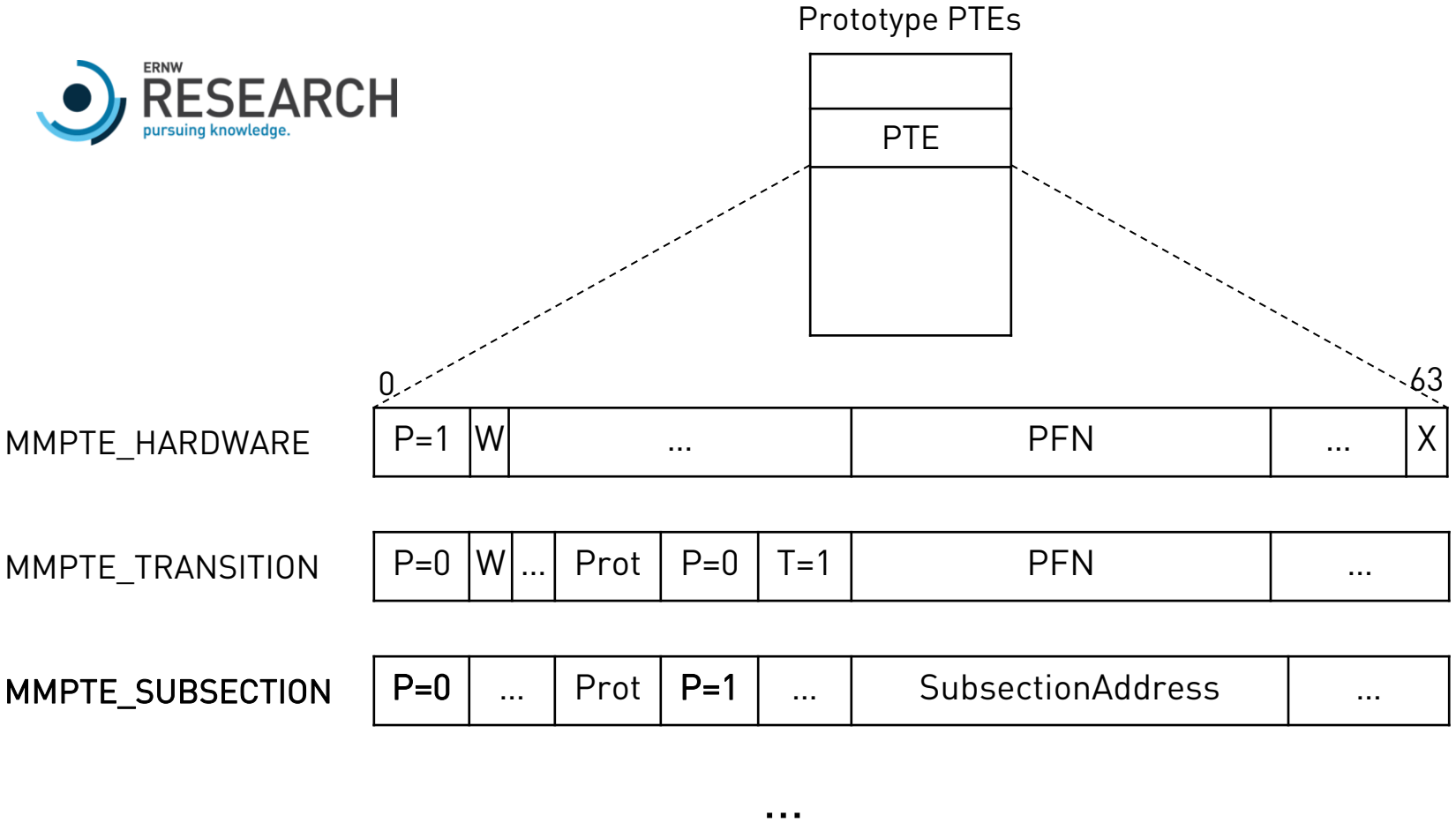
- Unmodified image-file pages are backed by the file on disk.
- If we modify such a page, it becomes a private copy and has no backing file anymore (at first!).
- The Problem: If a page gets modified, swapped to the Pagefile and read back in, **it is 'backed by a file': The Pagefile**.

ERNW RESEARCH
pursuing knowledge.

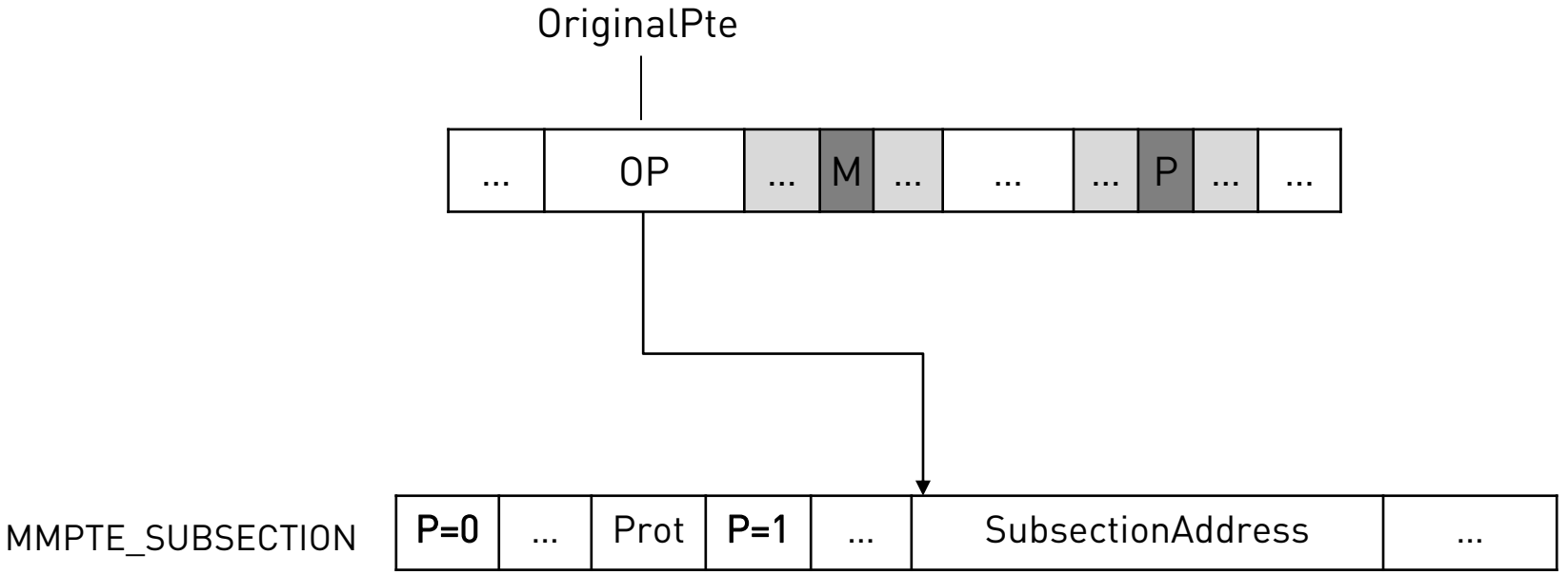
OriginalPte

"For a mapped file backed page, it stores an _MMPTE_SUBSECTION pointing to the subsection which covers the file page mapped by the PTE." [8, p. 352]

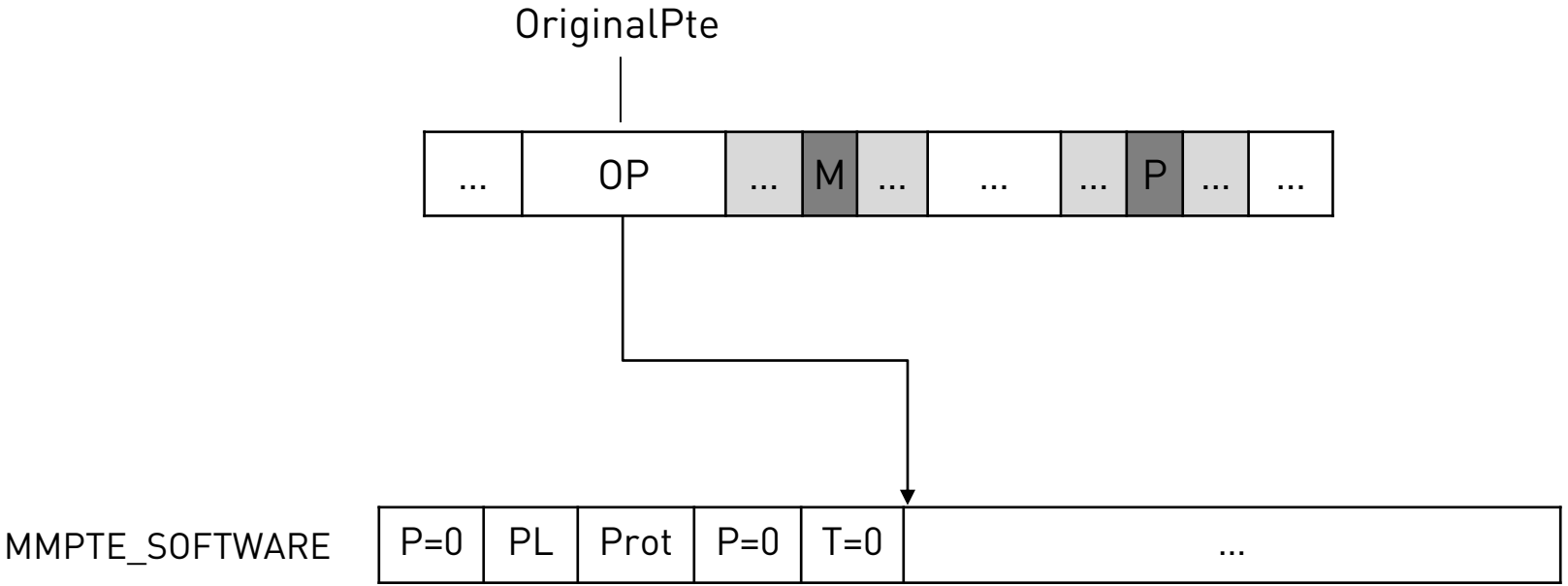


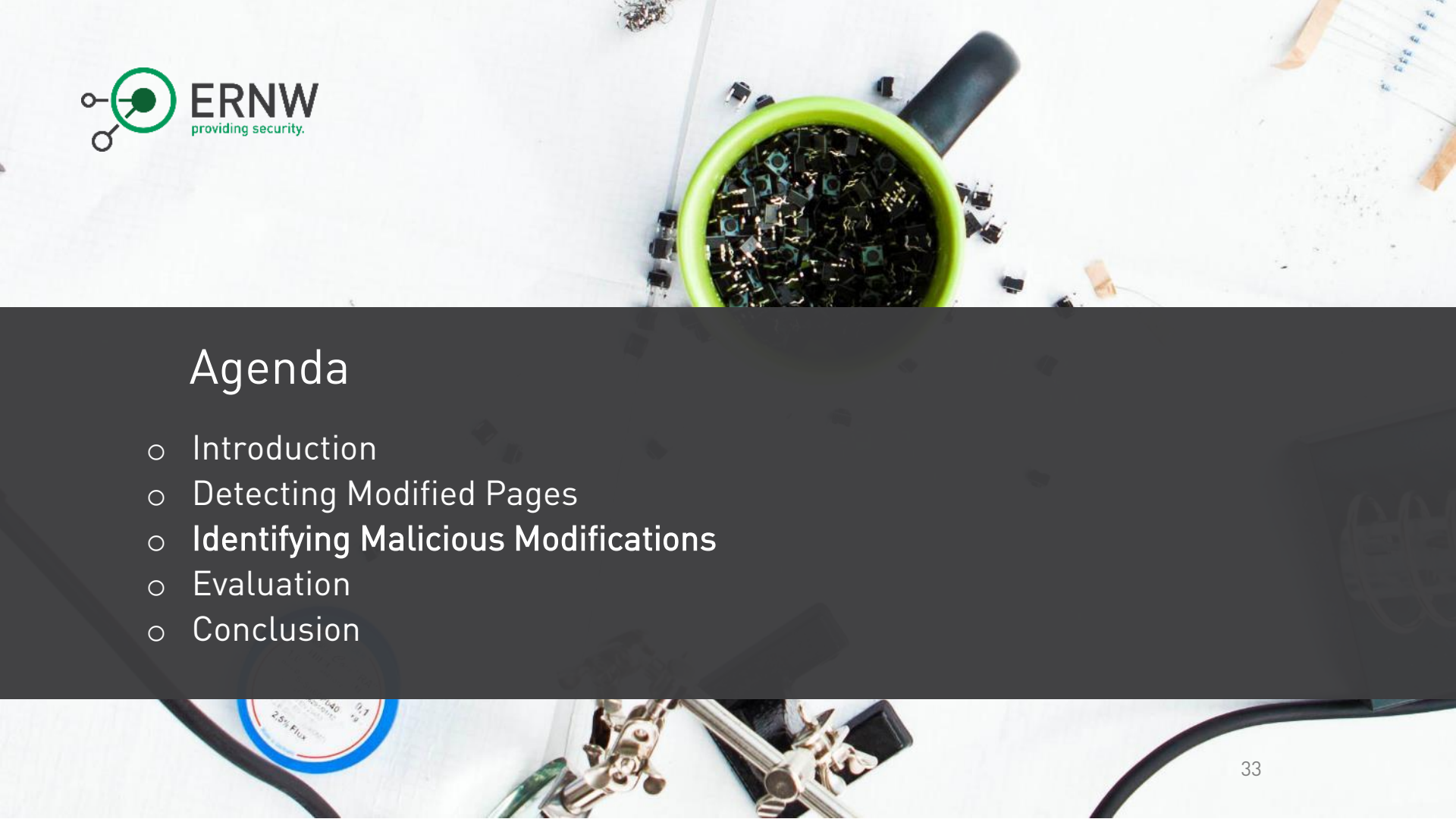

Prototype PTEs
PTE
0
63
MMPTE_HARDWARE
P=1
W
...
PFN
...
X
MMPTE_TRANSITION
P=0
W
...
Prot
P=0
T=1
PFN
...
MMPTE_SUBSECTION
P=0
...
Prot
P=1
...
SubsectionAddress
...
...

Unmodified MMIF



Modified MMIF





Agenda

- Introduction
- Detecting Modified Pages
- **Identifying Malicious Modifications**
- Evaluation
- Conclusion

ERNW RESEARCH pursuing knowledge.

Memory-Mapped Image Files

- There can be several legitimate/benign modifications:
  - Memory alignment
  - Relocations
  - Decryption/Unpacking
  - Modified global variables
  - IAT Patching
  - Benign Hooks

ERNW RESEARCH pursuing knowledge.

File Objects in Memory

- There are at least 4 possible representations for a MMIF, with 3 different versions of its content:
  - **VAD**: The process' version for that file, mapped in its process space.
  - **ImageSectionObject**: The basis for each process that loads that image file.
  - **DataSectionObject**: Same content as file on disk.
  - **SharedCacheMap**: Mainly a mechanism for efficient file loading. Its data seems to be the same as the DataSectionObject's data.

ERNW RESEARCH pursuing knowledge.

Identifying the Modified Bytes

- The *ImageSectionObject* is used as our Ground Truth
- Alignments and relocations are already applied.

- In this work, we focus only on executable memory.

ERNW RESEARCH
pursuing knowledge.

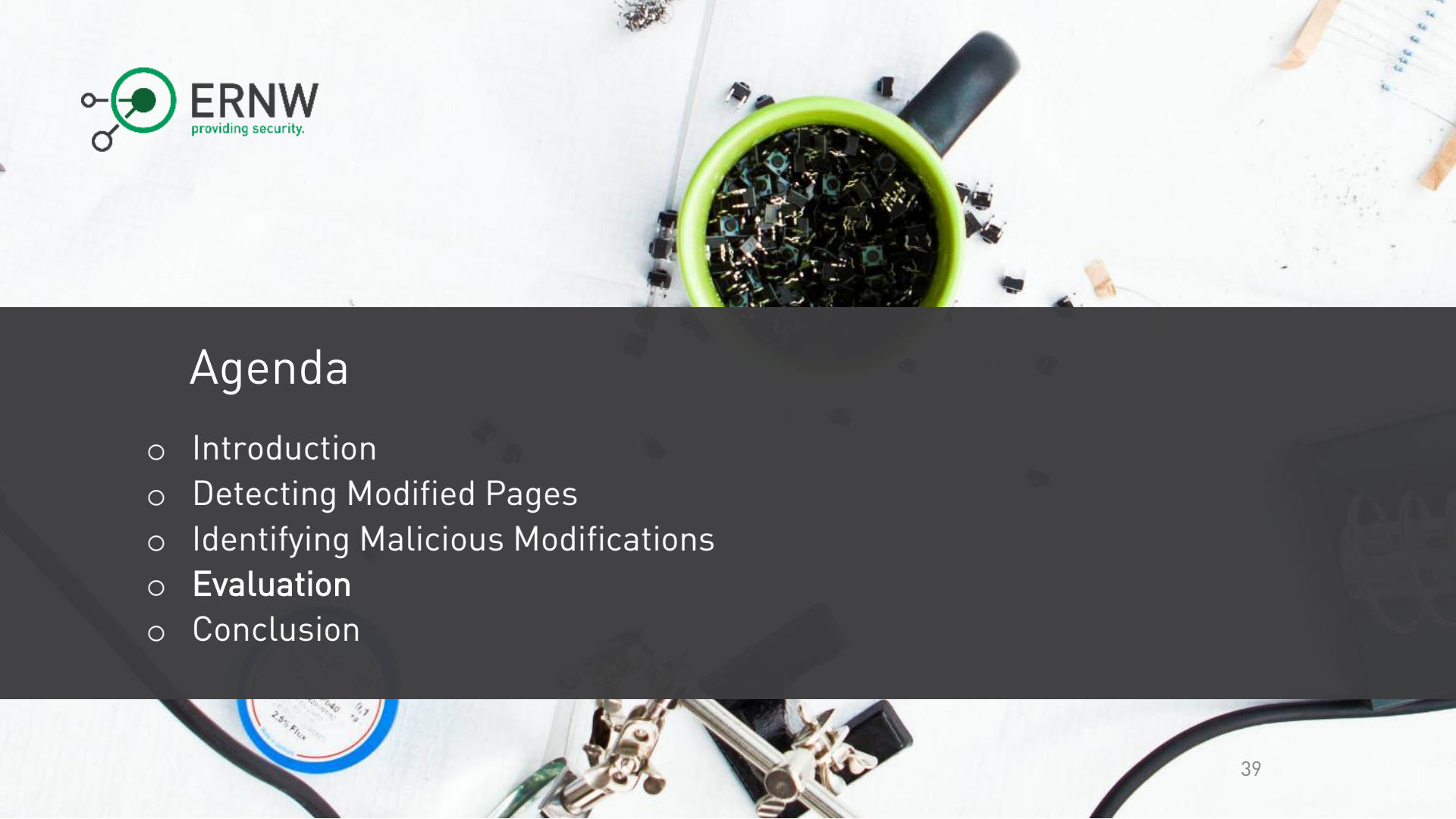
Detection Algorithm

1. Identify all executable pages belonging to MMIFs that have been modified.
2. Gather the corresponding page from the *ImageSectionObject* and compare their content on a byte level.
3. Filter any benign modification.
4. Present the results and for hooks, also the redirect target.

ERNW RESEARCH pursuing knowledge.

Benign Modifications

- Hooks
  - AntiVirus Hooks
  - Browser-Hooks: Sandbox functionality
  - *NtMapViewOfSection*-Hook by the Chromium-Project.
  - Microsoft Office Hooks (probably related to Microsoft's Application Virtualization).
- AVG's *SetUnhandledExceptionFilter*-Patch
- Modifications to the *clr.dll*



Agenda

- Introduction
- Detecting Modified Pages
- Identifying Malicious Modifications
- **Evaluation**
- Conclusion

ERNW RESEARCH pursuing knowledge.

imgmalfind - API Hooking

```
5244 msedge.exe \Windows\System32\ws2_32.dll .text 0x7fff27fb1d90
recv + 0x0  5

Original Data
48 89 5c 24 08 48 89 74   H.\$.H.t
0x7fff27fb1d90: mov qword ptr [rsp + 8], rbx

New Data
e9 3e f2 fd ff 48 89 74   .>...H.t
0x7fff27fb1d90: jmp 0x7fff27f90fd3

Target:
     The final target page is anonymous memory at 0x218857d0000
44 89 4c 24 20 44 89 44   D.L$.D.D
0x218857d7270: mov  dword ptr [rsp + 0x20], r9d
...
```

imgmalfind - AMSI Bypass

```
724  Powershell  \Windows\System32\amsi.dll .text
AmsiScanBuffer + 0x0  0x7ffdc57323e0  3

Original Data
4c 8b dc 49 89 5b 08 49   L..I.[.I
0x7ffdc57323e0: mov r11, rsp

New Data
31 c0 c3 49 89 5b 08 49   1..I.[.I
0x7ffdc57323e0: xor eax, eax
0x7ffdc57323e2: ret
```

imgmalfind - Module Stomping

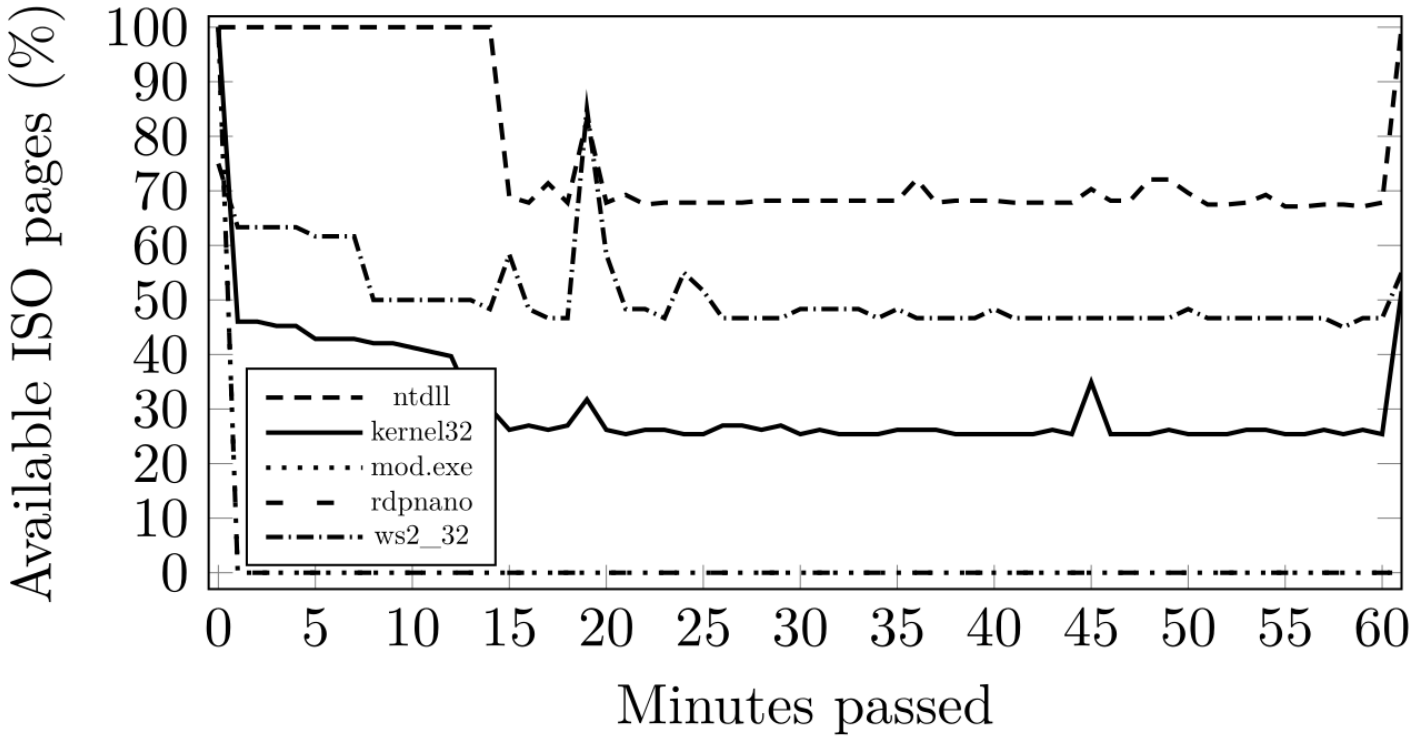
```
7052 Phantom  \Windows\System32\aadauthhelper.dll .text
0x7fff1c931000  N/A  270

Original Data
cc cc cc cc cc cc cc cc   ........
cc cc cc cc cc cc cc cc   ........
40 53 48 83 ec 40 4c 8b   @SH..@L.
94 24 80 00 00 00 48 8b   .$....H.
...

New Data
fc 48 83 e4 f0 eb 17 75   .H.....u
73 65 72 33 32 00 48 65   ser32.He
6c 6c 6f 20 66 72 6f 6d   llo.from
20 45 52 4e 57 00 e8 c8   .ERNW...
...
```

imgmalfind - Missing ISO Page

```
WARNING: Unable to get the base data for PID 10448 powershell.exe and
virtual address 0x7ffe99683000, so a direct comparison is not possible.
Hence, we simply show the first 64 bytes of the page...

10448  powershell.exe
\Windows\Microsoft.NET\Framework64\v4.0.30319\clr.dll  None found
0x7ffe99682000  N/A   -1

Orig Data



New Data
24 b8 00 00 00 4c 8b 84   $....L..
24 c0 00 00 00 4c 8b 8c   $....L..
24 c8 00 00 00 48 83 c4   $....H..
...
```

ERNW RESEARCH pursuing knowledge.

Purge Scenario



Agenda

- Introduction
- Motivation
- Detecting Modified Pages
- **Conclusion**

ERNW RESEARCH
pursuing knowledge.

Conclusion

- We introduced a more reliable approach for detecting modified MMIF pages.
- We can pinpoint the exact changes to a MMIF, no matter what and where they are.
- This allows us to detect not only API hooks but also other MMIF modifications such as AMSI and ETW bypasses, Module Stomping and Process Hollowing.
- It, moreover, detects modifications, where pattern matching fails.

ERNW RESEARCH pursuing knowledge.

Limitations

- Only applicable to modifications to MMIFs, not the covered attack techniques in general.
- If the ground truth (the corresponding page from the ImageSectionObject) is unavailable, we cannot pinpoint on the exact modified bytes.
- Packed/Crypted PE files.
- DKOM attacks.

ERNW RESEARCH pursuing knowledge.

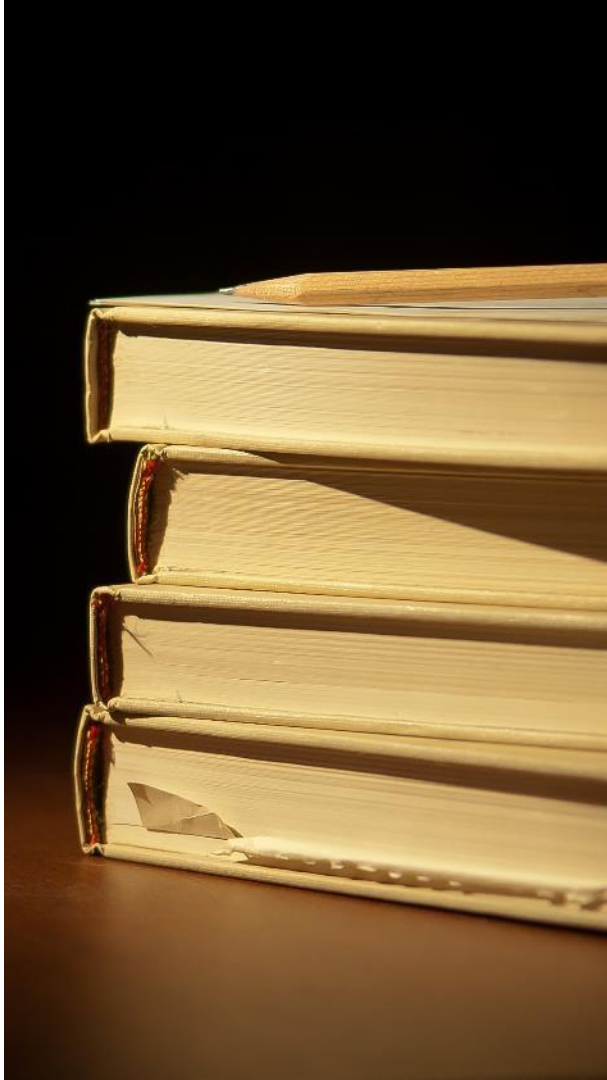
Future Work

- Support for using files from the FS or Microsoft's symbol server as ground truth.
- Fallback on unavailable ISO pages to approaches such as apihooks/hooktracer.
- Extend to non-executable memory:
  - IAT/EAT Hooking
  - Virtual table hooks.

ERNW RESEARCH
pursuing knowledge.

Sources

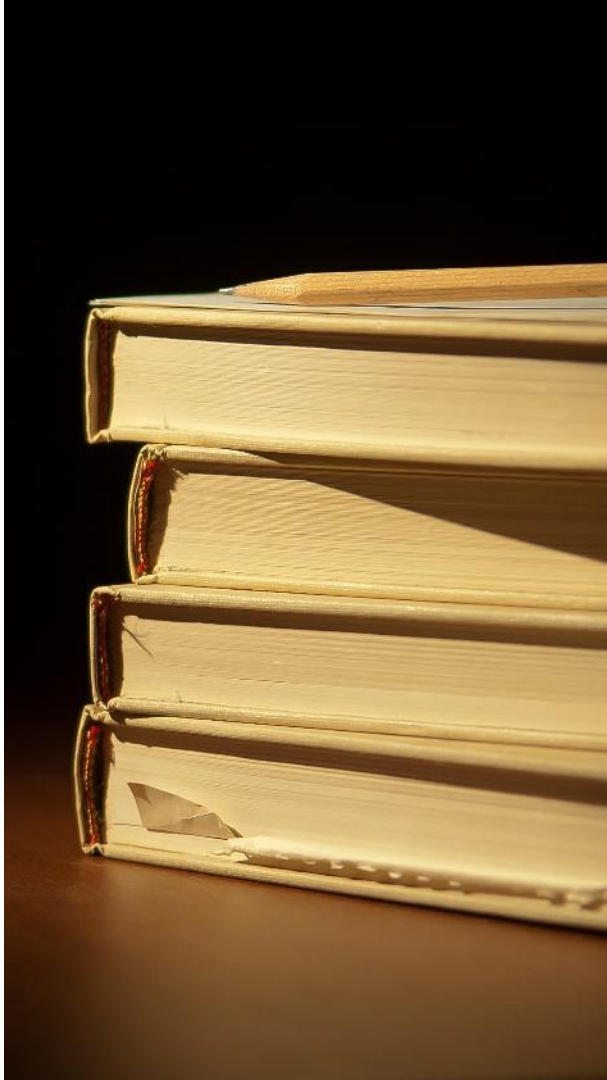
- [1] Cohen, M., 2016. Rekall and the windows pfn database. URL: https://web.archive.org/web/20170906073820/http://blog.rekall-forensic.com/2016/05/
- [2] Orr, F., 2020. Moneta - github repository. URL: https://github.com/forrest-orr/moneta/tree/4aad531dc7be04d094acfea2e2f66e411366a964
- [3] Hammond, A., 2019. Hiding malicious code with "module stomping": Part 3. URL: https://blog.f-secure.com/cowspot-real-time-module-stomping-detection/
- [4] White, A., Schatz, B., Foo, E., 2013. Integrity verification of user space code. Digital Investigation 10, S59–S68.



ERNW RESEARCH
pursuing knowledge.

Sources

- [6] Case, A., Jalalzai, M.M., Firoz-Ul-Amin, M., Maggio, R.D., Ali-Gombe, A., Sun, M., Richard III, G.G., 2019. Hooktracer: A system for automated and accessible api hooks analysis. Digital Investigation 29, S104–S112.
- [7] Manna, M., Case, A., Ali-Gombe, A., Richard III, G.G., 2022. Memory analysis of. net and. net core applications. Forensic Science International: Digital Investigation 42.
- [8] Martignetti, E., 2012. What Makes It Page? The Windows 7 (x64) Virtual Memory Manager



ERNW RESEARCH
pursuing knowledge.

Thank you for your Attention

Questions/Criticism/Remarks/Suggestions?

The online repository can be found at:
https://github.com/f-block/DFRWS-USA-2023

www.ernw.de

www.insinuator.net