Be very very quit. I'm hunting threats and actors.

/Jesper Mikkelsen

TREND MICRO™

Hunt with confidence

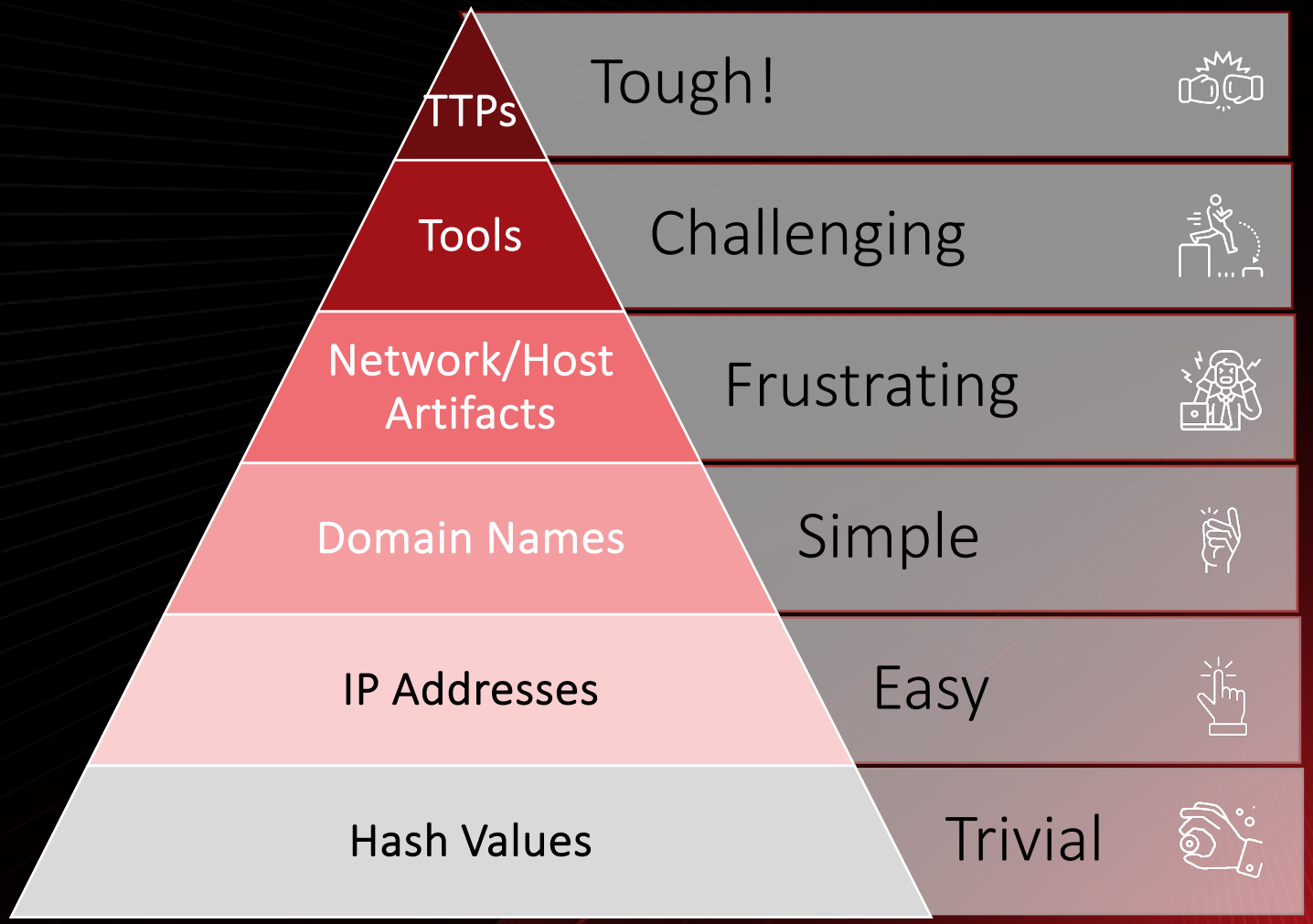
```
rule Find_Jesper_Mikkelsen
{
    strings:
        $str0="Technical Director Nordics"  ascii wide
        $str1="Been @ Trend Micro 9 years +"
        $str2="Former Pen-tester | Ethical Hacker" nocase
        $str3="Malware reversing" ascii wide
        $str4="19 years+" ascii wide
        $str5="https://twitter.com/jespermikkelsen" ascii wide

    condition:
        all of them and (infosec_geek=100%)
}
```

What is Threat Hunting?

Threat hunting is the process of proactively searching through computer networks or data to find adversary activities before the real damage or breach happen

Let’s use the Pyramid of Pain for Hunting



David Bianco's Pyramid of Pain

Typical Threat Hunting Hypothesis Creation Process

Threat Intelligence

IoC's, Tools, Actors, Campaign, TTP's for emerging threats

Context of the Organization Environment

OS, Software, Previous Incidents, Partners, Vulnerability scans, penetration testing results, policies

Based on findings, define key elements of your hypothesis:

Potential targets: systems, software, privileged users…

Adversary techniques: TTPs

Analytics: Toolsets needed to verify your hypothesis

The BAD news!

Software packing.

Anti-debugging

Evasion techniques

Anti-sandboxing

The BAD news!

Anti-sandboxing

Evasion

Bandit Stealer checks for the following to determine if it's running in a sandbox environment and alters its behavior accordingly to avoid detection or analysis:

- container
- jail
- KVM
- QEMU
- sandbox
- Virtual Machine
- VirtualBox
- VMware
- Xen

```
mov     rbp, [rbp+0]
lea     rcx, aVmware     ; "VMware"
mov     [rsp+110h+var_98], rcx
mov     [rsp+110h+var_90], 6
lea     rcx, aVirtualbox ; "VirtualBox"
mov     [rsp+110h+var_88], rcx
mov     [rsp+110h+var_80], 0Ah
lea     rcx, aQemu       ; "QEMU"
mov     [rsp+110h+var_78], rcx
mov     [rsp+110h+var_70], 4
lea     rcx, aXen        ; "Xen"
mov     [rsp+110h+var_68], rcx
mov     [rsp+110h+var_60], 3
lea     rcx, aKvm        ; "KVM"
mov     [rsp+110h+var_58], rcx
mov     [rsp+110h+var_50], 3
lea     rcx, aVirtualMachine ; "Virtual Machine"
mov     [rsp+110h+var_48], rcx
mov     [rsp+110h+var_40], 0Fh
lea     rcx, aSandbox    ; "sandbox"
mov     [rsp+110h+var_38], rcx
mov     [rsp+110h+var_30], 7
lea     rcx, aJail       ; "jail"
mov     [rsp+110h+var_28], rcx
mov     [rsp+110h+var_20], 4
lea     rcx, aContainer ; "container"
mov     [rsp+110h+var_18], rcx
mov     [rsp+110h+var_10], 9
lea     rax, aProcSelfStatus ; "/proc/self/status"
```


The BAD news!

Evasion

Notable Threat Characteristics

Anti-security, self-preservation (8)

Characteristic	Significance	Details
Attempts to evade detection and analysis	■	Process ID: 3860 Info: Delays execution
Attempts to detect active running processes	■	Process ID: 3860 Info: enum processes by WMI
Attempts to detect active running processes	■	Process ID: 3860 Info: enum processes
Attempts to detect sandbox strings	■	Sample attempted to detect Sandboxie using the following string: SbieDll.dll
Attempts to detect sandbox strings	■	Sample attempted to detect VirtualBox using the following string: VirtualBox
Attempts to detect sandbox strings	■	Sample attempted to detect VirtualBox using the following string: inVirtualBox
Attempts to detect sandbox strings	■	Sample attempted to detect Sandboxie using the following string: DetectSandboxie
Attempts to detect sandbox strings	■	Sample attempted to detect Sandboxie using the following string: inSandboxie

Autostart or other system reconfiguration (2)

```
mov     [rsp+110h+var_38], rcx
mov     [rsp+110h+var_30], 7
lea     rcx, aJail       ; "jail"
mov     [rsp+110h+var_28], rcx
mov     [rsp+110h+var_20], 4
lea     rcx, aContainer ; "container"
mov     [rsp+110h+var_18], rcx
mov     [rsp+110h+var_10], 9
lea     rax, aProcSelfStatus ; "/proc/self/status"
```


The BAD news!

Evasion

EVIL!

Notable Threat Characteristics

Anti-security, self-preservation (8)

Characteristic	Significance	Details
Attempts to evade detection and analysis	■	Process ID: 3860 Info: Delays execution
Attempts to detect active running processes	■	Process ID: 3860 Info: enum processes by WMI
Attempts to detect active running processes	■	Process ID: 3860 Info: enum processes
Attempts to detect sandbox strings	■	Sample attempted to detect Sandboxie using the following string: SbieDll.dll
Attempts to detect sandbox strings	■	Sample attempted to detect VirtualBox using the following string: VirtualBox
Attempts to detect sandbox strings	■	Sample attempted to detect VirtualBox using the following string: inVirtualBox
Attempts to detect sandbox strings	■	Sample attempted to detect Sandboxie using the following string: DetectSandboxie
Attempts to detect sandbox strings	■	Sample attempted to detect Sandboxie using the following string: inSandboxie

Autostart or other system reconfiguration (2)

```
mov     [rsp+110h+var_38], rcx
mov     [rsp+110h+var_30], 7
lea     rcx, aJail       ; "jail"
mov     [rsp+110h+var_28], rcx
mov     [rsp+110h+var_20], 4
lea     rcx, aContainer ; "container"
mov     [rsp+110h+var_18], rcx
mov     [rsp+110h+var_10], 9
lea     rax, aProcSelfStatus ; "/proc/self/status"
```


The BAD news!

Software packing.

Zingdoor

Zingdoor is a new HTTP backdoor written in Go. While we first encountered Zingdoor in April 2023, some logs indicate that the earliest developments of this backdoor took place in June 2022. However, it had rarely been seen in the wild and had only been observed being used in a limited number of victims, likely as a newly designed backdoor with cross-platform capabilities. Zingdoor is packed using UPX and heavily obfuscated by a custom obfuscator engine.

We noted that Zingdoor adopts anti-UPX unpacking techniques. Generally, the magic number of UPX is "UPX!", but in this case it was modified to "MSE!", and the UPX application cannot unpack this modified file. This technique is easy and in internet of things (IoT) types of malware, but it is considered rare in APT activities.

The BAD news!

Evasion

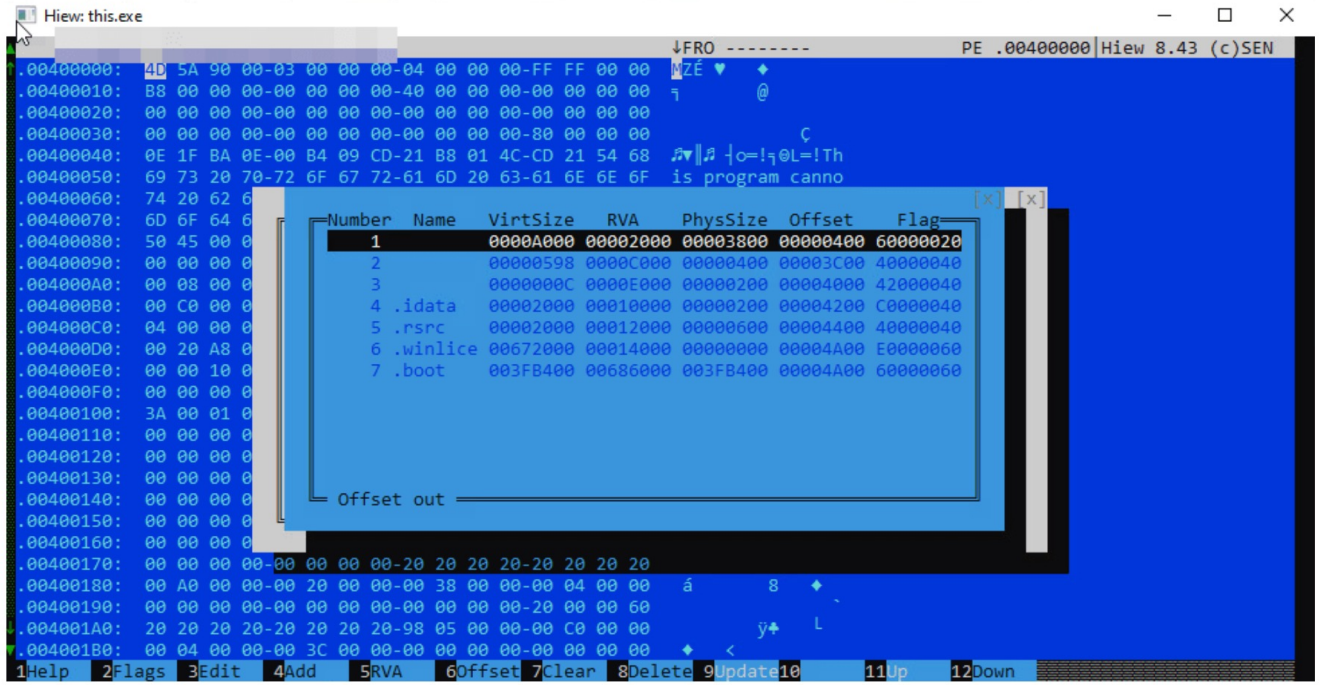
Bandit Stealer checks
behavior accordingly

- container
- jail
- KVM
- QEMU
- sandbox
- Virtual Machine
- VirtualBox
- VMware
- Xen

Discovery, reconnaissance, and staging

In April, we found a couple of ransomware activities that appear to be injected in legitimate processes. By tracing these activities back to the source process, we found that the ransomware appeared as an activity loaded into memory from a Cobalt Strike beacon. In some instances, the attackers dropped the ransomware in a folder or drive as a **.log* file:

- *E:\ITS.log*
- *C:\[Redacted]\Aps.log*



The BAD news!

dnSpy v6.1.8 (32-bit, .NET, Debugging)

File Edit View Debug Window Help C# Continue

Assembly Explorer

- System.Private.CoreLib (5.0.0.0)
- System.Private.Uri (5.0.0.0)
- System.Linq (5.0.0.0)
- System.Private.Xml (5.0.0.0)
- System.Xaml (5.0.0.0)
- WindowsBase (5.0.0.0)
- PresentationCore (5.0.0.0)
- PresentationFramework (5.0.0.0)
- dnlib (3.3.2.0)
- dnSpy (6.1.8.0)
  - dnSpy.dll
- FluffyPenguinsShe11 (1.0.0.0)
  - FluffyPenguinsShe11.exe
    - PE
    - Type References
    - References
    - { } -
      - <Module> @02000001
        - Base Type and Interfac
        - Derived Types
        - .cctor() : void @06000
        - Decompress(byte[]) : b
        - Decrypt(uint[], uint) : (
        - Main(string[]) : int @0
        - Resolve(object, Resolv
        - DataField : <Module>
        - key : byte[] @0400000
        - BitDecoder @02000003
        - BitTreeDecoder @0200
        - DataType @02000002
        - Decoder @02000005
        - LzmaDecoder @02000
        - OutWindow @020000
        - State @0200000B
      - ConfusedByAttribute @02

<Module>

```
55          ulong num4 = num % 9067703UL;
56          for (int k = 0; k < array4.Length; k++)
57          {
58              byte[] array5 = array4;
59              int num5 = k;
60              array5[num5] ^= (byte)num;
61              if ((k & 255) == 0)
62              {
63                  num = num * num % 9067703UL;
64              }
65          }
66          return result;
67      }
68
69      // Token: 0x06000002 RID: 2 RVA: 0x00003424 File Offset: 0x00001624
70      [STAThread]
71      private static int Main(string[] A_0)
72      {
73          uint[] array = new uint[]
74          {
75              1595157935U,
76              841091026U,
77              4104602669U,
```

100 %

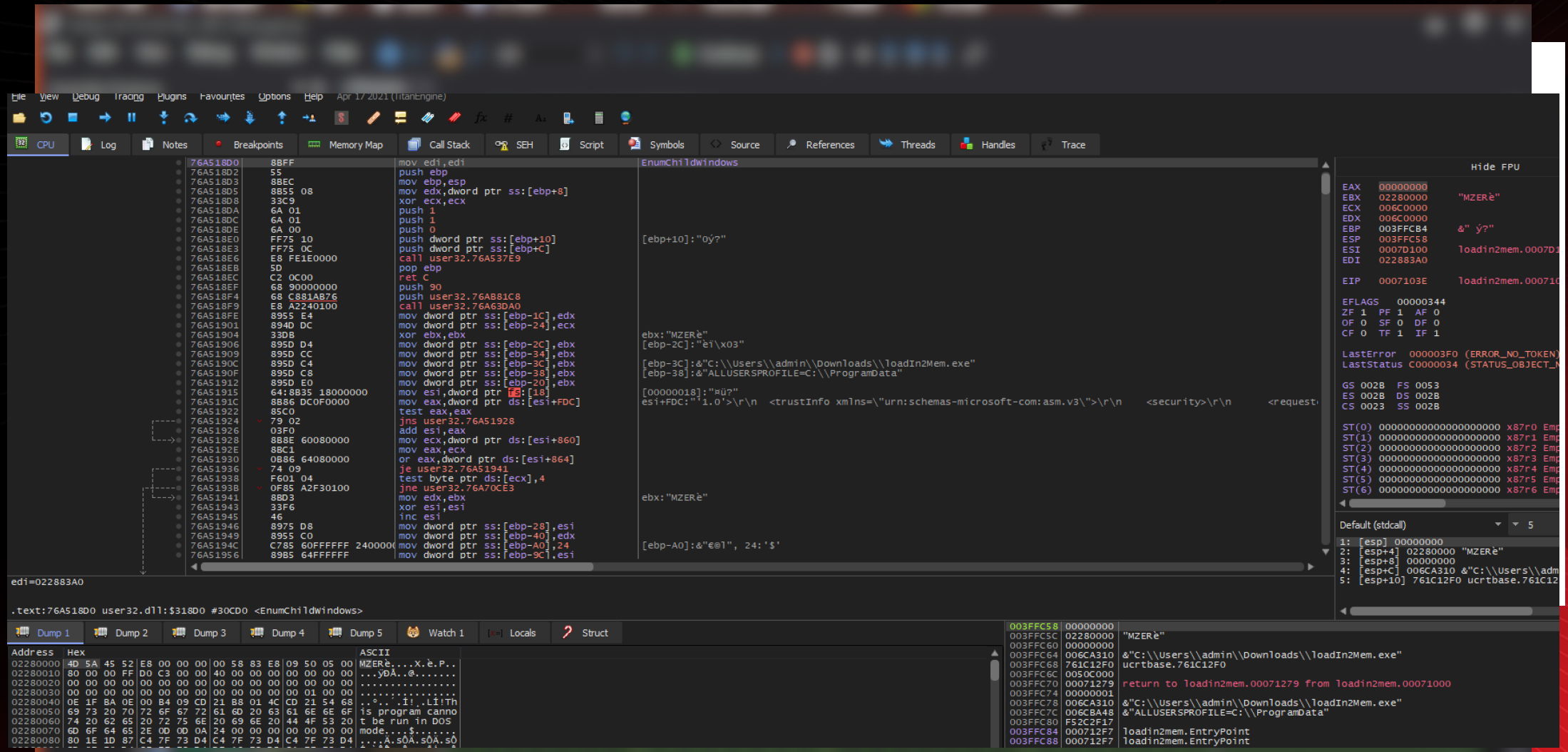
Memory 2

```
0310BF00 00 00 00 00 00 00 00 00 D0 68 6E 05 00 24 00 00 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 .........hn..$..MZ..............
0310BF20 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ........@.......................
0310BF40 00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 ........................!..L.!Th
0310BF60 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 is program cannot be run in DOS
0310BF80 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 50 45 00 00 4C 01 03 00 E2 C0 5A 63 00 00 00 00 mode....$.......PE..L.....Zc....
0310BFA0 00 00 00 00 E0 00 22 01 0B 01 30 00 00 1A 00 00 00 08 00 00 00 00 00 00 DE 38 00 00 00 20 00 00 ......"...0.............8... ..
0310BFC0 00 40 00 00 00 00 40 00 00 20 00 00 00 02 00 00 04 00 00 00 00 00 00 00 06 00 00 00 00 00 00 00 .@....@.. ......................
0310BFE0 00 80 00 00 00 02 00 00 00 00 00 00 03 00 60 85 00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00 ..............`...............
0310C000 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 8C 38 00 00 4F 00 00 00 00 40 00 00 F8 05 00 00 .................8..O....@......
0310C020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 60 00 00 0C 00 00 00 00 00 00 00 00 00 00 00 .................`..............
0310C040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................................
0310C060 00 00 00 00 00 00 00 00 00 20 00 00 08 00 00 00 00 00 00 00 00 00 00 00 08 20 00 00 48 00 00 00 ......... ............... ..H...
```

100 %

Locals Memory 1 Memory 2

The BAD news!


File View Debug Tracing Plugins Favourites Options Help Apr 17 2021 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
76A518D0 8BFF mov edi,edi EnumChildWindows
76A518D2 55 push ebp
76A518D3 8BEC mov ebp,esp
76A518D5 8B55 08 mov edx,dword ptr ss:[ebp+8]
76A518D8 33C9 xor ecx,ecx
76A518DA 6A 01 push 1
76A518DC 6A 01 push 1
76A518DE 6A 00 push 0
76A518E0 FF75 10 push dword ptr ss:[ebp+10] [ebp+10]:"0ý?"
76A518E3 FF75 0C push dword ptr ss:[ebp+C]
76A518E6 E8 FE1E0000 call user32.76A537E9
76A518EB 5D pop ebp
76A518EC C2 0C00 ret C
76A518EF 68 90000000 push 90
76A518F4 68 C881AB76 push user32.76AB81C8
76A518F9 E8 A2240100 call user32.76A63DA0
76A518FE 8955 E4 mov dword ptr ss:[ebp-1C],edx
76A51901 894D DC mov dword ptr ss:[ebp-24],ecx
76A51904 33DB xor ebx,ebx ebx:"MZERè"
76A51906 895D D4 mov dword ptr ss:[ebp-2C],ebx [ebp-2C]:"ëï\x03"
76A51909 895D CC mov dword ptr ss:[ebp-34],ebx
76A5190C 895D C4 mov dword ptr ss:[ebp-3C],ebx [ebp-3C]:&"C:\\Users\\admin\\Downloads\\loadIn2Mem.exe"
76A5190F 895D C8 mov dword ptr ss:[ebp-38],ebx [ebp-38]:&"ALLUSERSPROFILE=C:\\ProgramData"
76A51912 895D E0 mov dword ptr ss:[ebp-20],ebx
76A51915 64:8B35 18000000 mov esi,dword ptr fs:[18] [00000018]:"Hü?"
76A5191C 8B86 DC0F0000 mov eax,dword ptr ds:[esi+FDC] esi+FDC:"'1.0'>\r\n <trustInfo xmlns=\"urn:schemas-microsoft-com:asm.v3\">\r\n <security>\r\n <request
76A51922 85C0 test eax,eax
76A51924 79 02 jns user32.76A51928
76A51926 03F0 add esi,eax
76A51928 8B8E 60080000 mov ecx,dword ptr ds:[esi+860]
76A5192E 8BC1 mov eax,ecx
76A51930 0B86 64080000 or eax,dword ptr ds:[esi+864]
76A51936 74 09 je user32.76A51941
76A51938 F601 04 test byte ptr ds:[ecx],4
76A5193B 0F85 A2F30100 jne user32.76A70CE3
76A51941 8BD3 mov edx,ebx ebx:"MZERè"
76A51943 33F6 xor esi,esi
76A51945 46 inc esi
76A51946 8975 D8 mov dword ptr ss:[ebp-28],esi
76A51949 8955 C0 mov dword ptr ss:[ebp-40],edx
76A5194C C785 60FFFFFF 240000 mov dword ptr ss:[ebp-A0],24 [ebp-A0]:&"€@l", 24:'$'
76A51956 89B5 64FFFFFF mov dword ptr ss:[ebp-9C],esi
edi=022883A0
.text:76A518D0 user32.dll:$318D0 #30CD0 <EnumChildWindows>
Hide FPU
EAX 00000000
EBX 02280000 "MZERè"
ECX 006C0000
EDX 006C0000
EBP 003FFCB4 &" ý?"
ESP 003FFC58
ESI 0007D100 loadin2mem.0007D1
EDI 022883A0
EIP 0007103E loadin2mem.00071
EFLAGS 00000344
ZF 1 PF 1 AF 0
OF 0 SF 0 DF 0
CF 0 TF 1 IF 1
LastError 000003F0 (ERROR_NO_TOKEN)
LastStatus C0000034 (STATUS_OBJECT_N
GS 002B FS 0053
ES 002B DS 002B
CS 0023 SS 002B
ST(0) 0000000000000000000 x87r0 Emp
ST(1) 0000000000000000000 x87r1 Emp
ST(2) 0000000000000000000 x87r2 Emp
ST(3) 0000000000000000000 x87r3 Emp
ST(4) 0000000000000000000 x87r4 Emp
ST(5) 0000000000000000000 x87r5 Emp
ST(6) 0000000000000000000 x87r6 Emp
Default (stdcall) 5
1: [esp] 00000000
2: [esp+4] 02280000 "MZERè"
3: [esp+8] 00000000
4: [esp+C] 006CA310 &"C:\\Users\\adm
5: [esp+10] 761C12F0 ucrtbase.761C12
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct
Address Hex ASCII
02280000 4D 5A 45 52 E8 00 00 00 00 58 83 E8 09 50 05 00 MZERè....X.è.P..
02280010 80 00 00 FF D0 C3 00 00 40 00 00 00 00 00 00 00 ...ÿÐÃ..@.....
02280020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
02280030 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 .....
02280040 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 ..º..´.Í!¸.LÍ!Th
02280050 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F is program canno
02280060 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 t be run in DOS
02280070 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 mode....$.....
02280080 80 1E 1D 87 C4 7F 73 D4 C4 7F 73 D4 C4 7F 73 D4 ....Ä.sÔÄ.sÔÄ.sÔ
003FFC58 00000000
003FFC5C 02280000 "MZERè"
003FFC60 00000000
003FFC64 006CA310 &"C:\\Users\\admin\\Downloads\\loadIn2Mem.exe"
003FFC68 761C12F0 ucrtbase.761C12F0
003FFC6C 0050C000
003FFC70 00071279 return to loadin2mem.00071279 from loadin2mem.00071000
003FFC74 00000001
003FFC78 006CA310 &"C:\\Users\\admin\\Downloads\\loadIn2Mem.exe"
003FFC7C 006CBA48 &"ALLUSERSPROFILE=C:\\ProgramData"
003FFC80 F52C2F17
003FFC84 000712F7 loadin2mem.EntryPoint
003FFC88 000712F7 loadin2mem.EntryPoint

The BAD news!

dnSpy v6.1.8 (32-bit, .NET, Debugging)

```
76A518D0  8BFF          mov edi,edi                 EnumChildWindows
76A518D2  55            push ebp
76A518D3  8BEC          mov ebp,esp
76A518D5  8B55 08       mov edx,dword ptr ss:[ebp+8]
76A518D8  33C9          xor ecx,ecx
76A518DA  6A 01         push 1
76A518DC  6A 01         push 1
76A518DE  6A 00         push 0
76A518E0  FF75 10       push dword ptr ss:[ebp+10]  [ebp+10]:"0ý?"
76A518E3  FF75 0C       push dword ptr ss:[ebp+C]
76A518E6  E8 FE1E0000   call user32.76A537E9
76A518EB  5D            pop ebp
76A518EC  C2 0C00       ret C
76A518EF  68 90000000   push 90
76A518F4  68 C881AB76   push user32.76AB81C8
76A518F9  E8 A2240100   call user32.76A63DA0
76A518FE  8955 E4       mov dword ptr ss:[ebp-1C],edx
76A5190D  894D DC       mov dword ptr ss:[ebp-24],ecx
76A51904  33DB          xor ebx,ebx                 ebx:"MZERè"
76A51906  895D D4       mov dword ptr ss:[ebp-2C],ebx  [ebp-2C]:"èÿ\x03"
76A51909  895D CC       mov dword ptr ss:[ebp-34],ebx
76A5190C  895D C4       mov dword ptr ss:[ebp-3C],ebx  [ebp-3C]:&"C:\\Users\\admin\\Downloads\\loadIn2Mem.exe"
76A5190F  895D C8       mov dword ptr ss:[ebp-38],ebx  [ebp-38]:&"ALLUSERSPROFILE=C:\\ProgramData"
76A51912  895D E0       mov dword ptr ss:[ebp-20],ebx
76A51915  64:8B35 18000000  mov esi,dword ptr fs:[18]  [00000018]:"¤ü?"
76A5191C  8B86 DC0F0000 mov eax,dword ptr ds:[esi+FDC]  esi+FDC:"'1.0'>\r\n  <trustInfo xmlns=\"urn:schemas-microsoft-com:asm.v3\">\r\n    <security>\r\n      <request
76A51922  85C0          test eax,eax
76A51924  79 02         jns user32.76A51928
76A51926  03F0          add esi,eax
76A51928  8B8E 60080000 mov ecx,dword ptr ds:[esi+860]
76A5192E  8BC1          mov eax,ecx
76A51930  0B86 64080000 or eax,dword ptr ds:[esi+864]
76A51936  74 09         je user32.76A51941
76A51938  F601 04       test byte ptr ds:[ecx],4
76A5193B  0F85 A2F30100 jne user32.76A70CE3
76A51941  8BD3          mov edx,ebx                 ebx:"MZERè"
76A51943  33F6          xor esi,esi
76A51945  46            inc esi
76A51946  8975 D8       mov dword ptr ss:[ebp-28],esi
76A51949  8955 C0       mov dword ptr ss:[ebp-40],edx
76A5194C  C785 60FFFFFF 240000  mov dword ptr ss:[ebp-A0],24  [ebp-A0]:&"€@l", 24:'$'
76A51956  89B5 64FFFFFF mov dword ptr ss:[ebp-9C],esi
edi=022883A0

.text:76A518D0 user32.dll:$318D0 #30CD0 <EnumChildWindows>
```

```
EAX   00000000
EBX   02280000   "MZERè"
ECX   006C0000
EDX   006C0000
EBP   003FFCB4   &" ý?"
ESP   003FFC58
ESI   0007D100   loadin2mem.0007D1
EDI   022883A0

EIP   0007103E   loadin2mem.00071C

EFLAGS  00000344
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 1  IF 1

LastError  000003F0 (ERROR_NO_TOKEN)
LastStatus C0000034 (STATUS_OBJECT_N

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 00000000000000000000 x87r0 Emp
ST(1) 00000000000000000000 x87r1 Emp
ST(2) 00000000000000000000 x87r2 Emp
ST(3) 00000000000000000000 x87r3 Emp
ST(4) 00000000000000000000 x87r4 Emp
ST(5) 00000000000000000000 x87r5 Emp
ST(6) 00000000000000000000 x87r6 Emp

Default (stdcall)  5
1: [esp] 00000000
2: [esp+4] 02280000 "MZERè"
3: [esp+8] 00000000
4: [esp+C] 006CA310 &"C:\\Users\\adm
5: [esp+10] 761C12F0 ucrtbase.761C12
```

```
Address   Hex                                       ASCII
02280000  4D 5A 45 52 E8 00 00 00 00 58 83 E8 09 50 05 00  MZERè....X.è.P..
02280010  80 00 00 FF D0 C3 00 00 40 00 00 00 00 00 00 00  ...ÿÐÃ..@.......
02280020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02280030  00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00  ................
02280040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
02280050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
02280060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS
02280070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
02280080  80 1E 1D 87 C4 7F 73 D4 C4 7F 73 D4 C4 7F 73 D4  ....Ä.sÔÄ.sÔÄ.sÔ
```

```
003FFC58  00000000
003FFC5C  02280000  "MZERè"
003FFC60  00000000
003FFC64  006CA310  &"C:\\Users\\admin\\Downloads\\loadIn2Mem.exe"
003FFC68  761C12F0  ucrtbase.761C12F0
003FFC6C  0050C000
003FFC70  00071279  return to loadin2mem.00071279 from loadin2mem.00071000
003FFC74  00000001
003FFC78  006CA310  &"C:\\Users\\admin\\Downloads\\loadIn2Mem.exe"
003FFC7C  006CBA48  &"ALLUSERSPROFILE=C:\\ProgramData"
003FFC80  F52C2F17
003FFC84  000712F7  loadin2mem.EntryPoint
003FFC88  000712F7  loadin2mem.EntryPoint
```

```
DataType @02000002
Decoder @02000005
LzmaDecoder @02000
OutWindow @0200000
State @0200000B
ConfusedByAttribute @02
```

```
0310BFE0  00 80 00 00 00 02 00 00 00 00 00 00 03 00 60 85 00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00  ..............`.................
0310C000  00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 3C 38 00 00 4F 00 00 00 00 40 00 00 F8 05 00 00  ................8..O....@......
0310C020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 60 00 00 0C 00 00 00 00 00 00 00 00 00 00 00  ................`...............
0310C040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
0310C060  00 00 00 00 00 00 00 00 00 20 00 00 08 00 00 00 00 00 00 00 00 00 00 00 08 20 00 00 48 00 00 00  ......... ................ ..H...
```

100 %

Locals Memory 1 Memory 2

The good news!

The good news!


```
"score": "6.033966452662256",
"machine_learning": {
  "classification": "malicious",
  "probability": "96.19 %"
},
"mitre_attck": [
  {
    "id": "T1040",
    "description": "Network Hop"
```


```
[ /core > unipacker 602c9b7452e4618580c0c3cd43e3a2c56b438c0a92aaeda32fe0ff19ad71843c
Next up: Sample: [PEtite] 602c9b7452e4618580c0c3cd43e3a2c56b438c0a92aaeda32fe0ff19ad71843c
Emulation starting at 0x414f19
Section hopping detected into sect_0! Address: 0x401000
Totalsize:0x9f000, VirtualMemorySize:0x9f000
Allocated Chunks:
Setting unpacked Entry Point
OEP:0x1000
Fixing Imports...
ptr_iat: 0x421bc
writing dllname WININET.dll#0 to: 0x9116e
patch_addr: 0x92370, 0x92381, 0x92394, 0x923a7, 0x923ba, 0x923cd, 0x923e3
ptr_iat: 0x42360
Fixing sections
Size of raw data (): 0x14200, fixed: 0x8d000
Size of raw data (petite): 0x308, fixed: 0x11000
Set IAT-Directory to 0 (VA and Size)
RVA to import table: 0x91000
Totalsize:0x9f000, VirtualMemorySize:0x9f000, Allocated chunks: []
Fixing SizeOfImage...
Fixing Memory Protection of Sections

Fixing protections for:  with (True, True, True)
petite
Fixing protections for: petite with (True, True, True)
Fixing Checksum
Dumping state to ./unpacked_602c9b7452e4618580c0c3cd43e3a2c56b438c0a92aaeda32fe0ff19ad71843c

Emulation of 602c9b7452e4618580c0c3cd43e3a2c56b438c0a92aaeda32fe0ff19ad71843c finished.
--- Saved to ./unpacked_602c9b7452e4618580c0c3cd43e3a2c56b438c0a92aaeda32fe0ff19ad71843c ---
```

The good news!


```
"score": "6.033966452662256",
"machine_learning": {
  "classification": "malicious",
  "probability": "96.19 %"
},
"mitre_attck": [
  {
    "id": "T1040",
    "description": "Network Hop"
  },
  {
    "id": "T1001",
    "description": "Data Obfuscation"
  },
  {
    "id": "T1204",
    "description": "User Execution"
  },
  {
    "id": "T1129",
    "description": "Execution through Module Load"
  },
  {
    "id": "T1223",
    "description": "Compile After Delivery"
  },
  {
    "id": "T1218",
    "description": "Reflective Loading"
  }
],
"suspicious_strings": [],
"suspicious_strings_decoded": [],
"high_entropy_sections": [
  ""
],
"import_ratio": "0.2631578947368421",
"suspicious_section_names": [
  "",
  "petite"
],
"suspicious_imports": [
  [
    "kernel32.dll",
    "GetProcAddress"
```

```
},
"mitre_attck": [
  {
    "id": "T1023",
    "description": "Command Line Interface"
  },
  {
    "id": "T1005",
    "description": "Data from Local System"
  },
  {
    "id": "T1008",
    "description": "Fallback Channels"
  },
  {
    "id": "T1064",
    "description": "Scripting"
  },
  {
    "id": "T1040",
    "description": "Network Hop"
  },
  {
    "id": "T1053",
    "description": "Scheduled Task"
  },
  {
    "id": "T1001",
    "description": "Data Obfuscation"
  },
  {
    "id": "T1074",
    "description": "Data Staged"
  },
  {
    "id": "T1020",
    "description": "Automated Exfiltration"
  },
  {
    "id": "T1095",
    "description": "Standard Non-Application Layer Protocol"
  },
  {
    "id": "T1119",
    "description": "Automated Collection"
```

The good news!


```
"score": "6.033966452662256",
"machine_learning": {
  "classification": "malicious",
  "probability": "96.19 %"
},
"mitre_attck": [
  {
    "id": "T1040",
    "description": "Network Hop"
  },
  {
    "id": "T1001",
```

```
},
"mitre_attck": [
  {
    "id": "T1023",
    "description": "Command Line Interface"
  },
  {
    "id": "T1005",
    "description": "Data from Local System"
  },
  {
    "id": "T1008",
```

```
],
"suspicious_strings": [
  "http://i2.tietuku.com/8975c2a506763d03.jpg",
  "6http://ns.adobe.com/xap/1.0/",
  "\" id=\"W5M0MpCehiHzreSzNTczkc9d\"?> <x:xmpmeta xmlns:x=\"adobe:ns:meta/\" x:xmptk=\"Adobe XMP Core 4.1-c036 46.276720, Mon Feb 19 2007 22:40:08        \"> <rdf:RDF xmlns:rdf=\"http://www.w3.org/1999/02/22-rdf-synta
x-ns#\"> <rdf:Description rdf:about=\"\" xmlns:xap=\"http://ns.adobe.com/xap/1.0/\" xmlns:dc=\"http://purl.org/dc/elements/1.1/\" xmlns:photoshop=\"http://ns.adobe.com/photoshop/1.0/\" xmlns:xapMM=\"http://ns.adobe.com/xap
/1.0/mm/\" xmlns:tiff=\"http://ns.adobe.com/tiff/1.0/\" xmlns:exif=\"http://ns.adobe.com/exif/1.0/\" xap:CreateDate=\"2014-11-09T19:51:16+08:00\" xap:ModifyDate=\"2014-11-09T19:51:16+08:00\" xap:MetadataDate=\"2014-11-09T1
9:51:16+08:00\" xap:CreatorTool=\"Adobe Photoshop CS3 Windows\" dc:format=\"image/jpeg\" photoshop:ColorMode=\"3\" photoshop:ICCProfile=\"sRGB IEC61966-2.1\" photoshop:History=\"\" xapMM:InstanceID=\"uuid:5D3521A60268E4119
EB199457098283D\" xapMM:DocumentID=\"uuid:5C3521A60268E4119EB199457098283D\" tiff:Orientation=\"1\" tiff:XResolution=\"720000/10000\" tiff:YResolution=\"720000/10000\" tiff:ResolutionUnit=\"2\" tiff:NativeDigest=\"256,257,
258,259,262,274,277,284,530,531,282,283,296,301,318,319,529,532,306,270,271,272,305,315,33432;4EA5999BA9AB948A9AC4AC83520F026A\" exif:PixelXDimension=\"99\" exif:PixelYDimension=\"16\" exif:ColorSpace=\"1\" exif:NativeDige
st=\"36864,40960,40961,37121,37122,40962,40963,37510,40964,36867,36868,33434,33437,34850,34852,34855,34856,37377,37378,37379,37380,37381,37382,37383,37384,37385,37386,37396,41483,41484,41486,41487,41488,41492,41493,41495,4
1728,41729,41730,41985,41986,41987,41988,41989,41990,41991,41992,41993,41994,41995,41996,42016,0,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20,22,23,24,25,26,27,28,30;2A6F90FB93AF569A4E8E18FEAC05D070\"> <xapMM:DerivedFrom rd
f:parseType=\"Resource\"/> </rdf:Description> </rdf:RDF> </x:xmpmeta>
```

```
],
"suspicious_strings": [],
"suspicious_strings_decoded": [],
"high_entropy_sections": [
  ""
],
"import_ratio": "0.2631578947368421",
"suspicious_section_names": [
  "",
  "petite"
],
"suspicious_imports": [
  [
    "kernel32.dll",
    "GetProcAddress"
```

```
{
  "id": "T1074",
  "description": "Data Staged"
},
{
  "id": "T1020",
  "description": "Automated Exfiltration"
},
{
  "id": "T1095",
  "description": "Standard Non-Application Layer Protocol"
},
{
  "id": "T1119",
  "description": "Automated Collection"
```

RISK TO RESILIENCE
– WORLD TOUR –

TREND MICRO

Now time to hunt threats!

Let’s have a look IRL.

The good news.

”The pattern matching swiss knife for malware researchers (and everyone else)”

in short.. grep on crack…

http://virustotal.github.io/yara/

The basics of Yara.

- Uses descriptions a.k.a rules.
- Aid’s in finding badness in a binary.
- Scan’s files and memory
- Pattern matching
  - Strings
  - Regex
  - Binary patterns {Hex strings}

The basics of Yara.

- Example of a YARA rule to hunt for Cobalt beacons.

rules.yar

```
rule WIN64_MAL_BKDR_COBEACON {
    meta:
        description = "Detects presence of Cobaltstrike beacon."
        author = "Jesper Mikkelsen"
        reference = "Not provided"
        date = "2023-08-07"
        sharing = "TLP:WHITE"
        // techniques and mitre_att may not be accurate. As they are based on static analyis.
        // For higher accuracy, Dynamic analysis is needed.
        techniques = "Software Packing:Command Line Interface:Data from Local System:Network Service Scanning:System Time Discovery:Scripting:Account
        mitre_att = "T1045:T1023:T1005:T1046:T1124:T1064:T1087:T1033:T1018:T1053:T1021:T1001:T1078:T1049:T1074:T1070:T1035:T1095:T1119:T1204:T1080:T1
        score = 75
        scan_type = "file"
        dname = "Backdoor.Win64.COBEACON.YXDHGZ"
        malseq_sequence_length = "45"
        malseq_sequence_step = "10"
        malseq_sample_similarity = "0.1"
        hash0 = "ee0cca0f3aff863531112053a75d27db0624a82c0f3b109a4423e1be7ec9ea7b"
        hash1 = "d30cd86cf146eb923b912b378add3d6c96aa4bb6b57283426a7fbd7e5b40b77a"
        hash2 = "4c110f8657f1ee577bcd3bcf8317a9d1d548ea7ebb3e5add1a10976deef62e49"
        hash3 = "be9736f5f079f0d19526e01a2599f1279916feb52f8ec4f10e339d7bd96593b0"
    strings:
        $sequence_0 = { 0300 0000 0000 7600 0000 50 8b0a 0001 0008 0303 0000 0000 00840000000000 0000 0100 0803 0200 0000 0000 8f00 0000 800000 0001
        // Opcode Sequence 0:
        // 0x00001000      0300        add eax, dword ptr [rax]
        // 0x00001002      0000        add byte ptr [rax], al
```

RISK TO RESILIENCE
– WORLD TOUR –

TREND MICRO

Threat Hunting using YARA in VisionOne

Let’s have a look IRL.

Trend Vision One™ | Forensics and Analysis *Preview* › YARA-IS-KING

Back

YARA-IS-KING

Run YARA Rules | Run osquery | Collect Evidence | Isolate Endpoint | Remove Endpoint | 2 selected

Endpoint name	Agent GUID	IP address
TM-AD01	0f57e701-4749-3025-0072-db5f9...	192.168.10.110
TM-AD02	d0e7d2d6-1dea-40be-87ba-97b27...	192.168.10.111

Run YARA Task

This task executes custom YARA rules on the specified endpoints.

Selected Endpoint: 2

Target: Process | Enter process name

YARA rules: Select File... File size must not exceed 1 MB. Supported file types: Text files

```
rule APT_CobaltStrike_Beacon_Indicator {
   meta:
      description = "Detects CobaltStrike beacons"
      author = "JPCERT"
      reference = "https://github.com/JPCERTCC/aa-tools/blob/master/cobaltstrikescan.py"
      date = "2018-11-09"
   strings:
      $v1 = { 73 70 72 6F 67 00 }
```

Validate Rules

Description:

Hunting for Cobalt Stike in memory

Create Cancel

We have a match!

Trend Vision One™ | Forensics and Analysis *Preview* › YARA-IS-KING › Query Results

YARA | osquery

Task ID:	Selected endpoints:	Target:	YARA rules:	Description:	Matched results:
IR-LQ-20230901-0000001	2 / 2	Process(All)	View Details	Hunting for Cobalt Stike in memory	10

Search endpoint name

Clear all

- [x] TM-AD01
- [x] TM-AD02

Endpoint name ↑	Matched rule name	Process ID	Process name
TM-AD01	HKTL_CobaltStrike_Beacon_Strings	8352	rundll32.exe
TM-AD01	CobaltStrike_Unmodifed_Beacon_v2	8352	rundll32.exe
TM-AD01	HKTL_CobaltStrike_Beacon_Strings	1728	qwkejqkwjewqkje.exe
TM-AD01	CobaltStrike_Unmodifed_Beacon_v2	1728	qwkejqkwjewqkje.exe
TM-AD01	HKTL_CobaltStrike_Beacon_Strings	2060	coreServiceShell.exe
TM-AD01	CobaltStrike_Unmodifed_Beacon_v2	2060	coreServiceShell.exe
TM-AD02	HKTL_CobaltStrike_Beacon_Strings	176	rundll32.exe
TM-AD02	CobaltStrike_Unmodifed_Beacon_v2	176	rundll32.exe
TM-AD02	HKTL_CobaltStrike_Beacon_Strings	2032	qwkejqkwjewqkje.exe
TM-AD02	CobaltStrike_Unmodifed_Beacon_v2	2032	qwkejqkwjewqkje.exe

RISK TO RESILIENCE
– WORLD TOUR –

TREND MICRO

Automating Adversary Based Threat Hunting

Let’s have a look IRL.

Trend Micro Vision One™ | Campaign Intelligence *Preview*

2023-03-02 01:55 (UTC+00:00) 2 Vonese

Important: This is a "Pre-release" feature and is not considered an official release. Please review the Pre-release Disclaimer before using the feature.

New Threat 2023-02-18

Hive Ransomw...

Type: Cybercrime

Campaign Overview:

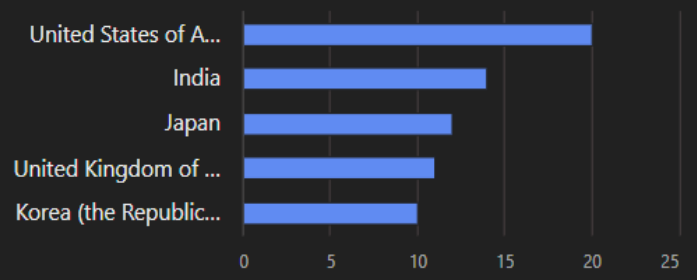
- **Hive** is a ransomware family used in Ransomware-as-a-Service (RaaS) attacks. This ransomware was first found in the wild in **late June** and **early July 2021**....

Learn more

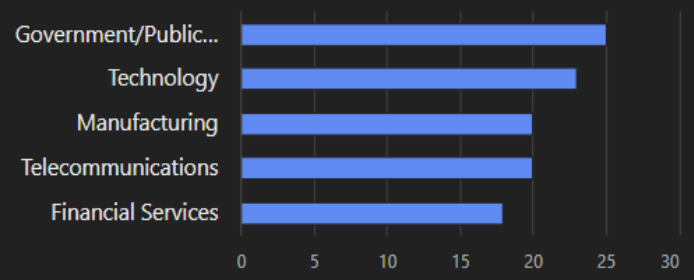
1 / 29

Last seen: Any time | Matched result: All | Type: All | Targeted country/region: All | Targeted industry: All | Search

Top targeted country/region



Top targeted industries



Top matched alerts

Water Goblin 1

Threat	Threat status	AKA	Type	Impact scope	Last seen
Water Goblin	Active	Conti	Cybercrime	0 1	January 2023
Water Mare	Active	REvil, REvix, Sodinokibi, UNC2628	Cybercrime	0 0	February 2023
Water Selkie	Active	Lockbit	Cybercrime	0 0	February 2023
Water Koromodako	Active	BokBot, IcedID	Cybercrime	0 0	February 2023

Important: This is a "Pre-release" feature and is not considered an official release. Please review the Pre-release Disclaimer before using the feature.

Water Goblin

Type	Cybercrime
AKA	Conti
Targeted country/region	-
Targeted industries	Financial Services, Insurance
Motivation	personal-gain
First seen	May 2020
Last seen	January 2023
Last update	2023-02-13 07:54:38

First Observed in May 2020, Conti is a Ransomware as a Service operation and the successor to another Ransomware as a Service, 'RYUK'. The same group behind both Conti and RYUK are also believed to be behind the development TrickBot and BazarLoader malware, both of which act as initial access techniques leading to Conti Ransomware attacks. Trend Micro tracks the activities of Conti affiliates and their associates using the Intrusion Set name 'Water Goblin'.

Noteworthy Attributes

- Continues to successfully compromise businesses globally
- Acquires access via Initial Access Brokers

Intelligence Data

Intelligence Reports (18) | Tactic, Technique, and Procedures | Tools (9) | Malware (12) | CVEs (4) | Indicators (486)

Tactic name or ID, Technique name or ID

Tactic ID	Tactic name	Technique ID	Technique name
TA0007	Discovery	T1135	Network Share Discovery
TA0007	Discovery	T1049	System Network Connections Discovery
TA0007	Discovery	T1482	Domain Trust Discovery
TA0007	Discovery	T1082	System Information Discovery
TA0007	Discovery	T1083	File and Directory Discovery

Impact scope

Workbenches (1) | Servers (0) | Desktops (0)

Workbench ID	Report name	Last sweep ↓
WB-16666-20211210-00007	[Targeted Attack] Top malicious IOC found in company possibly attacked by LockBit - 2023-02-03	2023-02-03 10:55:33

RISK TO RESILIENCE
– WORLD TOUR –

TREND MICRO

IoC and TTP Based Hunting

Intelligence Reports

Trend Micro Vision One™ | Intelligence Reports

Curated | Custom

Trend Micro integrates up-to-the-minute intelligence reports from internal and exte

Last updated: All | View: Matched sweeps only

Matched sweeps	Report name
1 out of 6	Kinsing & Dark.IoT botnet amon 34
3 out of 3	Mixed IOCs, Week 39
3 out of 3	Mixed IOCs, Week 39
3 out of 3	Mixed IOCs, Week 39
1 out of 1	LazarusGroup IOCs, Week 40
3 out of 7	LazarusGroup IOCs, Week 39

Matched: 3 | Total: 7

Created ↓	Type
2022-10-07 13:11:00	Auto Sweeping
2022-10-06 13:45:00	Auto Sweeping

Auto Sweeping Settings

If you turn on **Auto Sweeping** for a source type, Trend Micro Vision One generates a scheduled sweep and runs once every day for 7 consecutive days based on each new report from the source.
Auto Sweeping supports a limited number of indicators per day. For details, check the Online Help.

Source

Source ↓	Auto Sweeping
Trend Micro	ON
EXTERNAL SOURCES	
Corporate security teams	ON
Government agencies - Europe	ON
Government agencies - North America	ON
Information sharing organizations	ON
Security researchers	ON
Security vendors	ON

2023-02-08 23:28 (UTC-05:00) | 99+ | Trend Micro Canada Demo

Auto Sweeping

	Last updated ↓
organizations	2022-10-24 08:00:04
organizations	2022-10-07 14:0
organizations	2022-10-07 14:0
organizations	2022-10-07 14:00:03
organizations	2022-10-07 10:00:03
organizations	2022-09-30 14:00:03

INTELLIGENCE
Download STIX Intelligence Report
Start Sweeping

Start Sweeping

sult -

External Threat Intelligence

Via TAXII and MISP Integrations

Trend Micro Vision One™ | Third-Party Integration › TAXII Feeds

2023-02-08 11:42 (UTC-05:00) 99+ Trend Micro Canada Demo

Third-Party Integration

Note: After successful subscription, TAXII feeds are processed to produce custom intelligence reports. For details on generated reports, go to Intelligence Reports > Custom.

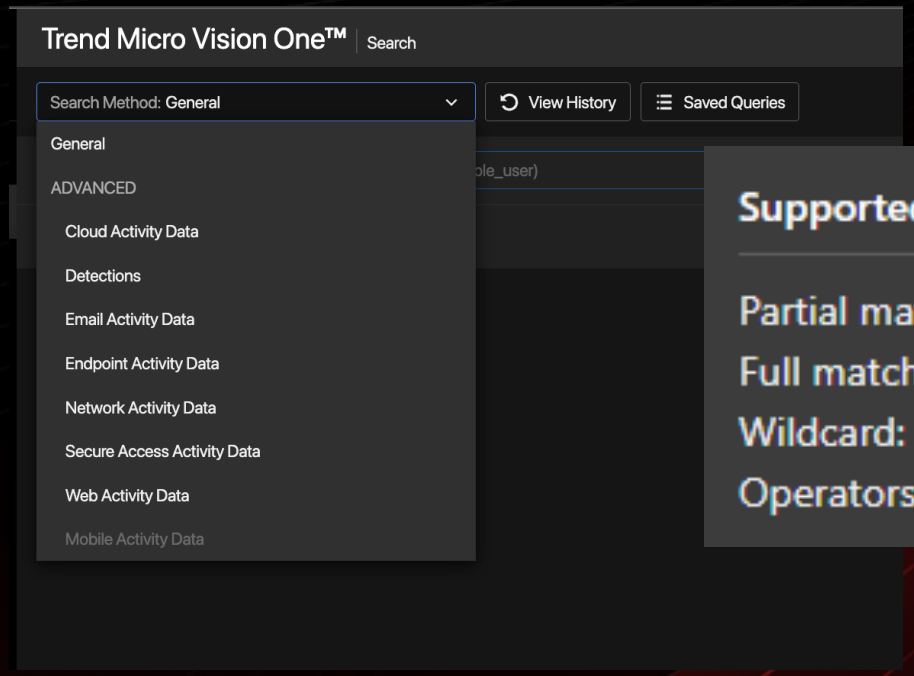
+ Add | TAXII version: All | Feed or collection name

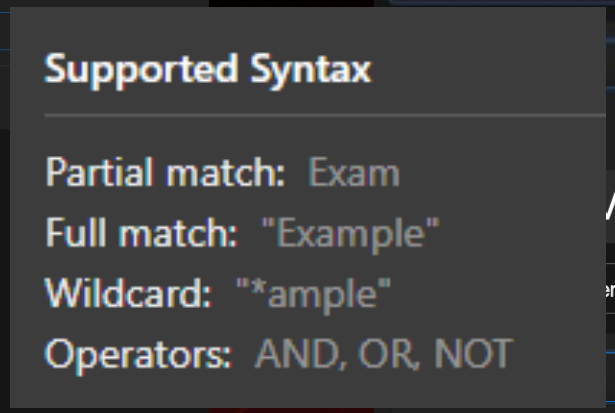
Feed	TAXII version	Description	Polling interval	Last polled	Feed status
9 https://limo.anomali.com/api/v1/taxii2/...	2.0	N/A	Once every 1 hour	2023-02-08 11:52:25	
1 https://otx.alienvault.com/taxii/	2.1	N/A	Once every 1 hour	2023-02-08 11:00:03	

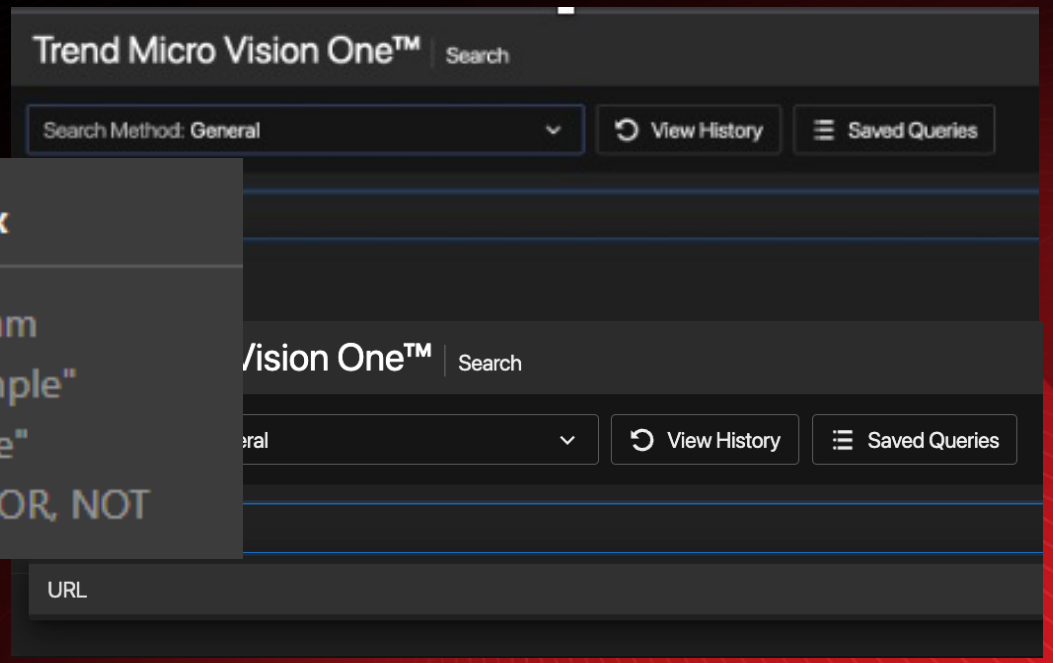
Collection	Collection ID	API root	Last synced	Collection status
Data feed for user: Stevens007	e27b3e3e-13d8-409d-b58d-e5067547a280	Open Threat Exchange TAXII Server	2023-02-08 11:33:08	

IoC based Hunting

In the Search App, you can search for many different objects like IP's, URL, Domains, hashes, registry keys, file names, process names and command lines. But you can also search for partial matches or wildcard, not just Full match which may give you less hits.







Query Language Samples

eventSubId	Data Field Mapping
1	TELEMETRY_PROCESS_OPEN
2	TELEMETRY_PROCESS_CREATE
3	TELEMETRY_PROCESS_TERMINATE
4	TELEMETRY_PROCESS_LOAD_IMAGE
5	TELEMETRY_PROCESS_EXECUTE
6	TELEMETRY_PROCESS_CONNECT
7	TELEMETRY_PROCESS_TRACME
101	TELEMETRY_FILE_CREATE
102	TELEMETRY_FILE_OPEN
103	TELEMETRY_FILE_DELETE
104	TELEMETRY_FILE_SET_SECURITY
105	TELEMETRY_FILE_COPY
106	TELEMETRY_FILE_MOVE

Tip 1 - Partial search
For partial searches, only full words are supported.

Query	Result for "eddie_chen"
endpointHostName: edd	No
endpointHostName: eddie	Yes

Tip 2 - Asterisk wildcards
Use asterisk (*) wildcards to represent single characters.

Query	Result for "eddie_chen"
endpointHostName: edd*	No
endpointHostName: "edd**"	Yes

Tip 3 - Trigger REGEX search
Type an asterisk (*) in the middle of the search string to search for all results containing the search string.

Query	Result for "eddie" and "edison"
endpointHostName: "e*d"	Yes
endpointHostName: "ed**"	Yes
endpointHostName: "**ed"	Yes

TTP Based Hunting

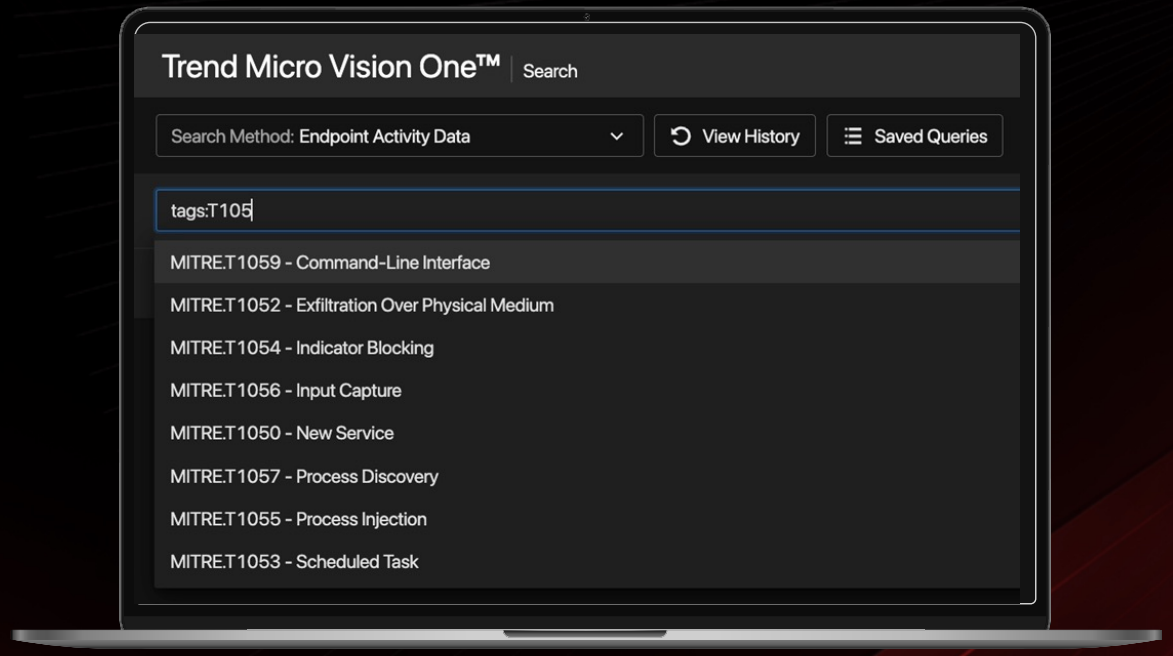
Detection of malicious activity though signature or indicators can easily be defeated by changing attributes like IP's, domains, URL's, hashes.

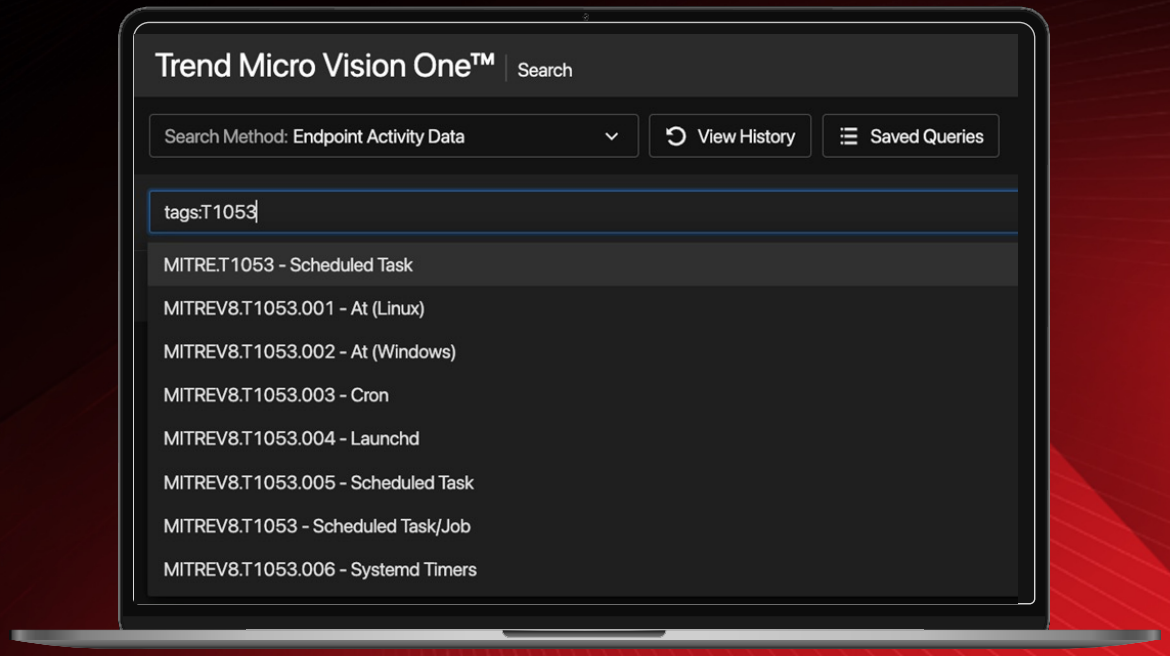
TTP's on the other end don't change as fast as indicators. Threat Actors, Campaign & malware variants share the same TTP's & therefore can be fingerprinted & hunted.

How to get TTPs (MITRE ATT&CK Techniques)

Search App query based on tags : Txxx

Techniques Sub Techniques



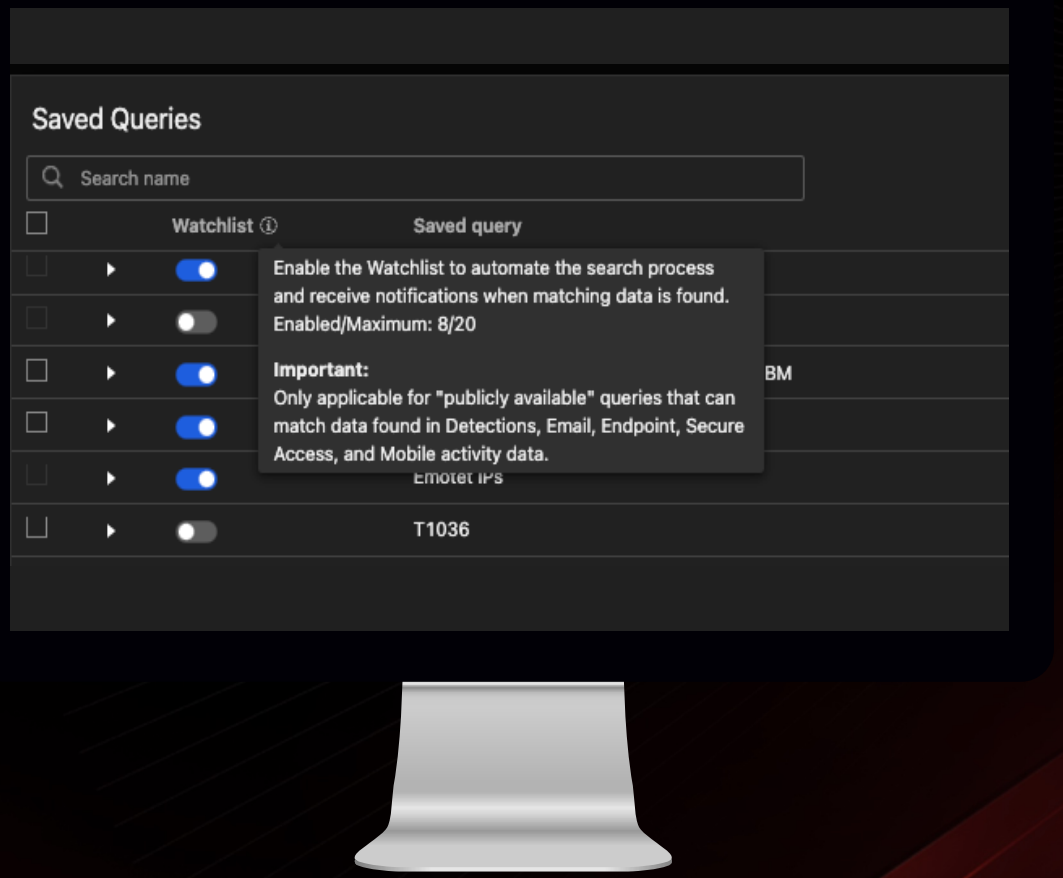


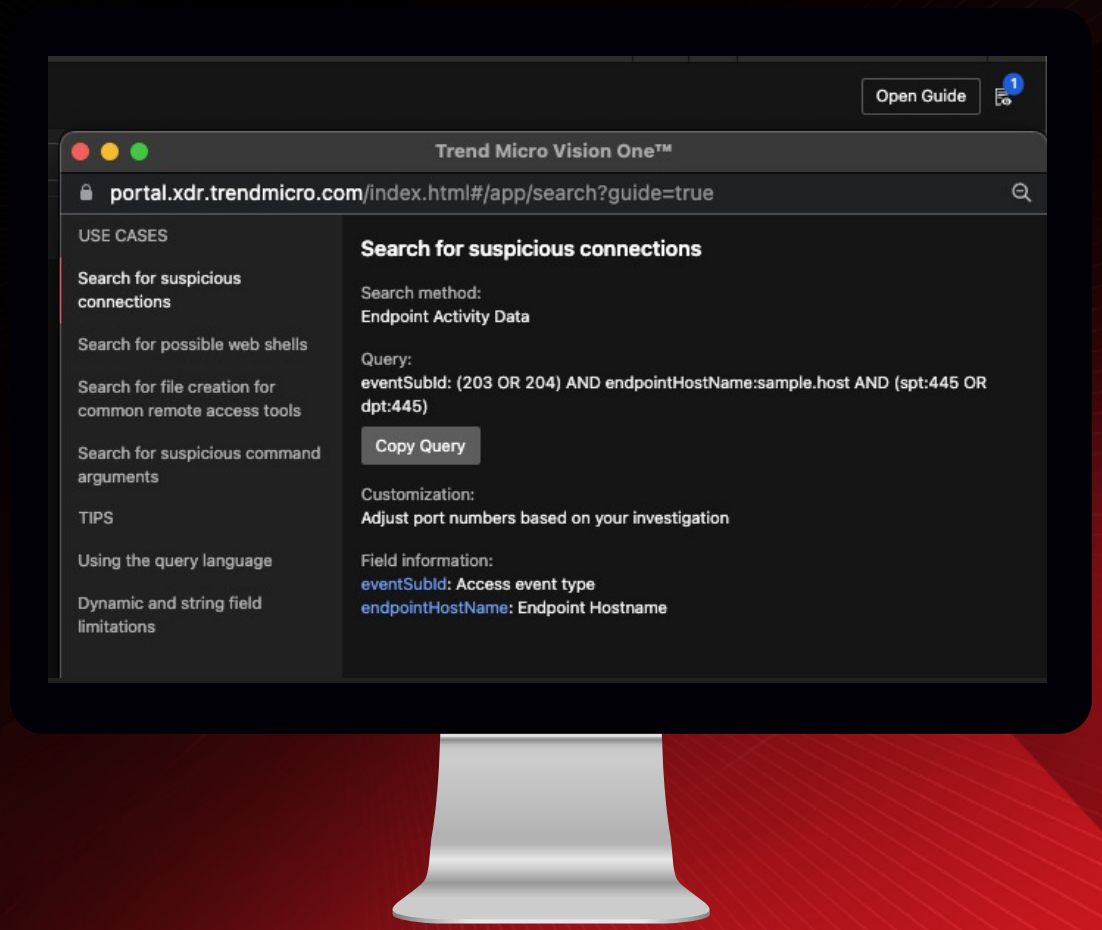
RISK TO RESILIENCE
– WORLD TOUR –

TREND MICRO

What other automations for Threat Hunting are available?

Search App - Watchlist





Automate the Hunt with APIs and XDR Detection Models

XDR Automation Center (API)

- https://automation.trendmicro.com/xdr/home

Locate v3.0 Public API under API Reference

mode | Home | Guides | API Reference | API Cookbook | Change Log

As of May 2022, the Trend Micro Vision One public APIs v1.0 are deprecated. All v1.0 APIs will be removed permanently in April 2023. Trend Micro recommends migrating to the latest version of the APIs.

RESPONSE ACTIONS
- Common
- Custom Script
- Domain Account
- Email
- Endpoint

Parameter that allows you to select the type of data the query displays. Supported values:

- 'default' - Displays the data returned by the query
- 'countOnly' - Displays the number of records returned by the query

HEADER PARAMETERS

TMV1-Query
required

string
Example: `dpt:443 or src:"192.169.1.1"`
Statement that allows you to retrieve a subset of the collected endpoint activity data. Supported fields:

- 'dpt' - Match type support: Full match

Example
ase-endpointActivityDatav3

Copy | Expand all | Collapse all


```
{
  "nextLink": "https://api.xdr.trendmicro.com/v3.0/endpointAct
  "progressRate": 30,
- "items": [
    + { … }
  ]
}
```


Trend Micro Vision One™ | Detection Model Management

2023-02-08 23:28 (UTC-05:00) | 99+ | Trend Micro Canada Demo

Detection Models | Exceptions

Severity: All | Applicable products: All products | Status: All | Last updated: All | Hunt | Reset

Severity	Model	Description	Applicable products	Last updated ↓	Status
Medium	[Threat Hunting] Suspicious script in scheduled task	Persistence of script as scheduled task	Trend Micro Apex One as a Service, Trend Micro Cloud One - Endpoint & Workload Security, Endpoint Sensor, Trend Micro Deep Security Software	2022-10-16 23:14:13	
Low	[Threat Hunting] BITSAdmin Job Execution	Detect suspicious BITS Job execution	Trend Micro Apex One as a Service, Trend Micro Cloud One - Endpoint & Workload Security, Endpoint Sensor, Trend Micro Deep Security Software	2022-09-06 04:17:44	

Jesper Mikkelsen

Technical Director Nordics.