A Deep Dive into the Digital Weapons of North Korean Cyber Army

Ashley_Shen@HITB
Moonbeom_Park@HITB

[RELEASE VERSION]
cat Ashley_Shen

# Senior Threat Analyst

> TeamT5 Inc. Taiwan
> HITCON GIRLS
> Black Hat Asia Review Board
> Tracking APT Attacks & Actors
cat Moobeom_Park

# Deputy General Researcher

> TTPA, South Korea
> BoB
> Speaker of TROOPERS, HITCON, Ekoparty, VXCON
> Tracking NK APT group
AGENDA

# Why this talk?
# Related Work
# The Legos, Malwares and Attack Cases
# The Exploit and Attack Cases
# Takeaways
# Q&A
>> Why this talk?
Why this talk?

# Reconnaissance General bureau (RGB) (revealed in 2009)
# Cyber intelligence operations
# Cyber attacks is a long-term mission
Terrible fact

# Difference in APT Kill Chain

Reconnaissance

Weaponized

Deliver
The instability of international relations
AGENDA

# Why this talk?
# Related Work
# The Legos, Malwares and Attack Cases
# The Exploit and Attack Cases
# Takeaways
# Q&A
Related Work

# 2013

> Operation Troy – cyber espionage and DDOS attacks (MaAfee)

# 2016

> Operation Blockbuster – Lazerus group (Novetta)
> From Seoul to Sony (BlueCoat)

# 2017

> Lazarus under the hood – Bluenoroff group (Kaspersky)
> Campaign Rifle: Andariel, the Maiden of Anguish (Korea FSI)
<table>
<thead>
<tr>
<th>Group</th>
<th>Lazarus</th>
<th>Bluenoroff</th>
<th>Andariel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Targeted Industry</td>
<td>Domestic government, finance, broadcasting</td>
<td>Global and domestic financial institutes</td>
<td>Domestic financial institutes, SMB IT companies and large corporations. Defense industry</td>
</tr>
<tr>
<td>Purpose</td>
<td>Social chaos</td>
<td>Financial profit motivation</td>
<td>Information gathering and profit</td>
</tr>
<tr>
<td>Historical major incidents</td>
<td>• 2013 320 DarkSeoul</td>
<td>• 2015-2016 SWIFT banking attack</td>
<td>• 2016 Attack on cyber command center</td>
</tr>
<tr>
<td></td>
<td>• 2014 Sony Picture Entertainment breach</td>
<td>• 2017 Polish bank</td>
<td>• 2017 South Korea ATM breach</td>
</tr>
<tr>
<td></td>
<td>• 2017 WannaCry ransomware</td>
<td>• 2017 South Korea Bitcoin companies</td>
<td></td>
</tr>
</tbody>
</table>
AGENDA

# Why this talk?
# Related Work
# The Legos, Malwares and Attack Cases
# The Exploit and Attack Cases
# Takeaways
# Q&A
Delivery Method

# Software vulnerabilities
  > Developing 0 day of specific software

# Watering hole attack
  > Deploying exploit on compromised website to spread payload

# Spear-phishing email
  > Attaching malicious document in the spear-phishing email to infect targets
Adopting Social Engineering in the attacks

# Case: 2016 Compromised Online Shopping Site in South Korea

- Phishing attack
- Employee’s Cloud Drive Credential
- Spear-phishing email with malware-laden family’s photo screen saver program
- Trojan Alphanc!

Attacker

Employee of Shopping Site

Target’s Cloud Drive

Our Family

Our Family's Cloud Drive Credential

Family Pictures
Incorporating OpenSSL library into the file, causing large file size (about 900M)

Supporting the following commands:

<table>
<thead>
<tr>
<th>Command Code</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2F24BB19A401D</td>
<td>Gather victim’s information and transmit to C&amp;C</td>
</tr>
<tr>
<td>E8AFAB73D2BE55</td>
<td>Load specific DLL and call function for export</td>
</tr>
<tr>
<td>C7D3D97AE85AC1</td>
<td>Delete itself to ielowutil.exe</td>
</tr>
<tr>
<td>03AAEFA36E0646</td>
<td>Gather specific files in My Documents and transmit to C&amp;C</td>
</tr>
<tr>
<td>E2CE1DAA84A3B1</td>
<td>Detect to virtual mode(Environment)</td>
</tr>
<tr>
<td>2486C09D576ADA</td>
<td>Gather active process information and transmit to C&amp;C</td>
</tr>
<tr>
<td>4462929641CD6F</td>
<td>Gather Windows OS information and transmit to C&amp;C</td>
</tr>
<tr>
<td>653E648F2B3003</td>
<td>Download data of iehmmap.dll from other server</td>
</tr>
<tr>
<td>861A3688159498</td>
<td>Create iehmmap.dll and load it, call function for export</td>
</tr>
<tr>
<td>A6F60781FEF72C</td>
<td>End</td>
</tr>
</tbody>
</table>
> Trojan Alphant

# Three C&C servers in configuration

| MOV DWORD PTR SS:[ESP+0x30],0x7D7CB9BE | 1st IP | 1st Port |
| MOV WORD PTR SS:[ESP+0x2E],0x6BB01 | 2nd IP | 2nd Port |
| MOV DWORD PTR SS:[ESP+0x40],0x6EBF84DC | 3rd IP | 3rd Port |
| MOV WORD PTR SS:[ESP+0x34],0x6BB01 | |

C&C Server IP
- 190.185.124.15 (Port 443)
- 220.132.191.110 (Port 443)
- 202.137.244.198 (Port 443)

Country
- Honduras
- Taiwan
- New Zealand

# 2017 Linked to WannaCry Ransomware by Symantec

Sample C&C server

2014 SPE Breach

2016 Online Shopping Site Breach

Trojan Alphant

Download

2017 WannaCry Ransomware
The Malwares

# Using both customized version of public available malware and self-developed malware.

# Reuse shared code (lego) code heavily.

> Shared code are reused among different groups
> One of the keys to recognize attacks from DPRK
  • Very difficult to correlate with C&C infrastructure
> We called these shared code “legos”!
The Malwares and Attack Cases

Lego1: Multi_Keys_xor Function

```
LOBYTE(_CL) = 0x82u;
v13 = v3;
v6 = 5;
_EDX = 0x556F9482;
_EAX = 0xAF6C1205B;
if ( dwSize > 0 )
{
  v8 = (a1 - out);
  v12 = dwSize;
  do
    *
out = v6 ^ _EAX ^ (_CL ^ *(out + v8));
  v6 = v6 & _EAX ^ _CL & (v6 ^ _EAX);
  _CL = (((_EDX ^ (8 * _EDX)) & 0x7F8) << 20) | (_EDX >> 8);
  _EAX = (((_EAX << 7) ^ (_EAX ^ 0x10 * (_EAX ^ 2 * _EAX)) & 0xFFFFFFFF) << 17) | (_EAX >> 8);
  out = out + 1;
  v9 = v12-- == 1;
  _EDX = (((_EDX ^ (8 * _EDX)) & 0x7F8) << 20) | (_EDX >> 8);
} while ( !v9 );
out = v13;
v2 = dwSize;
memcpy(a1, out, v2);
return VirtualFree(out, 0, 0x8000u);
```
Multi_Keys_xor Decode Function

# Frequently used for decode strings and APIs

```
u0 = Base64 Decode(16, "lNhC1sYUQ/B9235n", &dwSize);
XOR Transform(u0, dwSize);
Kernal32.dll = LoadLibraryA(u0);
free(u0);
if ( Kernal32.dll )
{
    dwSize = 0;
    u2 = Base64 Decode(28, "mHna6Z2zX6cgzFpuyho-", &dwSize);
    XOR Transform(u2, dwSize);
    *GetStartupInfoA _0 = GetProcAddress(Kernal32.dll, u2);
    free(u2);
    dwSize = 0;
    u3 = Base64 Decode(16, "19HPywqKFaMn2g==", &dwSize);
    XOR Transform(u3, dwSize);
    dword_416D34 = GetProcAddress(Kernal32.dll, u3);
    free(u3);
    dwSize = 0;
    u4 = Base64 Decode(16, "mszHzmxK6E2z6E="", &dwSize);
    XOR Transform(u4, dwSize);
    *ExitProcess _0 = GetProcAddress(Kernal32.dll, u4);
    free(u4);
    dwSize = 0;
    u5 = Base64 Decode(28, "mHna6zuX6cgzFpuyho-", &dwSize);
    XOR Transform(u5, dwSize);
    *GetProcessHeap _0 = GetProcAddress(Kernal32.dll, u5);
    free(u5);
    dwSize = 0;
```
The Malwares and Attack Cases

Lego2: FE_XOR Function

```
16  v7 = result;
17  if ( a3 )
18  {
19      do
20      {
21          *(BYTE *)(u6 + a4) ^= u4 ^ (unsigned __int8)result;
22          v8 = v5 >> 8;
23          v4 = BYTE3(result) ^ BYTE1(result) & ((unsigned int)result >> 16) ^ v5 & BYTE1(v5) & (v5 >> 16) ^ v10 & result;
24          result = ((unsigned int)result >> 8) | (v9 << 24);
25          v5 = (v5 >> 8) | (((v7 ^ (unsigned __int16)(2 * v7)) & 0x1FE) << 22);
26          ++v6;
27          v10 = v4;
28          v9 = v8 | (((v7 ^ (unsigned __int16)(2 * v7)) & 0x1FE) << 22);
29          v7 = result;
30      }
31      while ( v6 < a3 );
32  }
33  return result;
34```

Encode every strings and loads dynamically
The Malwares and Attack Cases

# Lego3: TABLE_LOOKUP_Decode Function

```c
encode_string_length = str_length;
encode_string = Encoded_String;
selector = 0;
v6 = 0;
result = malloc(str_length + 1);
if ( result )
{
    for ( target_str = *(BYTE *)encode_string; *(BYTE *)encode_string; target_str = *(BYTE *)encode_string )
    {
        v9 = encode_string_length;
        encode_string = (signed int *)((char *)encode_string + 1);
        --encode_string_length;
        if ( v9 <= 0 || target_str == ' ' )
            break;
        if ( target_str == ' ' )
            target_str = '*';
        matched_value = Table[target_str];
    }
}
```
Lego3: TABLE_LOOKUP_Decode Function

Table

Table = [0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF
The Malwares and Attack Cases

# Lego3: TABLE_LOOKUP_DECODE Function

```c
matched_value = Table[target_str];
if ( matched_value >= 0 )
{
    switch ( selector % 4 )
    {
        case 0:
            *((_BYTE *)result + u6) = 4 * matched_value;
            break;
        case 1:
            *((_BYTE *)result + u6++) |= matched_value >> 4;
            *((_BYTE *)result + u6) = 16 * matched_value;
            break;
        case 2:
            *((_BYTE *)result + u6++) |= matched_value >> 2;
            *((_BYTE *)result + u6) = (BYTE)matched_value << 6;
            break;
        case 3:
            *((_BYTE *)result + u6++) |= matched_value;
            break;
        default:
            break;
    }
    ++selector;
}
```

**Input String:**
"mszHzxmKH6E2zGE="

**Output String:**
ExitProcess

**Input String:**
"mNHa6zuXE6cgzFpuyho=""

**Output String:**
GetProcessHeap
The Malwares and Attack Cases

Lego4: S_Hat_DECODE Function

```c
sprintf(&v9, "%s", "K\A");
memset(byte_4448D0, 0, 0xBB7u);
if ( *(_BYTE *)v1 != 'S' || *(_BYTE *)v1 + 1 != '^' )
{
    v4 = *(_WORD *)v1;
    if ( *(_WORD *)v1 > 0xBB7u )
        v4 = 2999;
    if ( (unsigned __int16)v4 > 0u )
    {
        v5 = (int)((char *)v1 + 2);
        v6 = (unsigned __int16)v4;
        v7 = BYTE1(v10);
        v8 = (int)byte_4448D0;
        do
        {
            *(BYTE *)v8++ = v7 ^ *(BYTE *)v5++;
            --v6;
        } while ( v6 );
    }
    result = (int)byte_4448D0;
}
```
# OK... so how do we going to do with these legos?
Stories of Three incidents

GHOSTRAT
- 2016
- Mar

DESERTWOLF
- 2016
- Aug

VANATM
- 2017
- Mar

Operation named by FSI Rifle Campaign Report
Removed from Release version
The Malwares

# Gh0st Variant
# Rifdoor (on C&C)
# Hacking Tool for DRM A
# Hacking Tool for DRM B
# Gh0st Origi Variant

```assembly
mov [ebp+var_1C8], '0'
mov [ebp+var_1BF], 'r'
mov [ebp+var_1BE], 'i'
mov [ebp+var_1BD], 'g'
mov [ebp+var_1BC], 'i'
```
RifDoor (aka Rifle)

# Coined Rifle because of the pdb string
E:\Data\My Projects\Troy Source Code\tcp1st\rifle\Release\rifle.pdb

# A simple backdoor

# Encode string with xor 0F

# Support commands

> $downloadexec (download sec.exe)
> $internal (sleep)
> $download (download file)
> $exec (execute command)
if ( !strcmpA(*LPCSTR *)command, "k`xac`nkjwj1") ) // $downloadexec
{
    if ( Downloadexec(*LPCSTR *)(command + 4), *(LPCSTR *)(command + 8)) ) // Download sec.exe
    {
        lstrcpyA(&String, "K`xac`nk/|z1ljl|"); // Download Success
        v21 = Xor_F((int)&String);
        lstrcatA((LPSTR)v21, "\r\n");
        if ( sub_401F10() )
        {
            lstrcpyA(&String1, "Jwjlz{f`a/|z1ljl|}"; // Execution Success
            v23 = Xor_F((int)&String1);
            lstrcpyA(&String, (LPCSTR)v23);
        }
        else
        {
            lstrcpyA(&String1, "Jwjlz{f`a/infczj}j"; // Execution failure
            v22 = Xor_F((int)&String1);
            lstrcatA(&String, (LPCSTR)v22);
        }
        lstrcatA(&String, "\r\n");
        v32 = (int)&String;
        v20 = strlenA(&String);
    }
South Korean Ministry of National Defense (Cyber Command) announced that North Korean infiltrated a military network.

- 3200 hosts were compromised, 700 military intranet.
- 39 samples collected, 20 confirmed linked to NK groups.
- Cyber command announced that found Shenyang IP address.
The Malwares

- Type A Backdoor
- Type B (Phandoor)
- Type C Backdoor
- Keylogger A
Phandoor

# Loading API dynamically with Lego4

```c
v122 = 0;
v121 = 0;
v120 = 0;
v0 = sub_401D40("\^Kernel32.dll");
v1 = LoadLibraryA((LPCSTR)v0);
if ( v1 )
{
    v2 = sub_401D40("\^HeapCreate");
dword_413E90 = (int)GetProcAddress(v1, (LPCSTR)v2);
v3 = sub_401D40("\^GetProcessHeap");
dword_413E34 = (int)GetProcAddress(v1, (LPCSTR)v3);
v4 = sub_401D40("\^HeapDestroy");
dword_413E84 = (int)GetProcAddress(v1, (LPCSTR)v4);
v5 = sub_401D40("\^HeapAlloc");
dword_413F34 = (int)GetProcAddress(v1, (LPCSTR)v5);
v6 = sub_401D40("\^HeapReAlloc");
dword_413F4C = (int)GetProcAddress(v1, (LPCSTR)v6);
v7 = sub_401D40("\^HeapFree");
dword_413EA8 = (int)GetProcAddress(v1, (LPCSTR)v7);
v8 = sub_401D40("\^GetModuleFileNameA");
dword_413F24 = (int)GetProcAddress(v1, (LPCSTR)v8);
```
Phandoor

- Upon execution, getting victim IP address with GetAdaptorInfo and encode it.
- Sending encode IP address & MAC address with a special string “Anonymous” to test C&C server connection.
Attacker tailored this trojan for different cases. The supported functions vary across different incidents.

<table>
<thead>
<tr>
<th>Backdoor Command</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Execute Windows command</td>
</tr>
<tr>
<td>9</td>
<td>Get disk information or search file</td>
</tr>
<tr>
<td>10</td>
<td>Find file and send</td>
</tr>
<tr>
<td>11</td>
<td>Receive data and save it to a file</td>
</tr>
<tr>
<td>15</td>
<td>Terminate</td>
</tr>
<tr>
<td>16</td>
<td>Close connection and reconnect</td>
</tr>
<tr>
<td>18</td>
<td>Copy and move nehomegpa.dll to another path</td>
</tr>
<tr>
<td>26</td>
<td>Search process</td>
</tr>
<tr>
<td>49</td>
<td>Search file</td>
</tr>
</tbody>
</table>
Attacker attacks an ATM service provider
Compromised internal network with Antivirus vaccine update server (VMS).
Lateral movement was taken to compromised ATM management server connected with the VMS server.
More than 600 ATM machines were infected with RAT and keylogger
Malware connects to same C2 discovered in DesertWolf case
Removed from Release version
The Malwares

# Rifdoor
# Gh0st
# Hacking Tool (Sniffer)
# Keylogger A
# Trojan D
Links different campaign with Legos

- GHOSTRAT
- DESERTWOLF
- VANATM

- Same C&C server
- Same Keylogger
The Malwares and Attack Cases

# Other TTP on binaries

> PACKERS! PACKERS! PACKERS!

> Love VMP (Feel the pain!!)

> Aspacker, upx, Armadillo v1.71, Themida
The Malwares and Attack Cases

# Other TTP on binaries

- Encode every strings and loads dynamically
- Sometimes encode twice!
DMC Webshell (Proprietary tool)

Removed from Release version
DMC Webshell (Proprietary tool)

Removed from Release version
DMC Webshell (Proprietary tool)

Removed from Release version
DMC Webshell (Proprietary tool)

Removed from Release version
Exploits of malicious documents

# HWP exploit documents

> Hangul Word Processor (HWP) is a proprietary word processing application published by the South Korean company Hancom Inc.

> The most popular word processor in South Korea. (similar to Ichitaro in Japan)

> Attacker deployed HWP exploit documents in attacks targeting Korea individual/organization.
Exploits of malicious documents

# CVE 2013-0808

- EPS Viewer buffer overflow vulnerability

> Trigger by Ghostscript in HWP (Hangul word)
Exploits of malicious documents

# Dropping EPS file with NOP sled and shellcode

# Downloading payload from C&C server

Ghostscript commands
Exploits of malicious documents

# CVE 2017-0621

- EPS restore Use-After-Free
- Applied frequently in recent attacks targeting financial industry in South Korea by Bluenoroff.
  - Targeting a lot of Bitcoin companies recently.
- No alert and error would be trigger during exploitation.
- Triggering in HWP files.
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>Case File</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>CloseFile</td>
<td>2620</td>
</tr>
<tr>
<td>2020.12.20</td>
<td>C:\Users\Admin\AppData\Local\Temp\Apk\WinData\VMD\000009825943.FS</td>
<td>2620</td>
</tr>
</tbody>
</table>
A Recent Sample

Ghost Script

Shellcode

Ghost Script

Embedded PE 32 bits

Embedded PE 64 bits
A Recent Sample

# Trojan Manuscrypt encoded with XOR.

# Manuscrypt packed with VMP.

# Decoded by shellcode and inject into “explorer.exe” process directly (fileless)

Download trojan AsdfDoor (7caa500b60a536d7501e7a6c02408538)

Korean in the Macro script
Exploit for Watering hole attack

# CVE 2016-0189

> Vulnerability works on Internet Explorer 9-11

> Remote execute Javascript

> Compromised website to targeted North Korea defectors
Fun Facts

# Interesting PDB Strings

Removed from Release version
AGENDA

# Why this talk?
# Related Work
# The Malwares and Attack Cases
# The Exploit and Attack Cases
# Takeaways
# Q&A
Takeaways

- We introduce some “legos” codes, exploits and webshell for identify attacks from DPRK cyber army.
- Cyber attacker from DPRK frequently reuse function codes in their attacks.
- We are building a shared code library called “The Legos” project. Encouraging researchers to release the YARA rules of lego functions.
- The legos indicates a share code database or dedicated group responsible for tools development.
- More attacks from Lazarus/Bluenoroff/Andariel are exceptive, be prepared and update to the latest intelligence.
Questions?

ashley_shen_920
krNeoTra
Some Reference

# Reports

> Financial Security Institute – Campaign Rifle: Andariel, the Maiden of Anguish
> https://www.fireeye.com/blog/threat-research/2017/05/eps-processing-zero-days.html