Sponsor with batch-filed whiskers: Ballistic Bobcat's scan and strike backdoor

By Adam Burgher

Archived: 2026-04-06 00:12:32 UTC

ESET researchers discovered a Ballistic Bobcat campaign targeting various entities in Brazil, Israel, and the United Arab Emirates, using a novel backdoor we have named Sponsor.

We discovered Sponsor after we analyzed an interesting sample we detected on a victim's system in Israel in May 2022 and scoped the victim-set by country. Upon examination, it became evident to us that the sample was a novel backdoor deployed by the Ballistic Bobcat APT group.

Ballistic Bobcat, previously tracked by ESET Research as APT35/APT42 (aka Charming Kitten, TA453, or PHOSPHORUS), is a suspected Iran-aligned advanced persistent threat group that targets education, government, and healthcare organizations, as well as human rights activists and journalists. It is most active in Israel, the Middle East, and the United States. Notably, during the pandemic, it was targeting COVID-19-related organizations, including the World Health Organization and Gilead Pharmaceuticals, and medical research personnel.

Overlaps between Ballistic Bobcat campaigns and Sponsor backdoor versions show a fairly clear pattern of tool development and deployment, with narrowly targeted campaigns, each of limited duration. We subsequently discovered four other versions of the Sponsor backdoor. In total, we saw Sponsor deployed to at least 34 victims in Brazil, Israel, and the United Arab Emirates, as outlined in Figure 1.

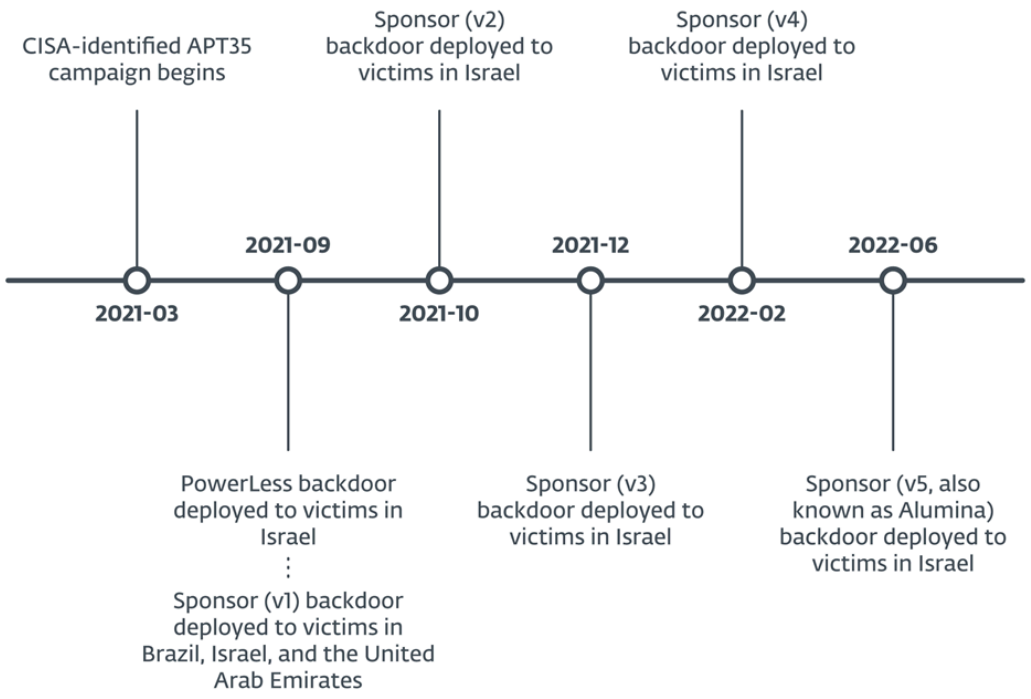

Figure 1. Timeline of the Sponsoring Access campaign

> **Key points of this blogpost:**
>
> - *We discovered a new backdoor deployed by Ballistic Bobcat that we subsequently named Sponsor.*

> - *Ballistic Bobcat deployed the new backdoor in September 2021, while it was wrapping up the campaign documented in CISA Alert AA21-321A and the PowerLess campaign.*
> - *The Sponsor backdoor uses configuration files stored on disk. These files are discreetly deployed by batch files and deliberately designed to appear innocuous, thereby attempting to evade detection by scanning engines.*
> - *Sponsor was deployed to at least 34 victims in Brazil, Israel, and the United Arab Emirates; we have named this activity the Sponsoring Access campaign.*

Initial access

Ballistic Bobcat obtained initial access by exploiting known vulnerabilities in internet-exposed Microsoft Exchange servers by first conducting meticulous scans of the system or network to identify potential weaknesses or vulnerabilities, and subsequently targeting and exploiting those identified weaknesses. The group has been known to engage in this behavior for some time. However, many of the 34 victims identified in ESET telemetry might best be described as victims of opportunity rather than preselected and researched victims, as we suspect Ballistic Bobcat engaged in the above-described scan-and-exploit behavior because it was not the only threat actor with access to these systems. We have named this Ballistic Bobcat activity utilizing the Sponsor backdoor the Sponsoring Access campaign.

The Sponsor backdoor uses configuration files on disk, dropped by batch files, and both are innocuous so as to bypass scanning engines. This modular approach is one that Ballistic Bobcat has used quite often and with modest success in the past two and a half years. On compromised systems, Ballistic Bobcat also continues to use a variety of open-source tools, which we describe – together with the Sponsor backdoor – in this blogpost.

Victimology

Figure 2. Geographical distribution of entities targeted by Ballistic Bobcat with the Sponsor backdoor

A significant majority of the 34 victims were located in Israel, with only two located in other countries:

- Brazil, at a medical cooperative and health insurance operator, and
- the United Arab Emirates, at an unidentified organization.

Table 1 describes the verticals, and organizational details, for victims in Israel.

Table 1. Verticals and organizational details for victims in Israel

Vertical	Details
Automotive	· An automotive company specializing in custom modifications. · An automotive repair and maintenance company.
Communications	· An Israeli media outlet.
Engineering	· A civil engineering firm. · An environmental engineering firm. · An architectural design firm.
Financial services	· A financial services company that specializes in investment counseling. · A company that manages royalties.
Healthcare	· A medical care provider.
Insurance	· An insurance company that operates an insurance marketplace. · A commercial insurance company.
Law	· A firm specializing in medical law.
Manufacturing	· Multiple electronics manufacturing companies. · A company that manufactures metal-based commercial products. · A multinational technology manufacturing company.
Retail	· A food retailer. · A multinational diamond retailer. · A skin care products retailer. · A window treatment retailer and installer. · A global electronic parts supplier. · A physical access control supplier.

Technology	· An IT services technology company. · An IT solutions provider.
Telecommunications	· A telecommunications company.
Unidentified	· Multiple unidentified organizations.

Attribution

In August 2021, the Israeli victim above that operates an insurance marketplace was attacked by Ballistic Bobcat with the tools CISA reported in November 2021. The indicators of compromise we observed are:

- MicrosoftOutlookUpdateSchedule,
- MicrosoftOutlookUpdateSchedule.xml,
- GoogleChangeManagement, and
- GoogleChangeManagement.xml.

Ballistic Bobcat tools communicated with the same command and control (C&C) server as in the CISA report: 162.55.137[.]20.

Then, in September 2021, the same victim received the next generation of Ballistic Bobcat tools: the PowerLess backdoor and its supporting toolset. The indicators of compromise we observed were:

- http://162.55.137[.]20/gsdhdDdfgA5sS/ff/dll.dll,
- windowsprocesses.exe, and
- http://162.55.137[.]20/gsdhdDdfgA5sS/ff/windowsprocesses.exe.

On November 18th, 2021, the group then deployed another tool (Plink) that was covered in the CISA report, as MicrosoftOutLookUpdater.exe. Ten days later, on November 28th, 2021, Ballistic Bobcat deployed the Merlin agent (the agent portion of an open-source post-exploitation C&C server and agent written in Go). On disk, this Merlin agent was named googleUpdate.exe, using the same naming convention as described in the CISA report to hide in plain sight.

The Merlin agent executed a Meterpreter reverse shell that called back to a new C&C server, 37.120.222[.]168:80. On December 12th, 2021, the reverse shell dropped a batch file, install.bat, and within minutes of executing the batch file, Ballistic Bobcat operators pushed their newest backdoor, Sponsor. This would turn out to be the third version of the backdoor.

Technical analysis

Initial access

We were able to identify a likely means of initial access for 23 of the 34 victims that we observed in ESET telemetry. Similar to what was reported in the PowerLess and CISA reports, Ballistic Bobcat probably exploited a known vulnerability, CVE-2021-26855, in Microsoft Exchange servers to gain a foothold on these systems.

For 16 of the 34 victims, it appears Ballistic Bobcat was not the only threat actor with access to their systems. This may indicate, along with the wide variety of victims and the apparent lack of obvious intelligence value of a few victims, that Ballistic Bobcat engaged in scan-and-exploit behavior, as opposed to a targeted campaign against preselected victims.

Toolset

Open-source tools

Ballistic Bobcat employed a number of open-source tools during the Sponsoring Access campaign. Those tools and their functions are listed in Table 2.

Table 2. Open-source tools used by Ballistic Bobcat

Filename	Description
host2ip.exe	Maps a hostname to an IP address within the local network.
CSRSS.EXE	RevSocks, a reverse tunnel application.
mi.exe	Mimikatz, with an original filename of midongle.exe and packed with the Armadillo PE packer.
gost.exe	GO Simple Tunnel (GOST), a tunneling application written in Go.
chisel.exe	Chisel, a TCP/UDP tunnel over HTTP using SSH layers.
csrss_protected.exe	RevSocks tunnel, protected with the trial version of the Enigma Protector software protection.
plink.exe	Plink (PuTTY Link), a command line connection tool.

WebBrowserPassView.exe	A password recovery tool for passwords stored in web browsers.
sqlextractor.exe	A tool for interacting with, and extracting data from, SQL databases.
procdump64.exe	ProcDump, a Sysinternals command line utility for monitoring applications and generating crash dumps.

Batch files

Ballistic Bobcat deployed batch files to victims' systems moments before deploying the Sponsor backdoor. File paths we are aware of are:

- C:\inetpub\wwwroot\aspnet_client\Install.bat
- %USERPROFILE%\Desktop\Install.bat
- %WINDOWS%\Tasks\Install.bat

Unfortunately, we were unable to obtain any of these batch files. However, we believe they write innocuous configuration files to disk, which the Sponsor backdoor requires to function fully. These configuration filenames were taken from the Sponsor backdoors but were never collected:

- config.txt
- node.txt
- error.txt
- Uninstall.bat

We believe that the batch files and configuration files are part of the modular development process that Ballistic Bobcat has favored over the past few years.

Sponsor backdoor

Sponsor backdoors are written in C++ with compilation timestamps and Program Database (PDB) paths as shown in Table 3. A note on version numbers: the column Version represents the version that we track internally based on the linear progression of Sponsor backdoors where changes are made from one version to the next. The Internal version column contains the version numbers observed in each Sponsor backdoor and are included for ease of comparison when examining these and other potential Sponsor samples.

Table 3. Sponsor compilation timestamps and PDBs

Version	Internal version	Compilation timestamp	PDB
1	1.0.0	2021-08-29 09:12:51	D:\Temp\BD_Plus_Srvc\Release\BD_Plus_Srvc.pdb

2	1.0.0	2021-10-09 12:39:15	D:\Temp\Sponsor\Release\Sponsor.pdb
3	1.4.0	2021-11-24 11:51:55	D:\Temp\Sponsor\Release\Sponsor.pdb
4	2.1.1	2022-02-19 13:12:07	D:\Temp\Sponsor\Release\Sponsor.pdb
5	1.2.3.0	2022-06-19 14:14:13	D:\Temp\Alumina\Release\Alumina.pdb

The initial execution of Sponsor requires the runtime argument install, without which Sponsor gracefully exits, likely a simple anti-emulation/anti-sandbox technique. If passed that argument, Sponsor creates a service called SystemNetwork (in v1) and Update (in all the other versions). It sets the service's Startup Type to Automatic, and sets it to run its own Sponsor process, and grants it full access. It then starts the service.

Sponsor, now running as a service, attempts to open the aforementioned configuration files previously placed on disk. It looks for config.txt and node.txt, both in the current working directory. If the first is missing, Sponsor sets the service to Stopped and gracefully exits.

Backdoor configuration

Sponsor's configuration, stored in config.txt, contains two fields:

- An update interval, in seconds, to periodically contact the C&C server for commands.
- A list of C&C servers, referred to as relays in Sponsor's binaries.

The C&C servers are stored encrypted (RC4), and the decryption key is present in the first line of config.txt. Each of the fields, including the decryption key, have the format shown in Figure 3.

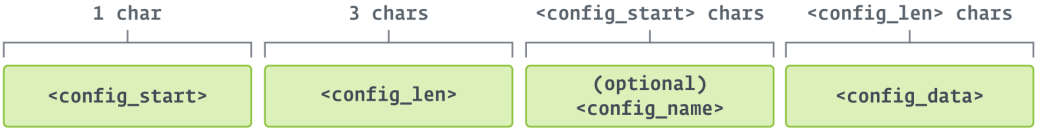

Figure 3. Format of configuration fields in config.txt

These subfields are:

- config_start: indicates the length of config_name, if present, or zero, if not. Used by the backdoor to know where config_data starts.
- config_len: length of config_data.
- config_name: optional, contains a name given to the configuration field.
- config_data: the configuration itself, encrypted (in the case of C&C servers) or not (all the other fields).

Figure 4 shows an example with color-coded contents of a possible config.txt file. Note that this is not an actual file we observed, but a fabricated example.

```
e005decryption_keyMyK33
f003update_interval120
601crelay1510dcb1205c85ce58fb6dedc76cf
6012relay25308d20d07d642f98c
```

Figure 4. Example of possible contents of config.txt

The last two fields in config.txt are encrypted with RC4, using the string representation of the SHA-256 hash of the specified decryption key, as the key to encrypt the data. We see that the encrypted bytes are stored hex-encoded as ASCII text.

Host information gathering

Sponsor gathers information about the host on which it is running, reports all of the gathered information to the C&C server, and receives a node ID, which is written to node.txt. Table 4 lists keys and values in the Windows registry that Sponsor uses to get the information, and provides an example of the data collected.

Table 4. Information gathered by Sponsor

Registry key	Value	Example
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters	Hostname	D-835MK1
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation	TimeZoneKeyName	Israel Standard Time
HKEY_USERS\.DEFAULT\Control Panel\International	LocaleName	he-IL
HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\BIOS	BaseBoardProduct	10NX0010
HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\CentralProcessor\0	ProcessorNameString	Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz

Registry key	Value	Example
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion	ProductName	Windows 1 Enterprise I
	CurrentVersion	6.3
	CurrentBuildNumber	19044
	InstallationType	Client

Sponsor also collects the host's Windows domain by using the following WMIC command:

wmic computersystem get domain

Lastly, Sponsor uses Windows APIs to collect the current username (GetUserNameW), determine if the current Sponsor process is running as a 32- or 64-bit application (GetCurrentProcess, then IsWow64Process(CurrentProcess)), and determines whether the system is running on battery power or connected to an AC or DC power source (GetSystemPowerStatus).

One oddity regarding the 32- or 64-bit application check is that all observed samples of Sponsor were 32-bit. This could mean that some of the next stage tools require this information.

The collected information is sent in a base64-encoded message that, before encoding, starts with r and has the format shown in Figure 5.

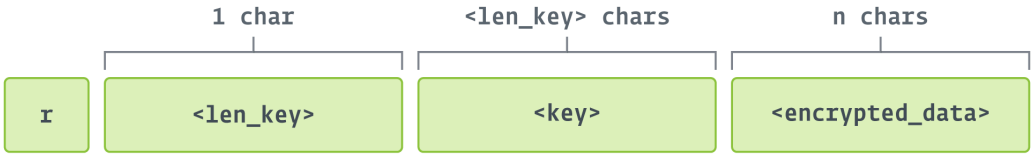

Figure 5. Format of the message sent by Sponsor to register the victimized computer

The information is encrypted with RC4, and the encryption key is a random number generated on the spot. The key is hashed with the MD5 algorithm, not SHA-256 as previously mentioned. This is the case for all communications where Sponsor has to send encrypted data.

The C&C server replies with a number used to identify the victimized computer in later communications, which is written to node.txt. Note that the C&C server is randomly chosen from the list when the r message is sent, and the same server is used in all subsequent communications.

Command processing loop

Sponsor requests commands in a loop, sleeping according to the interval defined in config.txt. The steps are:

1. Send a chk=Test message repeatedly, until the C&C server replies Ok.
2. Send a c (IS_CMD_AVAIL) message to the C&C server, and receive an operator command.
3. Process the command.
    - If there is output to be sent to the C&C server, send an a (ACK) message, including the output (encrypted), or
    - If execution failed, send an f (FAILED) message. The error message is not sent.
4. Sleep.

The c message is sent to request a command to execute, and has the format (before base64 encoding) shown in Figure 6.

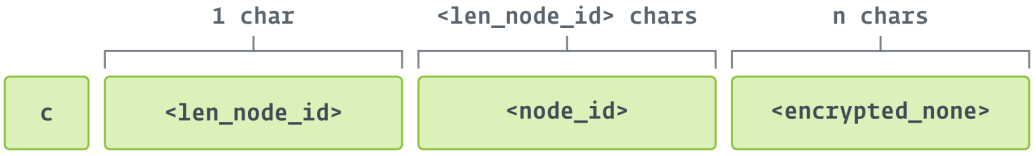

Figure 6. Format of the message sent by Sponsor to ask for commands to execute

The encrypted_none field in the figure is the result of encrypting the hardcoded string None with RC4. The key for encryption is the MD5 hash of node_id.

The URL used to contact the C&C server is built as: http://<IP_or_domain>:80. This may indicate that 37.120.222[.]168:80 is the only C&C server used throughout the Sponsoring Access campaign, as it was the only IP address we observed victim machines reaching out to on port 80.

Operator commands

Operator commands are delineated in Table 5 and appear in the order in which they are found in the code. Communication with the C&C server occurs over port 80.

Table 5. Operator commands and descriptions

Command	Description
p	Sends the process ID for the running Sponsor process.

e	Executes a command, as specified in a subsequent additional argument, on the Sponsor host using the following string: c:\windows\system32\cmd.exe /c <cmd> > \result.txt 2>&1 Results are stored in result.txt in the current working directory. Sends an a message with the encrypted output to the C&C server if successfully executed. If failed, sends an f message (without specifying the error).
d	Receives a file from the C&C server and executes it. This command has many arguments: the target filename to write the file into, the MD5 hash of the file, a directory to write the file to (or the current working directory, by default), a Boolean to indicate whether to run the file or not, and the contents of the executable file, base64-encoded. If no errors occur, an a message is sent to the C&C server with Upload and execute file successfully or Upload file successfully without execute (encrypted). If errors occur during execution of the file, an f message is sent. If the MD5 hash of the contents of the file does not match the provided hash, an e (CRC_ERROR) message is sent to the C&C server (including only the encryption key used, and no other information). The use of the term Upload here is potentially confusing as the Ballistic Bobcat operators and coders take the point of view from the server side, whereas many might view this as a download based on the pulling of the file (i.e., downloading it) by the system using the Sponsor backdoor.
u	Attempts to download a file using the URLDownloadFileW Windows API and execute it. Success sends an a message with the encryption key used, and no other information. Failure sends an f message with a similar structure.
s	Executes a file already on disk, Uninstall.bat in the current working directory, that most likely contains commands to delete files related to the backdoor.
n	This command can be explicitly supplied by an operator or can be inferred by Sponsor as the command to execute in the absence of any other command. Referred to within Sponsor as NO_CMD, it executes a randomized sleep before checking back in with the C&C server.
b	Updates the list of C&Cs stored in config.txt in the current working directory. The new C&C addresses replace the previous ones; they are not added to the list. It sends an a message with New relays replaced successfully (encrypted) to the C&C server if successfully updated.
i	Updates the predetermined check-in interval specified in config.txt. It sends an a message with New interval replaced successfully to the C&C server if successfully updated.

Updates to Sponsor

Ballistic Bobcat coders made code revisions between Sponsor v1 and v2. The two most significant changes in the latter are:

- Optimization of code where several longer functions were minimized into functions and subfunctions, and

- Disguising Sponsor as an updater program by including the following message in the service configuration:

App updates are great for both app users and apps – updates mean that developers are always working on improving the app, keeping in mind a better customer experience with each update.

Network infrastructure

In addition to piggybacking on the C&C infrastructure used in the PowerLess campaign, Ballistic Bobcat also introduced a new C&C server. The group also utilized multiple IPs to store and deliver support tools during the Sponsoring Access campaign. We have confirmed that none of these IPs are in operation at this time.

Conclusion

Ballistic Bobcat continues to operate on a scan-and-exploit model, looking for targets of opportunity with unpatched vulnerabilities in internet-exposed Microsoft Exchange servers. The group continues to use a diverse open-source toolset supplemented with several custom applications, including its Sponsor backdoor. Defenders would be well advised to patch any internet-exposed devices and remain vigilant for new applications popping up within their organizations.

> For any inquiries about our research published on WeLiveSecurity, please contact us at threatintel@eset.com.
> ESET Research offers private APT intelligence reports and data feeds. For any inquiries about this service, visit the ESET Threat Intelligence page.

IoCs

Files

SHA-1	Filename	Detection	Descripti
098B9A6CE722311553E1D8AC5849BA1DC5834C52	N/A	Win32/Agent.UXG	Ballistic Bobcat backdoor, Sponsor (v1).
5AEE3C957056A8640041ABC108D0B8A3D7A02EBD	N/A	Win32/Agent.UXG	Ballistic Bobcat backdoor, Sponsor (v2).
764EB6CA3752576C182FC19CFF3E86C38DD51475	N/A	Win32/Agent.UXG	Ballistic Bobcat backdoor, Sponsor (v3).

2F3EDA9D788A35F4C467B63860E73C3B010529CC	N/A	Win32/Agent.UXG	Ballistic Bobcat backdoor, Sponsor (v4).
E443DC53284537513C00818392E569C79328F56F	N/A	Win32/Agent.UXG	Ballistic Bobcat backdoor, Sponsor (v5, aka Alumina)
C4BC1A5A02F8AC3CF642880DC1FC3B1E46E4DA61	N/A	WinGo/Agent.BT	RevSocks reverse tunnel.
39AE8BA8C5280A09BA638DF4C9D64AC0F3F706B6	N/A	clean	ProcDump a commar line utility for monitorin applicatio and generating crash dumps.
A200BE662CDC0ECE2A2C8FC4DBBC8C574D31848A	N/A	Generik.EYWYQYF	Mimikatz
5D60C8507AC9B840A13FFDF19E3315A3E14DE66A	N/A	WinGo/Riskware.Gost.D	GO Simpl Tunnel (GOST).
50CFB3CF1A0FE5EC2264ACE53F96FADFE99CC617	N/A	WinGo/HackTool.Chisel.A	Chisel reverse tunnel.
1AAE62ACEE3C04A6728F9EDC3756FABD6E342252	N/A	N/A	Host2IP discovery

			tool.
519CA93366F1B1D71052C6CE140F5C80CE885181	N/A	Win64/Packed.Enigma.BV	RevSocks tunnel, protected with the trial versi of the Enigma Protector software protection
4709827C7A95012AB970BF651ED5183083366C79	N/A	N/A	Plink (PuTTY Link), a command line connectio tool.
99C7B5827DF89B4FAFC2B565ABED97C58A3C65B8	N/A	Win32/PSWTool.WebBrowserPassView.I	A passwo recovery tool for password stored in web browsers.
E52AA118A59502790A4DD6625854BD93C0DEAF27	N/A	MSIL/HackTool.SQLDump.A	A tool for interactin with, and extracting data from SQL databases

File paths

The following is a list of paths where the Sponsor backdoor was deployed on victimized machines.

%SYSTEMDRIVE%\inetpub\wwwroot\aspnet_client\

%USERPROFILE%\AppData\Local\Temp\file\

%USERPROFILE%\AppData\Local\Temp\2\low\

%USERPROFILE%\Desktop\

%USERPROFILE%\Downloads\a\

%WINDIR%\

%WINDIR%\INF\MSExchange Delivery DSN\

%WINDIR%\Tasks\

%WINDIR%\Temp\%WINDIR%\Temp\crashpad\1\Files

Network

IP	Provider	First seen	Last seen	Details
162.55.137[.]20	Hetzner Online GMBH	2021-06-14	2021-06-15	PowerLess C&C.
37.120.222[.]168	M247 LTD	2021-11-28	2021-12-12	Sponsor C&C.
198.144.189[.]74	Colocrossing	2021-11-29	2021-11-29	Support tools download site.
5.255.97[.]172	The Infrastructure Group B.V.	2021-09-05	2021-10-28	Support tools download site.

MITRE ATT&CK techniques

This table was built using version 13 of the MITRE ATT&CK framework**.**

Tactic	ID	Name	Description
Reconnaissance	*T1595*	Active Scanning: Vulnerability Scanning	Ballistic Bobcat scans for vulnerable versions of Microsoft Exchange Servers to exploit.
Resource Development	*T1587.001*	Develop Capabilities: Malware	Ballistic Bobcat designed and coded the Sponsor backdoor.
	T1588.002	Obtain Capabilities: Tool	Ballistic Bobcat uses various open-source tools as part of the Sponsoring Access campaign.
Initial Access	*T1190*	Exploit Public-Facing Application	Ballistic Bobcat targets internet-exposed Microsoft Exchange Servers.

Tactic	ID	Name	Description
Execution	*T1059.003*	Command and Scripting Interpreter: Windows Command Shell	The Sponsor backdoor uses the Windows command shell to execute commands on the victim's system.
	T1569.002	System Services: Service Execution	The Sponsor backdoor sets itself as a service and initiates its primary functions after the service is executed.
Persistence	*T1543.003*	Create or Modify System Process: Windows Service	Sponsor maintains persistence by creating a service with automatic startup that executes its primary functions in a loop.
Privilege Escalation	*T1078.003*	Valid Accounts: Local Accounts	Ballistic Bobcat operators attempt to steal credentials of valid users after initially exploiting a system before deploying the Sponsor backdoor.
Defense Evasion	*T1140*	Deobfuscate/Decode Files or Information	Sponsor stores information on disk that is encrypted and obfuscated, and deobfuscates it at runtime.
	T1027	Obfuscated Files or Information	Configuration files that the Sponsor backdoor requires on disk are encrypted and obfuscated.
	T1078.003	Valid Accounts: Local Accounts	Sponsor is executed with admin privileges, likely using credentials that operators found on disk; along with Ballistic Bobcat's innocuous naming conventions, this allows Sponsor to blend into the background.
Credential Access	*T1555.003*	Credentials from Password Stores: Credentials from Web Browsers	Ballistic Bobcat operators use open-source tools to steal credentials from password stores inside web browsers.
Discovery	*T1018*	Remote System Discovery	Ballistic Bobcat uses the Host2IP tool, previously used by Agrius, to discover other systems within reachable networks and correlate their hostnames and IP addresses.

Tactic	ID	Name	Description
Command and Control	*T1001*	Data Obfuscation	The Sponsor backdoor obfuscates data before sending it to the C&C server.

Source: https://www.welivesecurity.com/en/eset-research/sponsor-batch-filed-whiskers-ballistic-bobcats-scan-strike-backdoor/