2018 Sofacy Activity

By GReAT on March 9, 2018. 5:00 pm

Sofacy, also known as APT28, Fancy Bear, and Tsar Team, is a prolific, well resourced, and persistent adversary. They are sometimes portrayed as wild and reckless, but as seen under our visibility, the group can be pragmatic, measured, and agile. Our previous post on their 2017 activity stepped away from the previously covered headline buzz presenting their association with previously known political hacks and interest in Europe and the US, and examines their under-reported ongoing activity in middle east, central asia, and now a shift in targeting further east, including China, along with an overlap surprise. There is much understated activity that can be clustered within this set and overlap in APT activity. Here, we examine current deployment, code, cryptography, and targeting.

Essentially, this examination finds the group maintains subdivisions of efforts in targeting, development, and coding. Comparisons to other modules quickly shows a delineation in other Sofacy efforts. SPLM, GAMEFISH, and Zebrocy delivery all maintain their own clusters, but frequently overlap later.

Because SPLM is their primary and selective second stage tool, SPLM deployment is of much interest. But Zebrocy efforts are in such high volume, that these modules need examination as well.

SPLM/CHOPSTICK/XAgent Code

SPLM, otherwise known as CHOPSTICK, or by the author(s) as "XAgent", is described as Sofacy's signature second stage tool, selectively used for years against around the world. Really, many modified XAgent modules have been deployed over the years. Even the individual Linux modules renamed as "Fysbis" backdoors released in 2016 were merely modified and reduced portions of recompiled XAgent C/C++ codebase. Anyway, SPLM/CHOPSTICK has maintained various combinations of code, with some recognizable functionality listed here.

Source	Modules	Channels
Modules	Keylogger	HTTP
Channels	RemoteShell	FTP
Boot	FileSystem	SMTP
Library	Launcher	
MainHandler	CameraShot	

InjectApp
Screenshot
FileObserver
PasswordFirefox
InfoOS

Version 3 and early version 4 SPLM modules maintained keylogger, remoteshell, and filestealer code all within the larger compiled backdoor, and executed each body of functionality from within that process memory space. Later v4 SPLM injected individual keylogger, filestealer, and remoteshell modules into memory space. But for the most part, deployed SPLM maintained the structure of earlier executable code. In 2018, we now see them pushing individual and separate blobs of keylogger, filesystem, and remoteshell code that never touch disk. The larger, 300kb+ SPLM backdoors deployed in 2016 and 2017 are not observed any longer at targets in 2018. Instead, in-memory modules appear in isolation.

In addition to purely XAgent based code, we also observe zebrocy modules completely recoded into powershell from .Net.

Code and Crypto Comparisons

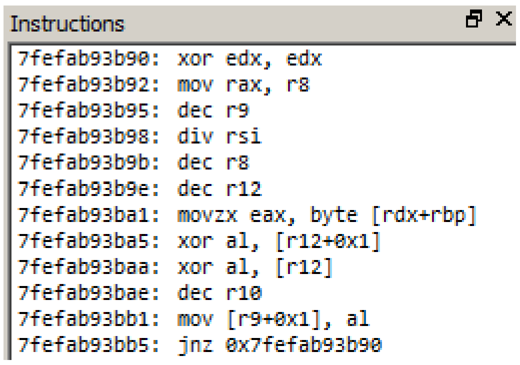
Current SPLM code maintains the unusual cipher hack that Sofacy began deploying in 2017 to hide away configuration details. Comparisons with cipher design and implementations we see in WhiteBear, earlier SPLM and Zebrocy modules tell a few things about design decisions and culture. And when specific malware sets are selectively deployed, that may tell us something about how efforts are divided.

SPLM full backdoor and plugins crypto and strings v4
973ff7eb7a5b720c5f6aafe4cd0469d5
Summary: SPLM is being carved up and delivered as memory-only chunks of compiled code. We observe the “retranslator” code, or ProcessRetranslator.dll, currently being delivered to systems without the presence of the previous, large, SPLM code and injection capabilities. The smaller plugins deployed in 2018 now maintain the same dynamic encryption code as the large 330kb full SPLM backdoors seen in more widespread use in 2017. Strings are well organized and concise.

Code and strings example (decrypted from 2018 “ProcessRetranslator.dll” plugin):

```
success command not correct
error save parameters
error set parameters for channel, see full info
command processing error
not correct command
command loading func/net module error
command unloading func/net module error
cmd.exe
Retranslator is now launched
Retranslator is now stopped
the process is destroyed
one thread has died so the process is killed too
create process failed on: (%s) , error (%d)
Retranslator is already running
Retranslator is not running
exit
command is successful
command is unsuccessful
```

SPLM crypto v3 (DNC hack)
cc9e6578a47182a941a478b276320e06
Summary: This earlier SPLM variant found on the DNC network in 2016 still maintains the internal name "splm.dll", with only one export "init" that was called at runtime. The C++ structure of this 280kb+ dll is familiar SPLM/CHOPSTICK, but it maintains a completely different cipher for decrypting configuration data and error messages, etc. The loop performing the bulk of the work is less than 100 bytes, performing pointer arithmetic alongside a couple xor operations of a lower nibble against sequential bytes.


Instructions

```
7fefab93b90: xor edx, edx
7fefab93b92: mov rax, r8
7fefab93b95: dec r9
7fefab93b98: div rsi
7fefab93b9b: dec r8
7fefab93b9e: dec r12
7fefab93ba1: movzx eax, byte [rdx+rbp]
7fefab93ba5: xor al, [r12+0x1]
7fefab93baa: xor al, [r12]
7fefab93bae: dec r10
7fefab93bb1: mov [r9+0x1], al
7fefab93bb5: jnz 0x7fefab93b90
```

Here, the cipher uses a modolo pointer arithmetic along with a decryption key per blob. Reviewing all the older ciphers and newer EC based ciphers in openssl and elsewhere results in no match.

WhiteBear code and strings
Summary: WhiteBear is a cluster of activity targeting foreign embassies and MFA organizations, starting in early 2016 and continued into early 2017. Our private GReAT report on this activity pushed in February 2017,

and a public report from another vendor described much of the same content almost seven months later as “Gayzer”. It appeared to be a parallel project to WhiteAtlas Turla, and maintained quirks like modular, well logged code with an elegant, professional RSA and 3DES encryption implementation and high quality code injection capabilities, but lots of immature and crude language and english mistakes. Clearly, english and maturity was not the developers’ native language.

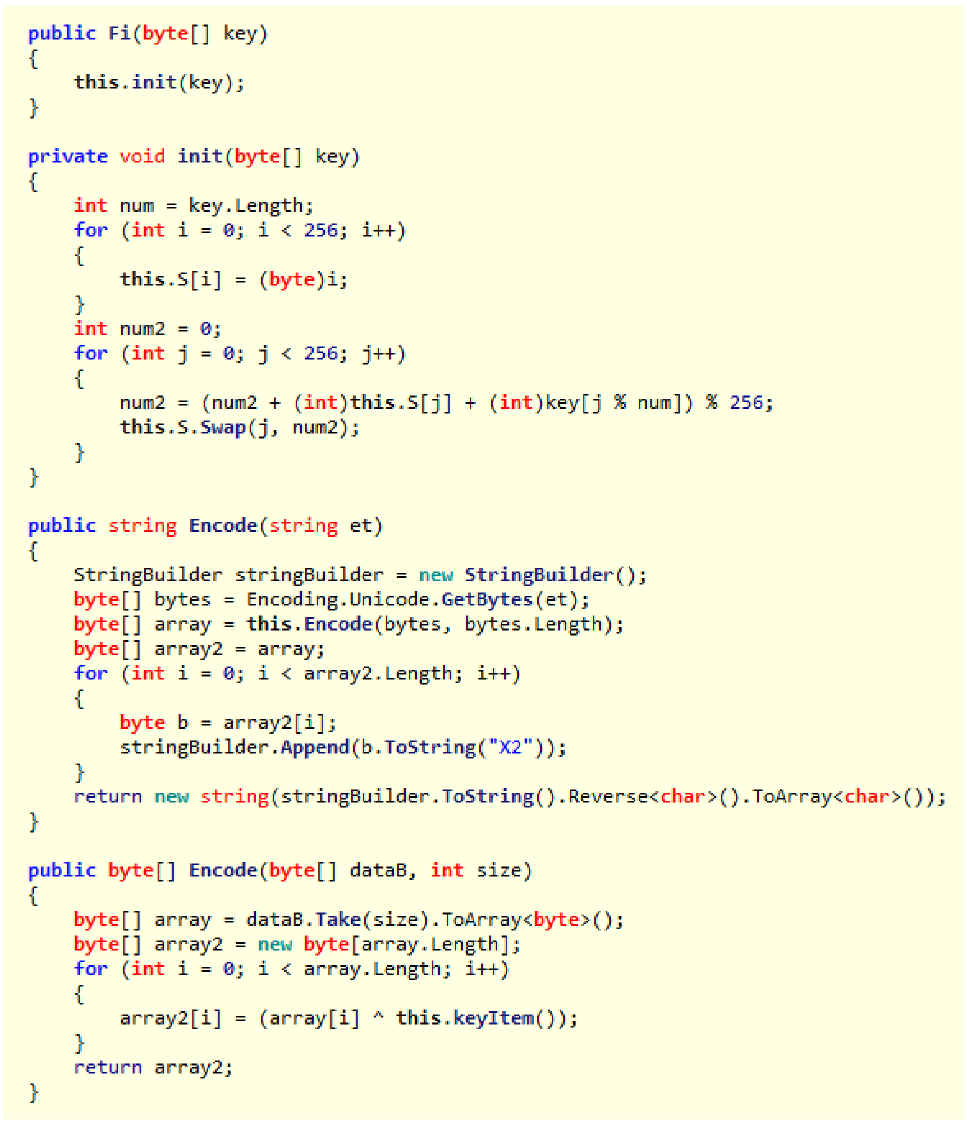
While WhiteBear is Turla related, it is interesting to compare to other ongoing development styles. Strings and code are crass.

Debug and command strings

```
i cunt waiting anymore #%d
lights aint turnt off with #%d
Not find process
CMessageProcessingSystem::Receive_NO_CONNECT_TO
_GAYZER
CMessageProcessingSystem::Receive_TAKE_LAST_CONN
ECTION
CMessageProcessingSystem::Send_TAKE_FIN
```

Zebrocy custom crypto
Summary: innovative .Net, AutoIT, Delphi, and powershell components are continually updated and deployed to new and old targets. Cryptography ranges from built-in windows api to custom RC4-based ciphers. Strings and code are simple, innovative, and concise.

Code:


```
public Fi(byte[] key)
{
    this.init(key);
}

private void init(byte[] key)
{
    int num = key.Length;
    for (int i = 0; i < 256; i++)
    {
        this.S[i] = (byte)i;
    }
    int num2 = 0;
    for (int j = 0; j < 256; j++)
    {
        num2 = (num2 + (int)this.S[j] + (int)key[j % num]) % 256;
        this.S.Swap(j, num2);
    }
}

public string Encode(string et)
{
    StringBuilder stringBuilder = new StringBuilder();
    byte[] bytes = Encoding.Unicode.GetBytes(et);
    byte[] array = this.Encode(bytes, bytes.Length);
    byte[] array2 = array;
    for (int i = 0; i < array2.Length; i++)
    {
        byte b = array2[i];
        stringBuilder.Append(b.ToString("X2"));
    }
    return new string(stringBuilder.ToString().Reverse<char>().ToArray<char>());
}

public byte[] Encode(byte[] dataB, int size)
{
    byte[] array = dataB.Take(size).ToArray<byte>();
    byte[] array2 = new byte[array.Length];
    for (int i = 0; i < array.Length; i++)
    {
        array2[i] = (array[i] ^ this.keyItem());
    }
    return array2;
}
```

Targeting Overlap and a Pivot to Asia

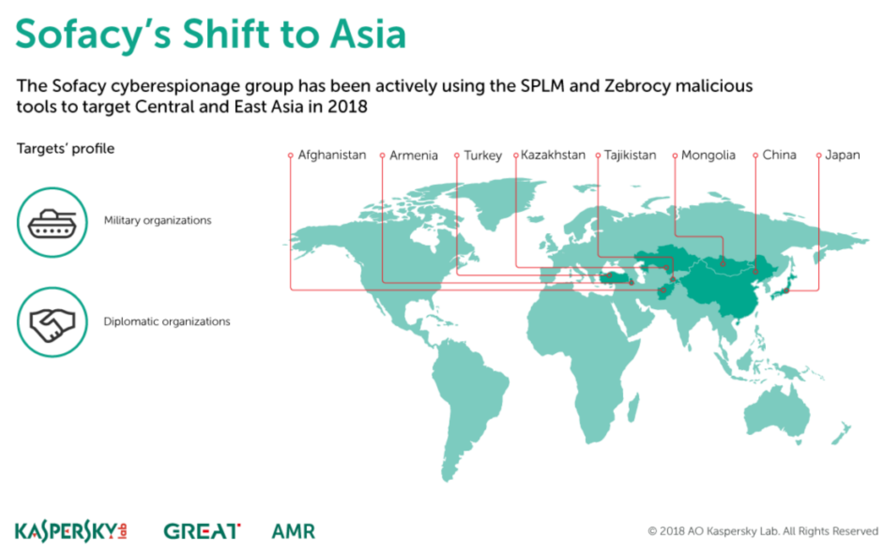
News headlines repeatedly trumpet Sofacy's foray into Western targets in the US and Europe, especially those connected with NATO. But these efforts only tell a portion of the Sofacy story.

Delineating groups and activity can be messy, and there appears to be overlap in targeting efforts across varying groups in central and east asia throughout 2017 and into 2018. Sofacy has been heavily interested in military and foreign affairs organizations here, resulting in multiple overlapped and competing targeting scenarios:

- Mosquito Turla and Zebrocy clusters – Zebrocy clusters targeting diplomatic and commercial organizations within Europe and Asia that match Mosquito targeting profiling
- SPLM overlap with traditional Turla – often Turla precedes SPLM deployments
- SPLM overlap with Zebrocy – Zebrocy often precedes SPLM deployments

- SPLM overlap with Danti

Currently, Sofacy targets large air-defense related commercial organizations in China with SPLM, and moves Zebrocy focus across Armenia, Turkey, Kazahkstan, Tajikistan, Afghanistan, Mongolia, China, and Japan. A previous, removed, report from another vendor claimed non-specific information about the groups' interest in Chinese universities, but that report has been removed – most likely detections were related to students' and researchers' scanning known collected samples and any "incidents" remain unconfirmed and unknown. On the other hand, this Chinese conglomerate designs and manufactures aerospace and air defense technologies, among many other enterprises. And here, an interest in military technologies is certainly within Sofacy purview.


Sofacy's Shift to Asia
The Sofacy cyberespionage group has been actively using the SPLM and Zebrocy malicious tools to target Central and East Asia in 2018
Targets' profile
Military organizations
Diplomatic organizations
Afghanistan
Armenia
Turkey
Kazakhstan
Tajikistan
Mongolia
China
Japan
KASPERSKY
GREAT
AMR
© 2018 AO Kaspersky Lab. All Rights Reserved

So, even more interesting than the shift eastward and other targeting overlap, is that the specific target system in China was previously a Grey Lambert target. The Sofacy modules at this system appeared to never touch disk, and resemble the Linux Fysbis code. Only one maintained the Filesystem.dll code, while another maintained ProcessRetranslator.dll code. However, it is unusual that a full SPLM backdoor was not detected on this system, nor was any powershell loader script. Because the injection source remains unidentified on such a unique system, we might speculate on what is going on here:

1. Sofacy attackers had recorded a previous Grey Lambert remote session and replayed the communication after discovering this host, essentially compromising the Grey Lambert module on the system to gain access and later injecting SPLM modules
2. Grey Lambert attackers inserted false flag and reduced SPLM modules
3. a new and unrecognized, full variant of SPLM attempted to inject module code into memory and deleted itself
4. an unknown new powershell script or legitimate but vulnerable web app was exploited to load and execute this unusual SPLM code

In all likelihood, the last option is accurate.

Conclusion

Sofacy is such a large, active group that appears to maintain multiple sub-groups and efforts that all fit under the Sofacy "umbrella". Sometimes, they share infrastructure, but more often they do not share infrastructure and appear to compete for access within targets. Either way, the group's consistent activity throughout central and eastern asia seems to be poorly represented in the public discussion.

SPLM did not change in substantial ways for several years, and now it is being split up and used for just functional modules. And much of the malware being deployed by Sofacy is quickly changed from C/C++ to .Net to powershell. Other open source and semi-legitimate pen-testing tools like nbtscan and powercat are being used for mapping available resources and lateral movement as well. It is easy to expect deliberate changes within this group in 2018, with even more .Net, Delphi, and powershell ports of various tools appearing at Sofacy targets throughout the year.

Technical Appendix

Early 2018 Reference Set
SPLM
452ed4c80c1d20d111a1dbbd99d649d5

ZEBROCY
efd8a516820c44ddbf4cc8ed7f30df9c
ff0e4f31a6b18b676b9518d4a748fed1

GAMEFISH
bd3e9f7e65e18bb9a7c4ff8a8aa3a784

GREY LAMBERT
86146d38b738d5bfaff7e85a23dcc53e

GREYWARE/PEN-TESTING NBTSCAN
f01a9a2d1e31332ed36c1a4d2839f412

APT   APT CRIMINAL   SOFACY   TARGETED ATTACK

Share post on:

Related Posts

The Slingshot APT FAQ

The devil’s in the Rich header

OlympicDestroyer is here to trick the industry