ESET discovers Attor, a spy platform with curious GSM fingerprinting

welivesecurity.com/2019/10/10/eset-discovers-attor-spy-platform

October 10, 2019

ESET researchers discover a previously unreported cyberespionage platform used in targeted attacks against diplomatic missions and governmental institutions, and privacy-concerned users

Zuzana Hromcová 10 Oct 2019 - 11:30AM

ESET researchers have discovered a new espionage platform with a complex architecture, a host of measures to make detection and analysis more difficult and two notable features. First, its GSM plugin uses the AT command protocol, and second, it uses Tor for its network communications. ESET researchers thus named the cyberespionage platform Attor.

Targets

Attor's espionage operation is highly targeted – we were able to trace Attor's operation back to at least 2013, yet we only identified a few dozen victims. Despite that, we were able to learn more about the intended victims by analyzing artifacts in the malware.

For example, in order to be able to report on the victim's activities, Attor monitors active processes to take screenshots of selected applications. Only certain applications are targeted – those with specific substrings in the process name or window title.

Besides standard services such as popular web browsers, instant messaging applications and email services, the list of targeted applications contains several Russian services, as detailed in Table 1.

Table 1. Domains misused in the campaign

Process name/window title substring	Context
ОДНОКЛАССНИКИ (transl. Classmates)	Russian social network (Odnoklass-niki)
AGENTVKONTAKTE	Russian social network (VKontakte)
WEBMONEY	Online payment system used in Russia (WebMoney)
MAIL.YANDEX, ЯНДЕКС.ПОЧТА (transl. Yandex.Mail), MAIL.RU, ПОСНТА (transl. Mail), MAGENT	Russian email services (Mail.ru, Yandex.Mail)
ПРИГЛАШЕНИЕ ДРУЖИТЬ (transl. Friend request)	Russian text
ВАМ СООБЩЕНИЕ (transl. Message for you)	Russian text
MULTIFON	Russian VoIP service

Table 1. Domains misused in the campaign

QIP, INFIUM	Russian IM application (QIP)
RAMBLER	Russian search engine (Rambler)

The list includes the two most popular social networks in Russia (Odnoklassniki, VKontakte) and a VoIP service provided by a Russian telecom operator (Multifon). Our conclusion is that Attor is specifically targeting Russian-speakers, which is further supported by the fact that most of the targets are located in Russia, as seen in Figure 1. Other targets are located in Eastern Europe, and they include diplomatic missions and governmental institutions.

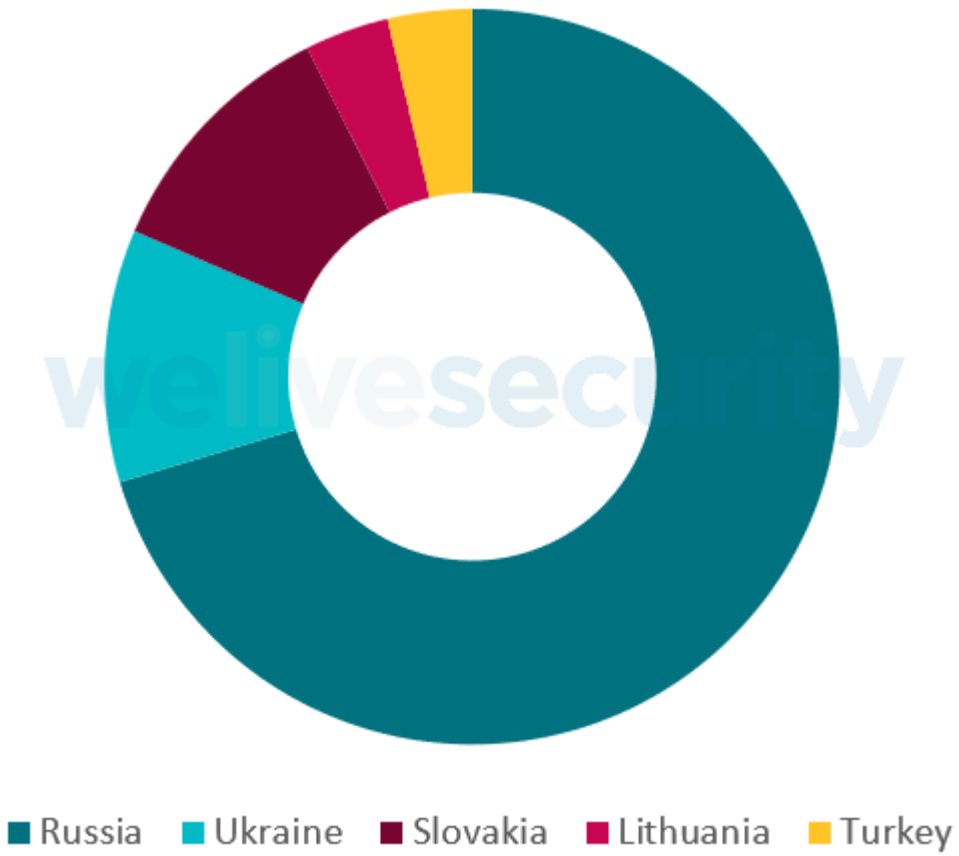

Figure 1. Countries affected by Attor

In addition to its geographical and language targeting, Attor's creators appear to be specifically interested in users concerned about their privacy.

Attor is configured to capture screenshots of encryption/digital signature utilities, the VPN service HMA, end-to-end encryption email services Hushmail and The Bat!, and the disk encryption utility TrueCrypt.

The victim's usage of TrueCrypt is further inspected in another part of Attor. It monitors hard disk devices connected to the compromised computer, and searches for the presence of TrueCrypt. If TrueCrypt is detected, its version is determined by sending IOCTLs to the TrueCrypt driver (0x222004 (TC_IOCTL_GET_DRIVER_VERSION) and 0x72018 (TC_IOCTL_LEGACY_GET_DRIVER_VERSION)). As these are TrueCrypt-specific control

codes, not standard codes, the authors of the malware must actually understand the open-source code of TrueCrypt installer. We have not seen this technique used before nor seen it documented in other malware.

```
.text:72E730B9                  mov     ecx, aTrueCrypt ; "TrueCrypt"
.text:72E730BF                  mov     ebx, ds:_snwprintf
.text:72E730C5                  push    ecx
.text:72E730C6                  push    offset aS       ; "\\\\.\\%s"
.text:72E730CB                  lea     edx, [esp+7CCh+fileName]
.text:72E730CF                  push    31h ; '1'       ; Count
.text:72E730D1                  push    edx             ; Dest
.text:72E730D2                  call    ebx ; _snwprintf
.text:72E730D4                  add     esp, 1Ch
.text:72E730D7                  push    ebp
.text:72E730D8                  push    ebp
.text:72E730D9                  push    3
.text:72E730DB                  push    ebp
.text:72E730DC                  push    ebp
.text:72E730DD                  push    ebp
.text:72E730DE                  lea     eax, [esp+7D0h+fileName]
.text:72E730E2                  push    eax             ; fileName
.text:72E730E3                  call    createFile
.text:72E730E8                  mov     esi, eax
.text:72E730EA                  cmp     esi, 0FFFFFFFFh
.text:72E730ED                  jz      loc_72E731A3
.text:72E730F3                  mov     edi, ds:DeviceIoControl
.text:72E730F9                  push    ebp             ; lpOverlapped
.text:72E730FA                  lea     ecx, [esp+7BCh+BytesReturned]
.text:72E730FE                  push    ecx             ; lpBytesReturned
.text:72E730FF                  push    4               ; nOutBufferSize
.text:72E73101                  lea     edx, [esp+7C4h+hDevice]
.text:72E73105                  push    edx             ; lpOutBuffer
.text:72E73106                  push    ebp             ; nInBufferSize
.text:72E73107                  push    ebp             ; lpInBuffer
.text:72E73108                  push    222004h         ; dwIoControlCode
.text:72E7310D                  push    esi             ; hDevice
.text:72E7310E                  call    edi ; DeviceIoControl
.text:72E73110                  test    al, al
.text:72E73112                  jz      short loc_72E7311B
.text:72E73114                  cmp     [esp+7B8h+BytesReturned], 4
.text:72E73119                  jnb     short loc_72E73132
.text:72E7311B
.text:72E7311B loc_72E7311B:                            ; CODE XREF: saveDeviceInfo+9A2↑j
.text:72E7311B                  push    ebp             ; lpOverlapped
.text:72E7311C                  lea     eax, [esp+7BCh+BytesReturned]
.text:72E73120                  push    eax             ; lpBytesReturned
.text:72E73121                  push    4               ; nOutBufferSize
.text:72E73123                  lea     ecx, [esp+7C4h+hDevice]
.text:72E73127                  push    ecx             ; lpOutBuffer
.text:72E73128                  push    ebp             ; nInBufferSize
.text:72E73129                  push    ebp             ; lpInBuffer
.text:72E7312A                  push    72018h          ; dwIoControlCode
.text:72E7312F                  push    esi             ; hDevice
.text:72E73130                  call    edi ; DeviceIoControl
```

Figure 2. The Device monitor plugin sends non-standard, TrueCrypt-specific control codes to the TrueCrypt driver, in order to determine the TrueCrypt version

Platform architecture

Attor consists of a dispatcher and loadable plugins, all of which are implemented as dynamic-link libraries (DLLs). The first step of a compromise comprises dropping all these components on disk and loading the dispatcher DLL.

The dispatcher is the core of the whole platform – it serves as a management and synchronization unit for the additional plugins. On each system start, it injects itself into almost all running processes and loads all available plugins within each of these processes. As an exception, Attor avoids injection into some system and security-product-related processes.

All plugins rely on the dispatcher for implementing basic functionalities. Rather than calling Windows API functions directly, the plugins use a reference to a helper function (a function dispatcher) implemented by the dispatcher DLL. A reference to the function dispatcher is passed to the plugins when they are loaded. Because the plugins are injected in the same process as the dispatcher itself, they share the same address space and are thus able to call this function directly.

Calls to the function dispatcher take as their arguments the function type and its numerical identifier. This design makes it harder to analyze individual components of Attor without having access to the dispatcher, as it translates the specified identifier to a meaningful function that is then executed.

Figure 3 illustrates a part of one plugin, calling the function dispatcher on several occasions. In the disassembly on the right, we have replaced the numeric identifiers (that we recovered by reverse-engineering the dispatcher) with descriptive names. Refer to our white paper for a full analysis of the dispatcher's interface.

Figure 3. Additional plugins use functions implemented in the main module, by calling the function dispatcher (dubbed helperFnc here)

Furthermore, the dispatcher is the only component of the platform that has access to the configuration data. Attor's plugins retrieve their configuration data from the dispatcher via the interface, as described above.

Plugins

Attor's plugins are delivered to the compromised computer as DLLs, asymmetrically encrypted with RSA. The plugins are only fully recovered in memory, using the public RSA key embedded in the dispatcher. As a result, it is difficult to obtain Attor's plugins, and to decrypt them, without access to the dispatcher.

We were able to recover eight of Attor's plugins, some in multiple versions – we list them in Table 2. Assuming the numbering of plugins is continuous, and that actors behind Attor may use different sets of plugins on a per-victim basis, we suspect there are even more plugins that have not yet been discovered.

Table 2. The analyzed plugins and their versions

Plugin ID	Analyzed versions	Functionality
0x01	0x0E	Device monitor
0x02	(no version), 0x0C	Screengrabber
0x03	(no version), 0x08, 0x09, 0x0B, 0x0C	Audio recorder
0x05	0x0A	File uploader
0x06	0x0A	Command dispatcher/SOCKS proxy
0x07	0x02, 0x04, 0x09, 0x0A	Key/clipboard logger
0x0D	0x03	Tor client
0x10	0x01	Installer/watchdog

The plugins are responsible for persistence of the platform (Installer/watchdog plugin), for collecting sensitive information (Device monitor, Screengrabber, Audio recorder, Key/clipboard logger) and for network communication with the C&C server (File uploader, Command dispatcher/SOCKS proxy, Tor client).

Attor has built-in mechanisms for adding new plugins, for updating itself, and for automatically exfiltrating collected data and log files. These mechanisms are illustrated in Figure 4.

In the following sections, we focus on plugins responsible for the two notable features that gave Attor its name – GSM fingerprinting via AT commands, and elaborate network communication using Tor.

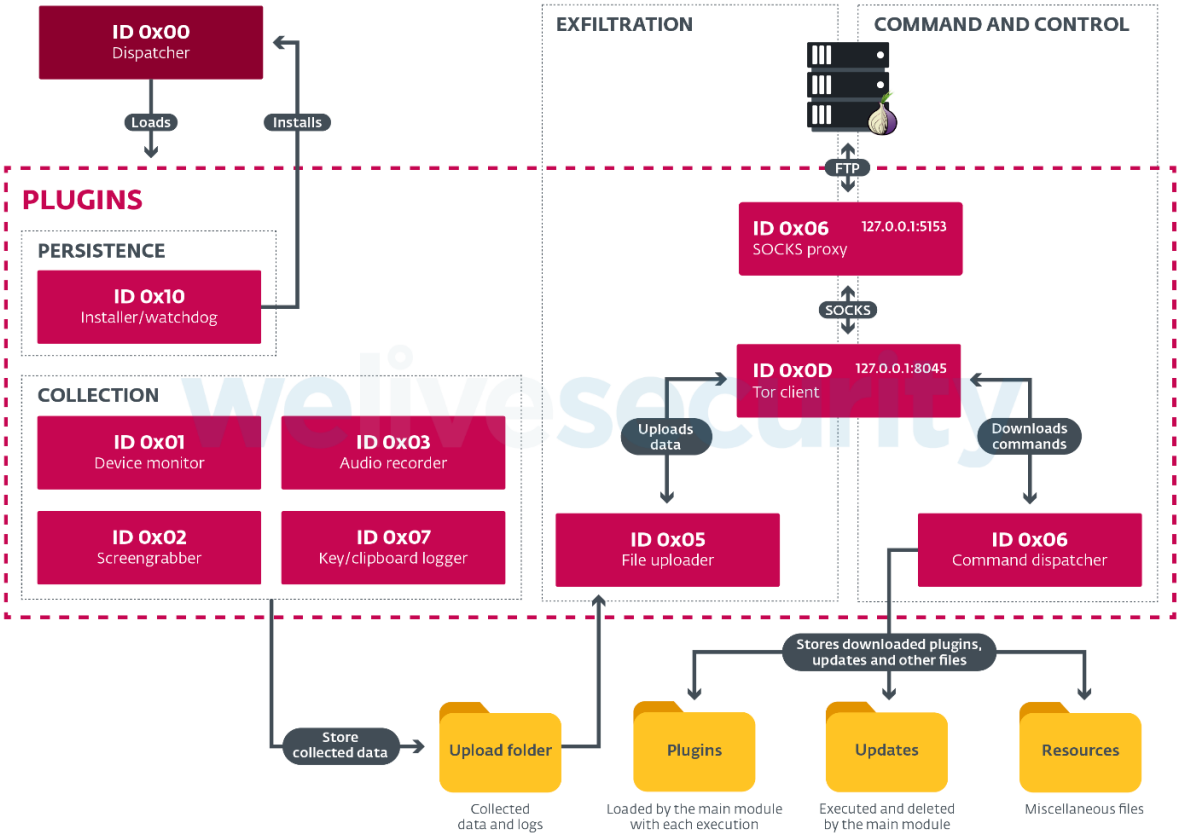

Figure 4. Attor's architecture. Note that ID 0x06 represents a single plugin, but the functionality is split here into two parts for clarity.

Network communication

Attor's espionage plugins collect sensitive data (such as a list of documents present on the disk) that are ultimately exfiltrated to a remote server, but these plugins themselves do not communicate over the network.

Only two of Attor's components communicate with its C&C server: File uploader and Command dispatcher.

Files collected by the "espionage plugins" (Device monitor, Screengrabber, Audio recorder, and Key/clipboard logger) are uploaded to the C&C server automatically by the File uploader plugin. These plugins use a dedicated Upload folder as a central folder to store collected data, and other plugins use it to store log files.

The Command dispatcher plugin downloads commands and additional tools from the C&C server and interprets them. Again, it uses dedicated folders to store its data – most prominently, freshly downloaded plugins and platform updates, and encrypted log data containing status/results of the executed commands.

Attor's dispatcher monitors the shared folders, and loads any new plugins and updates pushed to the compromised computer.

This means that neither Attor's dispatcher, nor espionage plugins, ever communicate with the C&C server – they only use local shared folders for storing data to be exfiltrated and for reading further instructions from the server.

Both File uploader, and Command dispatcher use the same infrastructure to reach the remote server – the network communication itself is scattered across four different Attor components, each implementing a different layer.

Attor uses Tor: Onion Service Protocol, with an onion address for the C&C server. In order to communicate with the C&C server, any plugin must thus first establish a connection with the Tor client plugin (listening on the non-default 127.0.0.1:8045) which is responsible for resolving the onion domain, choosing a circuit and encrypting data in layers. The Tor client plugin is based on the Tor client, and customized to the design of this malware (tor.exe with added interaction with Attor's dispatcher).

The Tor client plugin must communicate with the dispatcher, which implements the cryptographic functions. Furthermore, it communicates with the SOCKS proxy plugin (listening on 127.0.0.1:5153) that relays communications between the Tor client and the remote server.

Both File uploader and Command dispatcher use FTP; files are uploaded to/downloaded from an FTP server that is protected by credentials hardcoded in the configuration:

- **C&C server:** idayqh3zhj5j243t[.]onion
- **Username:** do
- **Password:** [Redacted]

The plugins log in to the FTP server and copy the collected data to, or download commands from, a victim-specific directory.

In total, the infrastructure for C&C communication spans four Attor components – the dispatcher providing encryption functions, and three plugins implementing the FTP protocol, the Tor functionality and the actual network communication, as illustrated in Figure 5. This mechanism makes it impossible to analyze Attor's network communication unless all pieces of the puzzle have been collected.

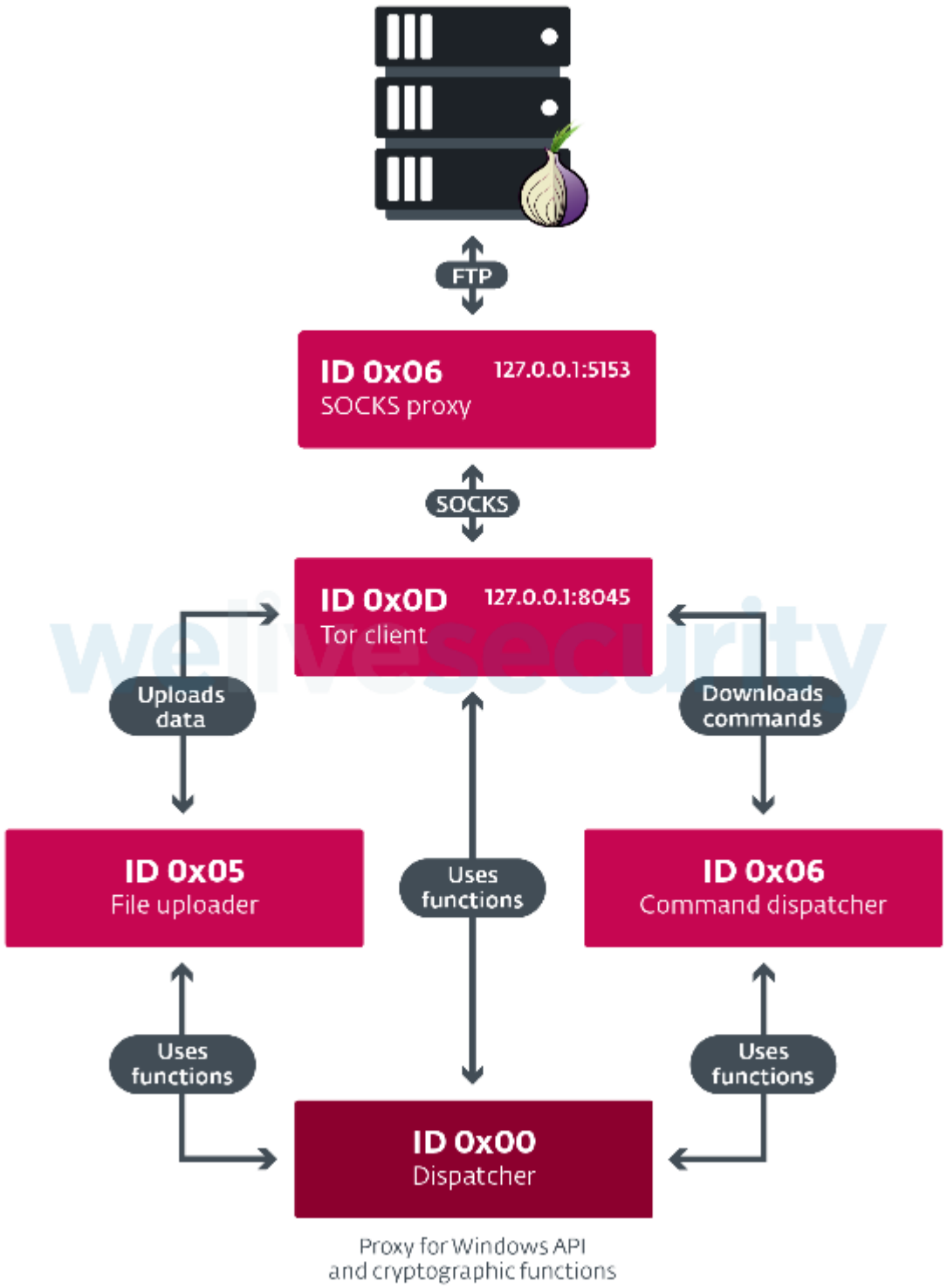

Figure 5. Four Attor components cooperate to enable communication with the C&C server

It is important to note that Attor uses several additional tricks to hide its communications from the user and security products:

First, the C&C server is a Tor service, aiming for anonymity and untraceability.

Second, all network-communication-related plugins are only activated if running within the process of a web browser or an instant messaging application or other network applications (this is determined by checking the process name against a hardcoded list). This trick hides the exfiltration-related network communication in a stream of legitimate communications made by that application, and thus reduces the risk of raising any suspicion.

GSM fingerprinting

The most curious plugin in Attor's arsenal collects information about both connected modem/phone devices and connected storage drives, and about files present on these drives. It is responsible for collection of metadata, not the files themselves, so we consider it a plugin used for device fingerprinting, and hence likely used as a base for further data theft.

While Attor's functionality of fingerprinting storage drives is rather standard, its fingerprinting of GSM devices is unique.

Whenever a modem or a phone device is connected to a COM port, Device monitor uses AT commands to communicate with the device, via the associated serial port.

AT commands, also known as Hayes command set, were originally developed in the 1980s to command a modem to dial, hang up or change connection settings. The command set was subsequently extended to support additional functionality, both standardized and vendor-specific.

In a recent paper, it was discovered that the commands are still in use in most modern smartphones. Those researchers were able to bypass security mechanisms and communicate with smartphones using AT commands through their USB interface. Thousands of commands were recovered and tested, including those to send SMS messages, emulate on-screen touch events, or leak sensitive information. That research illustrates that the old-school AT commands pose a serious risk when misused.

As for Attor's plugin, however, we may only speculate why AT commands are employed. We have detected a 64-bit version of this plugin in 2019, and we can confirm it is still a part of the newest Attor version (that we first saw in 2018). On the other hand, it seems unlikely it is targeting modern smartphone devices. The plugin ignores devices connected via a USB port, and only contacts those connected via a serial port (more precisely, devices whose *friendly names* match "COM*").

A more likely explanation of the plugin's main motive is that it targets modems and older phones. Alternatively, it may be used to communicate with some specific devices (used by the victim or target organization) that are connected to the COM port or to the USB port using a USB-to-serial adaptor. In this scenario, it is possible the attackers have learned about the victim's use of these devices using some other reconnaissance techniques.

In any case, the plugin retrieves the following information from the connected devices, using the AT commands listed in Table 3:

- Basic information about the mobile phone or GSM/GPRS modem: name of manufacturer, model number, IMEI number and software version
- Basic information about the subscriber: MSISDN and IMSI number

Table 3. The commands of the AT protocol used by the Device monitor plugin

AT command	Functionality
AT	Signals start of communication (AT for attention).
AT+MODE=2	Prepares the phone for an extended AT+ command set.

Table 3. The commands of the AT protocol used by the Device monitor plugin

AT+CGSN	Requests IMEI number (International Mobile Equipment Identity), which is a unique number to identify a device.
AT+CGMM	Requests information about the model of the device (model number).
AT+CGMI	Requests name of the device manufacturer.
AT+CGMR	Requests the version of the software loaded on the device.
AT+CNUM	Requests MSISDN (Mobile Station International Subscriber Directory Number), which is the mapping of the telephone number to the subscriber identity module in a mobile or cellular phone.
AT+CIMI	Requests IMSI (International Mobile Subscriber Identity), which is a unique number identifying a GSM subscriber. This number has two parts. The initial part is comprised of six digits in the North American standard and five digits in the European standard. It identifies the GSM network operator in a specific country with whom the subscriber holds an account. The second part is allocated by the network operator to identify the subscriber uniquely.

Note that many more (vendor-specific) AT commands exist that are not used by this plugin. It is possible that the malware operators use the listed commands to fingerprint the connected devices, and then deploy another plugin with more specific commands to extract information from the device.

Conclusion

Attor is an espionage platform, used for highly targeted attacks against high-profile users in Eastern Europe, and Russian-speaking, security-concerned users.

The malware, which has flown under the radar since 2013, has a loadable-plugin architecture that can be used to customize the functionality to specific victims. It includes an unusual plugin for GSM fingerprinting that utilizes the rarely used AT command set, and incorporates Tor with the aim of anonymity and untraceability.

Our research provides a deep insight into the malware and suggests that it is well worth further tracking of the operations of the group behind it.

ESET detection names and other Indicators of Compromise for these campaigns can be found in the full white paper: AT commands, TOR-based communications: Meet Attor, a fantasy creature and also a spy platform.

Acknowledgements to Anton Cherepanov, Peter Košinár, and Zoltán Rusnák for their work on this investigation.

MITRE ATT&CK techniques

Tactic	ID	Name	Description
Execution	T1106	Execution through API	Attor's dispatcher uses CreateProcessW API for execution.

Tactic	ID	Name	Description
	T1129	Execution through Module Load	Attor's dispatcher executes additional plugins by loading the respective DLLs.
	T1085	Rundll32	Plugin 0x10 schedules rundll32.exe to load the dispatcher.
	T1053	Scheduled Task	Plugin 0x10 schedules rundll32.exe to be executed on each boot/logon, and subsequently to load the dispatcher.
	T1035	Service Execution	Attor's dispatcher can be executed as a service.
Persistence	T1037	Logon Scripts	Attor's dispatcher can establish persistence via adding a Registry key with a logon script: HKEY_CURRENT_USER\Environment"UserInitMprLogonScript".
	T1050	New Service	Attor's dispatcher can establish persistence by registering a new service. HKEY_LOCAL_MACHINE\SYSTEM*ControlSet*\Control\SafeBoot\Minimal registry keys are updated to execute the service even in Safe mode and Safe mode with networking.
	T1053	Scheduled Task	Plugin 0x10 schedules a new task that loads the dispatcher on boot/logon.
Defense Evasion	T1140	Deobfuscate/Decode Files or Information	Strings are encrypted with a XOR cipher, using a hardcoded key. Configuration data, log files and plugins are encrypted using a hybrid encryption scheme – Blowfish-OFB combined with RSA.
	T1107	File Deletion	The collected files and log files are deleted after exfiltration by plugin 0x05.
	T1158	Hidden Files and Directories	The attributes of log files and directories are set to HIDDEN/SYSTEM/ARCHIVE (or combination of those).
	T1036	Masquerading	Attor's dispatcher disguises itself as a legitimate task (i.e., the task name and description appear legitimate).
	T1112	Modify Registry	Attor's dispatcher can modify the Run registry key.
	T1055	Process Injection	Attor's dispatcher injects itself into running processes, to gain higher privileges and to evade detection. It avoids specific system and Symantec processes.
	T1108	Redundant Access	Both 32-bit and 64-bit versions of Attor's dispatcher are executed; also they are injected into almost all processes. There is a watchdog component, implemented in the dispatcher or as a separate plugin, that reinstalls Attor if it has been removed.
	T1099	Timestomp	The time of last access to files and registry keys is manipulated after they have been created/modified.
	T1497	Virtualization/Sandbox Evasion	Attor can detect whether it is executed in some virtualized or emulated environments. If detected, it terminates itself immediately.

Tactic	ID	Name	Description
Credential Access	T1056	Input Capture	User credentials can be collected by plugin 0x07 via capturing keystrokes.
Discovery	T1083	File and Directory Discovery	Plugin 0x01 enumerates files with specific extensions on all hard disk drives and stores file information in encrypted log files.
	T1120	Peripheral Device Discovery	Plugin 0x01 collects information about inserted storage devices, modems and phone devices.
	T1082	System Information Discovery	Attor monitors the free disk space on the system.
Collection	T1123	Audio Capture	Plugin 0x03 is capable of recording audio using available input sound devices.
	T1119	Automated Collection	Attor automatically collects data about the compromised system.
	T115	Clipboard Data	Plugin 0x07 collects data stored in the Windows clipboard by using the OpenClipboard and GetClipboardData APIs.
	T1074	Data Staged	Collected data is staged in a central upload directory prior to exfiltration.
	T1056	Input Capture	Plugin 0x07 captures keystrokes pressed within the window of the process where Attor is injected.
	T1113	Screen Capture	Plugin 0x02 captures screenshots of target applications.
Command and Control	T1043	Commonly Used Port	Attor uses port 21 for C&C communication.
	T1188	Multi-hop Proxy	Attor uses Tor for C&C communication.
	T1079	Multilayer Encryption	Attor sends encrypted traffic using Tor, which itself uses multiple layers of encryption.
	T1105	Remote File Copy	Attor can download additional plugins, updates and other files.
	T1071	Standard Application Layer Protocol	FTP protocol is used for C&C communication.
	T1032	Standard Cryptographic Protocol	A combination of Blowfish-OFB and RSA is used for data encryption.
Exfiltration	T1020	Automated Exfiltration	Exfiltration of the collected data and log files is done automatically by plugin 0x05.
	T1022	Data Encrypted	Attor encrypts data with a combination of Blowfish and RSA ciphers before sending it to the C&C server.

Tactic	ID	Name	Description
	T1041	Exfiltration Over Command and Control Channel	Attor exfiltrates data over the C&C channel.

Zuzana Hromcová 10 Oct 2019 - 11:30AM