



ENJOY SAFER TECHNOLOGY**

AT COMMANDS, TOR-BASED COMMUNICATIONS: MEET ATTOR, A FANTASY CREATURE AND ALSO A SPY PLATFORM

Author: Zuzana Hromcová

In the fantasy book "A Court of Thorns and Roses" by Sarah J. Maas, Attor was an evil Faerie. In that book, the Faeries ruled over all the known world and humans were their slaves.

In cybersecurity, Attor is a cyberespionage platform designed to exfiltrate valuable information from its victims.

CONTENTS

1	SUMMARY
2	THE ESPIONAGE CAMPAIGN .
3	THE PLATFORM ARCHITECTURE
	3.1 Persistence
	3.2 Collection and exfiltration
	3.3 Network communication
4	TECHNICAL ANALYSIS .
	4.1 Dispatcher
	4.1.1 Encryption methods
	4.1.2 Evasion techniques
	4.1.3 Interface for plugins
	4.2 Shared resources
	4.2.1 Plugin folder, Update folder
	4.2.2 Upload folder
	4.2.3 Resource folder
	4.3 Plugins
	4.3.1 Device monitor (ID 0x01)
	4.3.2 Screengrabber (ID 0x02)
	4.3.3 Audio recorder (ID 0x03)
	4.3.4 File uploader (ID 0x05)
	4.3.5 Command dispatcher/SOCKS proxy (ID 0x06) 19
	4.3.6 Key/clipboard logger (ID 0x07)
	4.3.7 Tor client (ID 0x0D)
	4.3.8 Installer/watchdog (ID 0x10)
5	CONCLUSION
6	INDICATORS OF COMPROMISE (IOCS)
7	MITRE ATT&CK TECHNIQUES

1 SUMMARY

2

ESET researchers have discovered a previously unreported cyberespionage platform used in targeted attacks since at least 2013. Focusing on diplomatic missions and governmental institutions, Attor is designed specifically to attack privacy-concerned targets. Its most interesting features are a complex modular architecture, elaborate network communications, and a unique plugin to fingerprint GSM devices.

Attor's core lies in its dispatcher, which serves as a management and synchronization unit for additional plugins. It also provides an interface for the plugins to call Windows APIs and cryptographic functions indirectly.

Plugins themselves are heavily synchronized, with network communication alone being spread across four different components, each implementing a different layer. This allows the malware to communicate with its FTP C&C server, which resides on an onion domain. Tor is used for communication, aiming for anonymity and untraceability, and the overall setup makes it impossible to analyze the communication unless all pieces of the puzzle have been collected.

Apart from its dispatcher, all of Attor's capabilities are provided as plugins. This allows the attackers to customize the platform on a per-victim basis. The most notable plugin can detect connected GSM/ GPRS modems or mobile devices; this allows Attor to speak to them directly using the AT command set, in order to collect sensitive information such as the IMEI, IMSI or MSISDN numbers, possibly identifying both the device and its subscriber. Other plugins provide persistence, an exfiltration channel, C&C communication and several further spying capabilities.

We gave Attor its name because of two notable features of the platform: the AT protocol used by the GSM plugin, and that Tor is employed for network communication.

2 THE ESPIONAGE CAMPAIGN

Attor has been used in a highly targeted espionage campaign; we have only seen a few dozen victims. In general, Attor seems to be targeting Russian-speaking users. This claim is supported by the fact that most of the targets are located in Russia, and that the malware focuses on Russian applications: for example, it takes screenshots of Russian Instant Messenger (IM) apps.

Other targets are located in Eastern Europe, and they include diplomatic missions and governmental institutions.

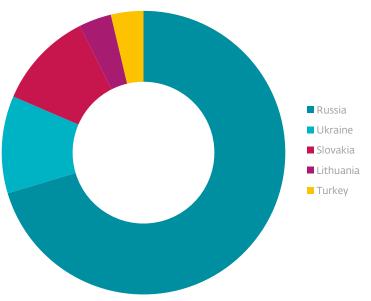


Figure 1 // Countries affected by Attor

TLP: WHITE

The campaign has been active in two waves, with different implementations of the platform, and substantial changes in architecture. We saw the first activity in 2013, while the most recent version has been active at least since 2018, with the latest samples detected in July 2019.

We don't know how the malware was initially distributed to its victims and executed. It is also difficult to establish the exact timeline of the operation, especially with such a small number of targets. The malware may have been used much longer: it could have been in use for years before we encountered it.

The PE timestamp information in the samples suggests compilation dates between 2005 and 2009 for the older version, and between 2011 and 2013 for the newer, but this information is likely fabricated. For example, in one of the samples¹, the PE header contains July 14th, 2009 as the compilation date, which is inconsistent with other metadata. Namely, the product version 1.626.7601.17514 in the sample's version info was apparently copied from a Windows 7 RTM library (usp10.dl1), in an attempt to make the file look more credible. However, that library was only released on November 20th, 2010, and thus its product version could not have been known in July 2009. This is a clear indicator that the timestamp is fake.

Therefore, we rely solely on our telemetry, which suggests that samples of the older version were used between 2013 and 2017, while the updated, modernized version has been in use since 2018.

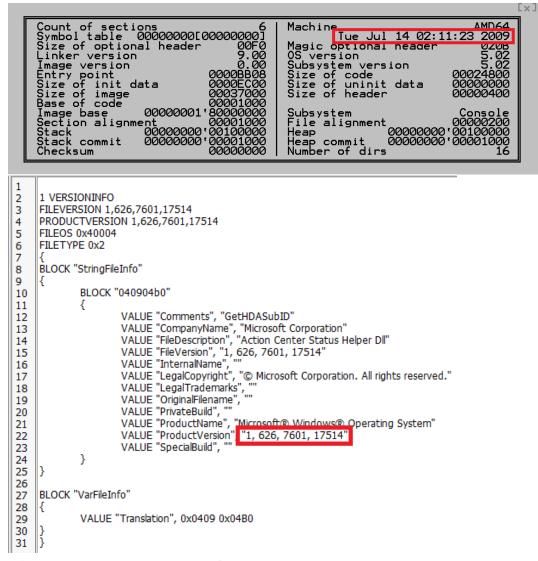


Figure 2 // Metadata from the malicious file showing that the PE timestamp was forged to an older date

3 THE PLATFORM ARCHITECTURE

4

In this paper, we focus on the newer version of Attor that has been active since 2018. This version consists of a dispatcher and loadable plugins, that use shared folders for synchronization, and provide functionality to each other.

In the first step of a compromise, a dropper delivers the main dispatcher and a number of plugins, and then loads the dispatcher, which is a dynamic-link library.

The dispatcher injects itself into almost all running processes and loads additional plugins within each of these processes. Attor avoids some system processes and Symantec products.

A feature characteristic to Attor is that the plugins then choose in which of the processes to activate their payload. For example, a Screengrabber plugin only captures screenshots of the window of a process in which it is injected, which allows monitoring only processes that are of interest of the attackers (such as IM applications or web browsers).

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

In subsequent sections, we explain how individual components of the platform contribute to these mechanisms, as is also illustrated in Figure 3.

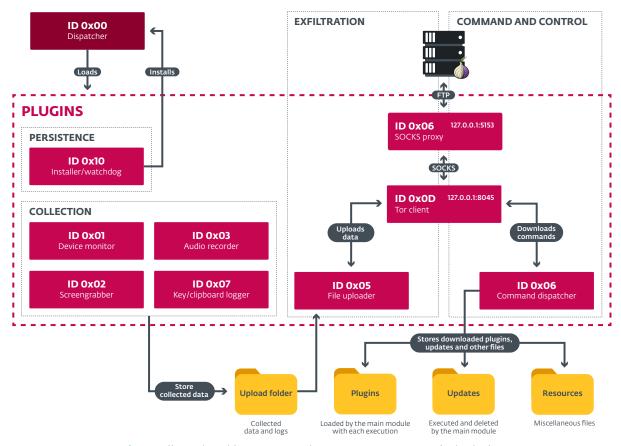


Figure 3 // Attor's architecture. Note that ID 0x06 represents a single plugin, but the functionality is split here into two parts for clarity.

3.1 Persistence

The **Installer/watchdog** plugin is responsible for ensuring Attor's persistence, by scheduling a task that loads the dispatcher.

In some versions of the malware, the dispatcher itself is responsible for this functionality, using one of two mechanisms depending on the privileges with which the malware was initially executed. Regardless of the persistence method, one of Attor's threads monitors whether it is still installed on the system, and reinstalls it if not.

The first technique used by the dispatcher is to create a logon script that loads the dispatcher DLL (using rundll32.exe) at each user logon. To achieve this, malware sets the UserInitMprLogonScript value under the HKEY_CURRENT_USER\Environment registry key.

If running in a privileged context, it registers the dispatcher as a service, and installs it to be executed on each system start by adding the service name to the **netsvcs** group.

Notably, Attor also instructs the service to be executed in Safe Mode and Safe Mode with Networking. This is achieved by creating a subkey for the malicious service under the SafeBoot registry keys:

- HKEY_LOCAL_MACHINE\SYSTEM*ControlSet*\Control\SafeBoot\Minimal
- HKEY_LOCAL_MACHINE\SYSTEM*ControlSet*\Control\SafeBoot\Network

Attor then sets the default value of these subkeys to Service, to indicate the malicious service should be loaded during a safe mode boot.

Note: In the provided registry keys, *ControlSet* represents all registry subkeys with the "ControlSet" substring in their names, e.g. ControlSet001 or CurrentControlSet.

3.2 Collection and exfiltration

Several plugins provide espionage capabilities; we refer to them as the **Device monitor**, **Screengrabber**, **Audio recorder** and **Key/clipboard logger** plugins. These plugins store the collected data and log files in a central Upload folder.

These plugins do not communicate with the C&C server themselves. Instead, it is the **File uploader** plugin that automatically uploads these files to the C&C server.

The main module monitors if there is enough space on the disk where the Upload folder is located – if not, the plugins do not proceed with creating additional files, in order to avoid suspicion by the user.

3.3 Network communication

There are only two Attor components that communicate with the C&C server: the aforementioned **File uploader**, and the **Command dispatcher** that downloads commands and additional tools from the C&C server.

Both plugins use FTP; files are uploaded to/downloaded from an FTP server that is protected by credentials hardcoded in the configuration. Passive FTP mode is used, where the client initiates both connections to the server.

The malware uses the Tor: Onion Service Protocol, with the C&C server having a .onion address, so the communication must be enabled by a **Tor client** plugin and relayed using a **SOCKS proxy** plugin. In order to communicate with the C&C server, any plugin must first establish a connection with the Tor client (listening on the non-default localhost:8045) which is responsible for resolving the onion domain, choosing a circuit and encrypting data in layers.

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

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. This mechanism is illustrated in Figure 4.

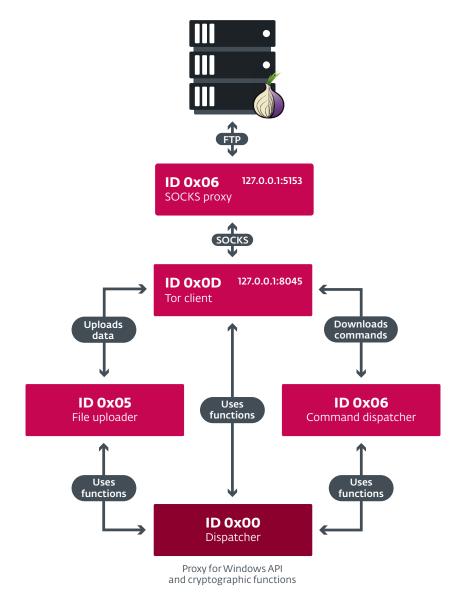


Figure 4 // Four Attor components cooperate to enable communication with the C&C server

4 TECHNICAL ANALYSIS

In this section, we will provide technical details about the respective Attor components.

4.1 Dispatcher

7

The dispatcher, which is a dynamic-link library, is a central unit of the platform. Its name usually starts with letters "SR", followed by some random letters. Dispatcher names we have seen include SRnhm.dll, SRSfxu.dll, SRzsp.dll, SRhhz.dll, SRanx.dll and sRWobwa.dll. For 64-bit versions, the "x64" suffix is added to the name, e.g., SRzspx64.dll.

The dispatcher contains encrypted configuration data with:

- Paths to working folders (Upload folder, Plugin folder, Update folder, Resource folder)
- FTP server domain and credentials (for C&C and exfiltration)
- RSA public key
- Name of the service under which the malware should be installed

Other components retrieve the configuration data from the dispatcher via a designated interface.

The dispatcher injects itself into all processes (except some system processes and some processes belonging to Symantec products). Subsequently, it loads all plugins from the Plugin folder (location of which is determined by the configuration).

The plugins then use functions implemented by the dispatcher, a reference to which is passed to the plugins on load.

There are both 32-bit and 64-bit versions of the main Attor dispatcher DLL. On 64-bit systems, both are loaded, which ensures the malware can inject itself into both 32-bit and 64-bit processes and monitor them (e.g., by taking screenshots and collecting keystrokes).

4.1.1 Encryption methods

Strings in the malware components are protected by an XOR cipher with a hardcoded key².

The configuration data (which is embedded in the main module), as well as the backdoor commands, are protected by a hybrid encryption scheme, where data is encrypted symmetrically using Blowfish and prepended with an RSA-encrypted Blowfish key.

The Blowfish cipher is used in OFB mode, with a 16-byte key randomly generated for each message. The public RSA key² used for encrypting the Blowfish key is embedded in the main module.

Encryption and decryption methods are implemented by the main module, but are also used by the loadable plugins via the interface provided by the main module (see Section 4.1.3). Attor's plugins use the same encryption scheme for encrypting the collected data.

4.1.2 Evasion techniques

Attor has implemented measures to detect whether it is being executed in an analytical environment. It terminates itself if an emulator or virtual machine is detected.

To detect (some) emulators, Attor performs two similar tests:

- It creates a randomly-named event, and then queries the event name via the NtQueryObject API function.
- It opens the HKEY_LOCAL_MACHINE\SYSTEM registry key and then queries its name by passing the key handle to the same API function.

If there is a mismatch in the names in any of these cases, Attor assumes the presence of an emulator and exits.

To detect virtual machines, Attor utilizes the following methods for detecting VMware, VirtualBox and Virtual PC:

- It calls the slat *instruction*, which stores the address of the Local Descriptor Table in a register; the address *should be* zero on native systems and non-zero in VMware. This is illustrated in Figure 5a.
- It executes an undefined instruction vpcext that is valid in Virtual PC and VirtualBox environments, but will generate an exception if executed in a native environment. This is illustrated in Figure 5b.
- It attempts to retrieve the VMware version by communicating with I/O ports, using the in instruction. If called in a VMware environment with the parameters seen in Figure 5c, the magic value VMxh is returned as the result.
- It tries to read from the \\.\VBoxMiniRdrDN pseudo-device, to discover whether it is running in VirtualBox. This is illustrated in Figure 5d.

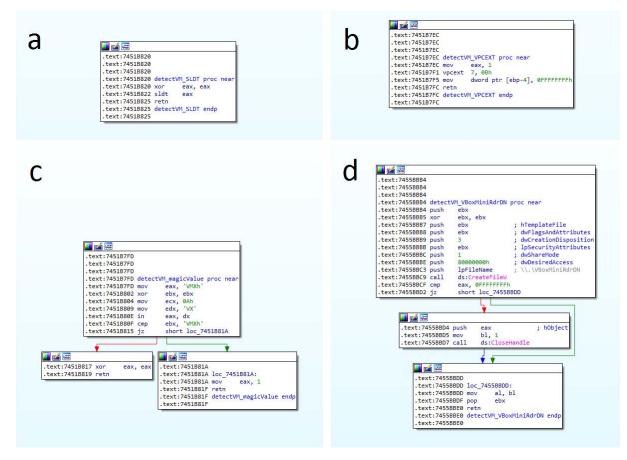


Figure 5 // Attor employs several techniques to detect specific virtual environments

Furthermore, Attor uses other techniques to avoid detection:

- Network communication plugins are injected into applications such as web browsers or IM applications, in order to blend in with legitimate network traffic.
- To prevent suspicion, whenever Attor manipulates a file or a registry key, it then restores the original values of their last access time.
- All files created by Attor have the hidden, system and archive attributes set in order to be hidden from Windows Explorer by default.

4.1.3 Interface for plugins

9

Attor's plugins use functions implemented by the main module, which makes them harder to detect and analyze, because they call functions of the dispatcher instead of Windows API functions.

The functions are called using the function dispatcher, which takes as its arguments the function type and its numerical identifier. Figure 6 illustrates a part of one plugin, calling the function dispatcher on several occasions. In the disassembly on the right, we replaced the numeric identifiers with descriptive names, that we recovered by reverse-engineering the dispatcher.

A reference to the function dispatcher is passed to the plugin on load. Since the plugins are loaded into the same process as the dispatcher, they share the same address space, and thus the plugins can call the functions of the dispatcher DLL directly. This design makes it harder to analyze individual components of Attor without access to the dispatcher, which translates the specified identifier to a meaningful function that is then executed.

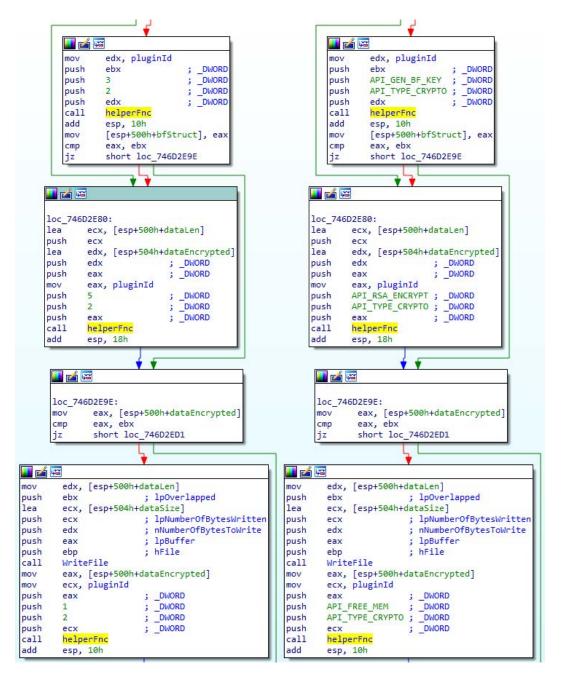


Figure 6 // Additional plugins use functions implemented in the main module, by calling the function dispatcher (dubbed helperFnc here)

According to our analysis, the dispatcher implements three to four dozen functions, depending on the dispatcher version (new functions are included in the newer samples). Table 1 lists functions recovered across all samples examined.

Function type	Function ID	Functionality	
	1	NtCreateFile wrapper	
	2	NtClose wrapper	
	3	SetFilePointer wrapper	
	4	NtFlushBuffersFile wrapper	
	5	Change file attributes	
	6	NtReadFile wrapper	
	7	NtWriteFile wrapper	
1 (resource manipulation	8	Create and close a file	
functions)	9	Create a session event	
	10	Create a mutex	
	11	Find first file	
	12	Find next file	
	13	Release all resources used by an unowned critical section object	
	14	Delete a file	
	15	NtOpenEvent wrapper	
	1	Free a memory block	
	2	Free a memory block and overwrite it with null bytes	
	3	Generate a new Blowfish key and initialize the encryption routine	
	4	Overwrite the Blowfish key with null bytes	
	5	Encrypt the Blowfish key with RSA	
	6	Encrypt/decrypt data with a previously initialized Blowfish key, in OFB mode	
2 (cryptographic	7	Hybrid decryption – decrypt the RSA-encrypted Blowfish key and then decrypt the Blowfish-encrypted data	
primitives)	8	Generate an RSA key	
	9	Hash data (SHA-1 or MD5)	
	10	Hash data – init part (SHA-1 or MD5)	
	11	Overwrite a memory block with null bytes	
	12	Hash data – update part (SHA-1 or MD5)	
	13	Hash data – final part (SHA-1 or MD5)	
	14	Hybrid encryption – generate a Blowfish key, encrypt it with RSA and then encrypt the data with Blowfish in OFB mode	

Table 1 // The interface provided by the dispatcher DLLs

Function type	Function ID	Functionality	
	1	Get path to the Upload folder (from the config)	
	2	Get path to the Update folder (from the config)	
	3	Get path to the Plugin folder (from the config)	
	4	Get path to the Resource folder (from the config)	
	5	Get 16 hardcoded bytes	
	6	Get attributes of ntdll.dll file (to be used for forging file timestamps)	
3 (retrieving configuration	7	Change file timestamps to those of ntdll.dll file	
data and global variables)	8	Check if there is enough space on disk	
	9	Get FTP directory name (from the config)	
	10	Get user ID (from the config)	
	11	Get structure with global variables and configuration data	
	12	Get user security descriptor	
	13	Get the specified config field	
	14	Resolve import	

4.2 Shared resources

11

Instrumental for synchronization between the main module and plugins is a set of four central folders, where various resources are stored.

The precise locations are hardcoded in Attor's configuration; most often we have seen the patterns in Table 2 used for naming these folders (x represents a variable lower-case letter).

Folder type	Naming pattern
Plugin folder	%ALLUSERSPROFILE%\Oracle\Java\.NET35\sxx
Update folder	%ALLUSERSPROFILE%\Oracle\Java\.NET35\v**
Resource folder	%ALLUSERSPROFILE%\Oracle\Java\.NET35\kxx
Upload folder	%ALLUSERSPROFILE%\Oracle\Java\.NET35\dxx

Table 2 // Naming convention for the folders

4.2.1 Plugin folder, Update folder

All plugins are permanently stored – compressed and encrypted – in the Plugin folder. The dispatcher loads plugins from this folder on each execution.

The Update folder serves as a temporary store for additional files the attacker wants to execute on the compromised machine. The files get pushed to this folder via the Command dispatcher plugin; once the dispatcher recognizes a DLL in the Update folder, it loads it automatically, and deletes it afterwards. According to the artifacts in the dispatcher, the thread responsible for this functionality is called Update_th; thus, we believe this mechanism most likely is used for updating Attor's components.

4.2.2 Upload folder

12

The Upload folder is used, by the plugins tasked with collection, as a central folder to store collected data, and by other plugins to store log files.

The File uploader plugin automatically uploads these files to the C&C server. Each file contains control data including plugin ID, file creation timestamp and a control flag 3ff3 and is encrypted using Blowfish-OFB combined with RSA. The hidden, system and archive attributes of the log files are set; the last access time is faked.

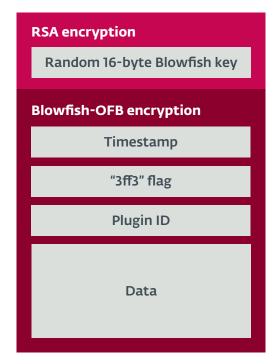


Figure 7 // Hybrid encryption scheme used by Attor

Additionally, index files are used to keep track of log files – they store the number of log files of a certain group, and the counter values are then used to name the log files.

Plugins usually store data in subdirectories of the Upload folder named by the plugin ID.

Filenames are derived from user names, collected data (such as clipboard contents or device information), timestamp or counter value, and are mostly hexadecimal values.

Table 3 // The properties of the files in the Upload folder

Filename pattern	Plugin ID	File content
1. <device-hash>\nm\index.dat</device-hash>	1	Index file for storage device metadata
1. <device-hash>\<counter><data-hash></data-hash></counter></device-hash>	1	Storage device information (bus type, device type, total disk size, TrueCrypt version)
1. <device-hash>\<counter><data-hash><data-hash></data-hash></data-hash></counter></device-hash>	1	Information about documents stored on the device (name, size, creation & last modified time)
1. <com-device-hash>\nm\index.dat</com-device-hash>	1	Index file for COM device metadata
1. <com-device-hash>\<counter><data-hash></data-hash></counter></com-device-hash>	1	COM device information (information about the phone/ modem, information about the subscriber)
2. <user-name-hash>\nm\index.dat</user-name-hash>	2	Index file for screenshots
2. <user-name-hash>\<counter><data-hash></data-hash></counter></user-name-hash>	2	Screenshot file
7\nm\info	7	Hash of last-stored clipboard data
7\nm\index.dat	7	Index file for keylog/clipboard files
7\ <counter><data-hash></data-hash></counter>	7	Keylog/clipboard data
nm\0_skp_ <user-name>\index.dat</user-name>	3	Index file for audio recordings
0_skp_ <variable-data></variable-data>	3	Audio recording file

Note: We only analyzed the older version of the plugin with ID 0x03. The file paths may be slightly different in the newer version, i.e., the one with the modern architecture.

4.2.3 Resource folder

This folder is used by all plugins to store miscellaneous resources that are not to be exfiltrated to the C&C server, such as scripts, timestamps or temporary files.

4.3 Plugins

13

Attor's architecture allows customization for individual targets, where the functionality of the platform relies heavily on plugins deployed on the compromised machine.

The plugins are implemented as DLLs, loaded by the dispatcher. They are stored on the disk in a compressed and encrypted form, with the valid form of the DLL only being recovered in memory, when the dispatcher loads the plugin. This probably is an attempt to thwart detection, as the plugin DLLs are never present unencrypted on disk.

All plugins export the same set of functions: DllCanUnloadNow, DllGetClassObject, DllRegisterServer, DllUnregisterServer, DllEntryPoint. When the dispatcher loads a plugin (DllEntryPoint), a start event is created, and several threads with the payload are started, which all wait until the event is signaled.

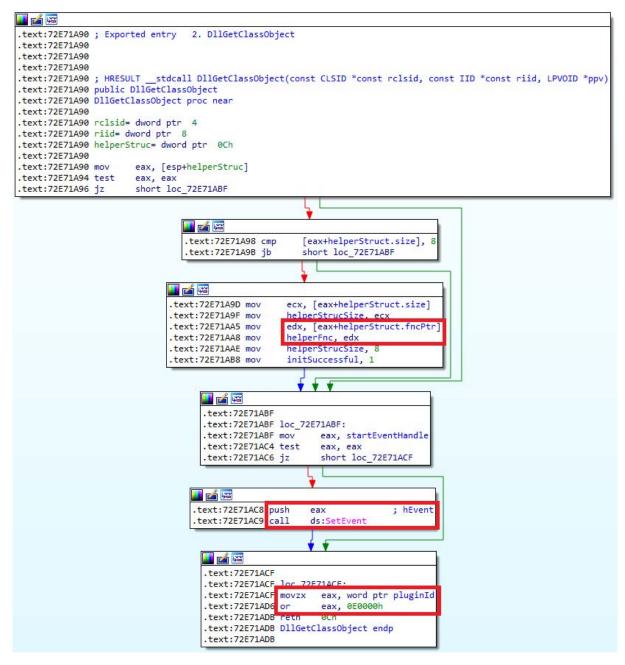


Figure 8 // DIIGetClassObject export of a plugin that is responsible for initialization and starting the actual payload

The plugins are activated by the main DLL calling the DllGetClassObject export (see Figure 8) with the following actions happening:

- The main DLL passes a reference to the helper function of the dispatcher, which the plugin will use for retrieving configuration data, for file manipulation, for encryption/decryption...
- The start event is signaled, which activates the threads
- The plugin returns the plugin ID and version, so that the main DLL can keep track of the active modules

The version information of plugins recovered in the course of our investigation, along with the plugins' functionality, is listed in Table 4.

Table 4 // The analyzed plugins and their versions

Plugin ID	Analyzed versions	Functionality
0x01	OxOE	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

Note that if the numbering is continuous (which it might be, since 0x10 is the last we have seen and is only version 1, while e.g., 0x03 is now on version 0x0C), there may be additional plugins available that we haven't seen.

We only can guess what functionality those plugins unknown to us might have. If we are to speculate, we feel that missing is a remote shell functionality, and also functionality to exfiltrate specific files silently; there may also be more victim-specific plugins available.

4.3.1 Device monitor (ID 0x01)

15

The most curious plugin in Attor's arsenal collects information about 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, used as a base for further data theft.

The Device monitor plugin creates a new window and uses SHChangeNotifyRegister API to register Shell-level notifications for that window on storage media insertion/removal, with a custom window message 0x421. The window procedure handles these events, and WM_DEVICECHANGE message, and collects a specific set of information for either a modem or a phone device, or an external storage drive.

1. External storage drive

In case a hard disk device is inserted, Device monitor collects the following information:

- Bus type, device type, device type modifier, vendor ID, product ID, product revision string, serial number, device interfaces
- Free space on disk, total disk size, filesystem used
- If applicable, also TrueCrypt version

The TrueCrypt 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 have been familiar with the code of the TrueCrypt driver, or TrueCrypt installer. This technique is very rare.



Figure 9 // Device monitor plugin sends non-standard, TrueCrypt-specific control codes to the TrueCrypt driver, in order to determine the TrueCrypt version

Further, it scans these hard disk devices for specific files and collects information about these files. It collects metadata such as filename, file size, creation and last-modified time of documents having the target extensions: .xls, .xlsx, .xlsm, .doc, .docx, .docm, .ppt, and .pptx.

The information gathered by this plugin is stored in separate log files in the Upload folder, in order to be exfiltrated by another plugin.

2. GSM/GPRS modem/phone device

The more unusual functionality of this plugin is the ability to fingerprint GSM devices. Whenever a modem or a phone device is connected to the 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 <u>recent paper</u>, it was discovered that the commands are still in use in most modern smartphones. The researchers were able to bypass security mechanisms and communicate with the smartphones using AT commands through their USB interface. Thousands of commands were recovered and tested, including those to send SMS messages, push touch events, or leak sensitive information.

This 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 modernized Attor platform (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 with "COM*" in the friendly name).

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 5:

- 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

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

Table 5 // The commands of the AT protocol used by the Device monitor plugin

Note that many more (vendor-specific) AT commands exist that are not used by this plugin. It is possible that 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.

4.3.2 Screengrabber (ID 0x02)

18

This plugin captures screenshots of targeted applications and stores them encrypted in the Upload folder. It is only activated in applications of the attackers' choice, determined by the process name.

After activation, the Screengrabber monitors the application's foreground window and if it is a web browser with a specific window title, or if it is another from the list of target applications, captures a screenshot every 10 seconds. For example, the plugin takes a screenshot if the victim is using an email client or has opened an email attachment.

Targeted are mostly web browsers, including all major (Firefox, Opera, Chrome, Safari, Internet Explorer) and a few less popular (Maxthon, Avant, SeaMonkey) ones. By checking the window title of the web browser, the malware can determine whether the user is visiting a website of interest.

We have encountered process names and window titles referring to these services:

- Social networks and blogging platforms (Facebook, Twitter, Google+, Blogger, Livejournal)
- Email services (Gmail, AOL Mail, Hotmail, Hotbox, Webmail, Yahoo! Mail)
- Office software (MS Word, Powerpoint, Excel, Visio, Notepad, Notepad++, Wordpad)
- Archiving utilities (WinRAR, WinZip, Hamster)
- Cloud storage and file sharing services (Turbobit, Skydrive, RapidShare)
- Messaging, VoIP applications and web-based messaging services (Skype, Viber, Miranda, Pidgin, QIP, qutIM)

Apart from these widely used applications, the list contains a number of irregularities, which can give us more insight about the intended victims.

For example, having TrueCrypt on the list suggests that the attackers are specifically interested in privacy-conscious users. Also targeted are encryption/digital signature utilities, a VPN application (HMA VPN), secure mail clients (The Bat! and HushMail) and a secure web browser (Dragon).

Furthermore, the applications on the list show the attackers are targeting Russian-speaking users – which is also confirmed by the geographical distribution of Attor, as explained in Section 2. The strings from the list of services targeted by the Screengrabber, that support this claim, are documented in Table 6.

 Table 6 // Artifacts from Attor's Screengrabber associated with Russian applications

Process name/window title substring	Context
ОДНОКЛАССНИКИ (transl. Classmates)	Russian social network (<u>Odnoklassniki</u>)
AGENTVKONTAKTE	Russian social network (<u>VKontakte</u>)
WEBMONEY	Online payment system used in Russia (<u>WebMoney</u>)
MAIL.YANDEX, ЯНДЕКС.ПОЧТА (transl. Yandex.Mail), MAIL.RU, POCHTA (transl. Mail), MAGENT	Russian email services (<u>Mail.ru</u> , <u>Yandex.Mail</u>)
ПРИГЛАШЕНИЕ ДРУЖИТЬ (transl. Friend request)	Russian text
ВАМ СООБЩЕНИЕ (transl. Message for you)	Russian text
MULTIFON	Russian VoIP service
QIP, INFIUM	Russian IM application (<u>QIP</u>)
RAMBLER	Russian search engine (<u>Rambler</u>)

Interestingly, the list of targeted applications contains some services that are now defunct or less popular than 10 years ago, which may suggest that the Attor platform is much older than we think.

4.3.3 Audio recorder (ID 0x03)

This plugin creates audio recordings using available microphones and stores them encrypted in the Upload folder.

The plugin is only activated if executed within a process of a teleconferencing application, which is determined by comparing the process name against a hardcoded list that includes Skype, Facebook Video Calling, Brosix or Cheap VoIP.

This may indicate the attackers are interested in eavesdropping on online calls.

With this mechanism, the attackers can also record the victim's conversations outside of teleconferencing apps. In this case, running in a process of a teleconferencing application can be an attempt to stay undetected, as the user is less likely to become suspicious that these applications access the microphone device.

4.3.4 File uploader (ID 0x05)

The File uploader plugin is responsible for exfiltration of collected data and log files to the C&C server.

The files are uploaded to the C&C server automatically, in batches. Files are uploaded in FTP passive mode, as binary content, and are deleted from the Upload folder after the exfiltration.

The C&C server and credentials are hardcoded in the configuration data; these credentials were used in all the collected plugins:

- C&C server: "idayqh3zhj5j243t[.]onion"
- username: "do"
- password: [Redacted]

The plugin logs in to the FTP server and copies the contents of the upload directory to a directory on the FTP server hardcoded in the config, creating a subdirectory there, with the victim ID as its name.

In order to reach out to the C&C server in the Tor network, the File uploader plugin uses other plugins (0x0D and 0x06) that provide the Tor and SOCKS proxy functionality (see Figure 4. Network communication.)

The File uploader plugin is 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 communication made by that application, and thus reduces the risk of raising any suspicion.

4.3.5 Command dispatcher/SOCKS proxy (ID 0x06)

This plugin serves two purposes.

First, it serves as a SOCKS proxy, used by the Tor client plugin (ID 0x0D) to communicate with the remote computer.

Second, it downloads commands from the C&C server and interprets them.

Just as with other plugins involved in C&C communication, Command dispatcher also hides its network communication using the trick described in Section 4.3.4.

The Command dispatcher plugin communicates with the C&C server via the FTP protocol using the 0x0D plugin (with Tor); the credentials are the same as those used by the File uploader plugin for exfiltration.

After the communication is established, Command dispatcher downloads a file called WAND.bin from the server, decrypts it (RSA + Blowfish) and parses the commands using the CommandLineToArgW API function.

The plugin supports four commands, as listed in Table 7.

Table 7 // Commands supported by Attor's Command dispatcher plugin

Command name	Functionality
SET	Set the value of an internal variable
MOVE	Rename a local file
DELETE	Delete a local file
СОРҮ	Download a file from the FTP server to the specified location

Of interest is the COPY command serves as a mechanism for the malware to download additional resources, namely:

- New plugins that will be stored in the Plugin folder and loaded by the main module on its next execution
- Updates that will be stored in the Updates folder and immediately loaded and then deleted by the dispatcher
- Miscellaneous files that will be stored in the Resource folder
- New versions of the dispatcher, either 32-bit or 64-bit

Command dispatcher stores encrypted log data containing status/results of the executed commands in the Upload folder, and timestamp of the last time the C&C server was contacted, in the Resource folder.

4.3.6 Key/clipboard logger (ID 0x07)

This plugin steals clipboard data, and collects keystrokes pressed within the process in which the DLL is injected.

Due to the fact the dispatcher injects itself into all running processes, and then loads additional plugins, this logger's DLL will be loaded into all processes as well.

This plugin scans the content of the clipboard and whenever a change is detected, it stores the data in a log file. It is only concerned with user GUI desktop ("WinSta0") and UNICODE format clipboard data (i.e., not files, objects etc.).

For keylogging, this plugin installs a hook function to the window of the current process that has the keyboard focus. To avoid detection, it uses a <u>less common technique</u>: instead of hooking <u>WH_KEYBOARD</u> or <u>WH_KEYBOARD_LL</u>, it hooks <u>WH_GETMESSAGE</u>. Whenever the window receives <u>WM_CHAR</u> or <u>WM_KEYDOWN</u> message, the plugin translates the pressed keystrokes into text.

The collected clipboard and keyboard data are stored in encrypted files in the Upload folder.

4.3.7 Tor client (ID 0x0D)

This plugin is based on the Tor client, customized to the design of this malware (tor.exe with added interaction with Attor's dispatcher).

It is responsible for computations, such as choosing a random circuit and providing multilayer encryption, and for establishing the communication with the Tor network by resolving the onion hostname.

For the actual network communication, it utilizes the SOCKS proxy implemented by plugin 0x06.

The attackers use Tor for C&C communication and exfiltration for anonymity and untraceability. Also, the Tor client plugin hides its network communication using the trick described in Section 4.3.4.

4.3.8 Installer/watchdog (ID 0x10)

This plugin establishes and monitors persistence of the main module.

It schedules a task that loads the main module with each system start or each user logon (depending on the process privileges). It then continues to monitor whether the task is still scheduled; if not, the plugin will reschedule it.

To run with necessary privileges, the Installer/watchdog plugin is only activated if injected into one of the following processes:

- %system%*
- %windir%\explorer.exe
- %windir%\regedit.exe

To make the task scheduling functionality stealthier, it doesn't utilize API calls; instead, it is implemented using the ITaskScheduler COM interface, registered using XML specification.

5 CONCLUSION

Attor is an espionage platform, used for highly targeted attacks, and has flown under the radar successfully since 2013.

Attor is professionally written malware with a loadable plugin architecture that can be used to customize the functionality to specific targets. It implements mechanisms for automated data collection and exfiltration, as well as for pushing additional plugins to the compromised machine.

Its functionality includes an unusual plugin for GSM fingerprinting that utilizes a rarely used AT command set. Attor incorporates Tor with the aim of anonymity and untraceability, and scatters the network communication functionality among several components, to make analysis rather difficult.

We were not able to recover the full operation timeline, nor the initial access vector. The versioning information in the plugins suggests there are other plugins that we have not yet seen. However, 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 this malware.

6 INDICATORS OF COMPROMISE (IOCS)

ESET detection names

Win32/Attor Win64/Attor

SHA-1

Note: We only present SHA-1 of older-version plugins, which used a simplified architecture and appeared unencrypted on disk.

Dropper

47dc997d08d53e55b8450940d9de94e2b5db631e 52213cab4954c850a1ac51974a24b878ca88eb59 8a6829b8615c5f6661a84ea3af0e15ab28c5840c d0528b8777d556809ac64935fa6247164aaabb4f f7e30a3fa186361794699d7a4fac6a9b85ccbf40

Dispatcher

1f69ba8063853dd80953b7881f4df9759a025780 37a1138f9ef575d3b7fc11d59473aea261be4aef 3c417b299532e62844869c4e0f33f3d719682f46 53c7232e00445fde63e7f7d75f6eecdb8a8a830e 6dec7c66cb5e6f86cddbe313b604603731c78e20 6f640e5acd1592424962df781707556f2c02cb17 87ded0f812ced722e86ff4a4f36bf8217c480388 9d8126148089646717c84f313fd52c1a9fec2aca a922558735227abe61754b02a6917a1aed18358a b7f1f151c3206e823fcaefcd0dab963be4efeb2f d1ca68eee49a4d25072855ebecc79d31f92efb2e e7371badae22c0ed0d93de2f1393c3e47dcee2d0

Screengrabber plugin

7b1e48154b93b6b374dd35bf821b2670ad1d0f69

Audio recorder plugin

02631dedb28c8d704ba689fc9efae057d2e8c8de 09d220b7da92a177d03eb6ca6711c500fe34667b 16b4c4da770991cc94e763ffec2a3fc418f9dc46 472eef198c8da7ea233d3f21393f5823968a8a0e 921e237b7e7c9945fb7cf5dd8960548a99896555 b194b2cdae26f5c65d62fe8f0204d1e80cd7a41f bd2845b2e7178da92c9014d0e8921b7efba5a924 c2371eb7a01149e87220bc7e62263266ee48be46 c3d7391863838fbbd88660392d114ab7536ca373 e64af8db5b6ccd076327a7c58c59e73eeb90be9b f9e9c4e45697880d8f601096e9dfc90e46557dcc

Key/clipboard logger plugin

5316abf8c4c148ce4d18419bb4e5057cf7535dcf a8112bd5e7dd78161c8b77638826c7cfc730f95f

C&C server

idayqh3zhj5j243t[.]onion

File-based indicators

Dispatcher filenames

ARMccl.dll ARMcclx64.dll odbccfg32.dll oledboz.dll SRanx.dll SRanxx64.dll SRhhz.dll SRhhzx64.dll SRnhm.dll SRnhmx64.dll SRSfxu.dlll SRSfxux64.dll sRWobwa.dll sRWobwax64.dll SRzsp.dll SRzspx64.dll

Plugin folder paths

%COMMONAPPDATA%\Lenovo\ImController\shared\US-df %COMMONAPPDATA%\Adobe\Setup\Replicate\US-sf %ALLUSERSPROFILE%\Oracle\Java\.NET35\sup %ALLUSERSPROFILE%\Oracle\Java\.NET35\sgp %ALLUSERSPROFILE%\Oracle\Java\.NET35\ssw %ALLUSERSPROFILE%\Oracle\Java\.NET35\ssm %ALLUSERSPROFILE%\Oracle\Java\.NET35\ssm %ALLUSERSPROFILE%\Oracle\Java\.NET35\ssm

Upload folder paths

%COMMONAPPDATA%\Lenovo\ImController\shared\US-vv %COMMONAPPDATA%\Adobe\Setup\Replicate\US-nh %ALLUSERSPROFILE%\Oracle\Java\.NET35\dmx %ALLUSERSPROFILE%\Oracle\Java\.NetFX\dtr %ALLUSERSPROFILE%\Oracle\Java\.NET35\dnf %ALLUSERSPROFILE%\Oracle\Java\.NET35\dtl %ALLUSERSPROFILE%\Oracle\Java\.NET35\deu %ALLUSERSPROFILE%\Oracle\Java\.NET35\deu %ALLUSERSPROFILE%\Sun\Java\Java Update\Caches\d51

Update folder paths

%COMMONAPPDATA%\Lenovo\ImController\shared\US-fu %COMMONAPPDATA%\Adobe\Setup\Replicate\US-zn %ALLUSERSPROFILE%\Oracle\Java\.NET35\vhb %ALLUSERSPROFILE%\Oracle\Java\.NET35\vtt %ALLUSERSPROFILE%\Oracle\Java\.NET35\vzu %ALLUSERSPROFILE%\Oracle\Java\.NET35\vwq %ALLUSERSPROFILE%\Sun\Java\Java Update\Caches\v1e

Resource folder paths

```
%COMMONAPPDATA%\Lenovo\ImController\shared\US-ba
%COMMONAPPDATA%\Adobe\Setup\Replicate\US-pq
%ALLUSERSPROFILE%\Oracle\Java\.NET35\knz
%ALLUSERSPROFILE%\Oracle\Java\.NET35\kya
%ALLUSERSPROFILE%\Oracle\Java\.NET35\kub
%ALLUSERSPROFILE%\Oracle\Java\.NET35\kmm
%ALLUSERSPROFILE%\Oracle\Java\.NET35\kmm
```

Synchronization object names

Note: xx represents a byte expressed in hexadecimal notation (a module ID or an object ID)

 $Global \{ 6A2B597F - 1C2F - 4f78 - AC94 - 7E1C8F136B0F \}$ Global \{ 3B68B0E9-08C4-4f23-B84C-CD76CCC4D362 } Global\{E5C403BC-555E-4162-8F18-EFF27A6957C1} Global \ { DC006092-9E86-4c20-8601-F643AF574444 } Global\{C4AE71DA-D55C-4a41-9F86-9F97D3CD47D9} Global \{7D715AC5-B909-4aaf-90C9-882CC4423F8C} Global \{377142C5-B7BD-42da-A8B3-4CAD9DA53326} Global \{343D67FA-567D-415e-9FAE-622153EFE241} $Global \{ 144DB9C5-A3B7-4eab-AD9A-0E80B3FF7853 \}$ Global \{7BD3390E-05C1-43a0-B938-94F52C3F3D32} Global\Root#SYSTEM#0000#{43850F15-CB8E-4a88-9BA0-F11FB0**xx**A2**xx**} {770F45C0-D879-4d4d-9F03-3AC9F2CA4ECB} {69DB17B0-4DA5-43b6-8A63-76B483AE7D5A} [HKEY_CURRENT_USER\Software\Classes\CLSID\{A7F7DE86-C5BF-4239-A220-BF7523F5501E}] [HKEY_CURRENT_USER\Software\Classes\CLSID\{A8F7DE86-C5BF-4239-A220-BF7523F5501E}] [HKEY_CURRENT_USER\Software\Classes\CLSID\{A7E7DE86-C5BF-4239-A220-BF7523F5501E}] [HKEY_CURRENT_USER\Software\Classes\CLSID\{A8E7DE86-C5BF-4239-A220-BF7523F5501E}]

Service display names

Adobe Acrobat Update v3.17.1kws Adobe Acrobat Update Service v3.2.muitnl Microsoft .NET Framework NGEN v3.5.10549_X64_86 Microsoft .NET Framework NGEN v3.5.10549_X64_86 Microsoft .NET Framework NGEN v3.5.10549_X64_86 Microsoft .NET Framework NGEN v3.5.10549_X64_86

Cryptographic keys

XOR cipher keys

25

CE2064368387DE357A3752F5 9EF725BA7881049E72F7A92C924403B0 9E772541AB8C93301F50D1CC942A1954720B8CDA 34695948760973CC78077E64576B934344E2525072DB71897F848EF3B823BCDF 53427F93B3716DB92511666C379758B5473790FFBB29811C4E8C9607B7CF385F 7C7B55DBA50A286B71F86EA5691580287FF0A8C5407D656540EAB3FC96582215 88EA7F35861891EF7AE62BA541545939B92E85BC24FD59E3B1C2991F99A14E9C AB783BBF82C9320F376A3D45B3CD402EA2D3AFFE648BBBBA5E9B724520A3627E AEB1A7C9BF3F61777B8B820C74ED6C62A838A2D5A0B56A2195552CA7713D40FE BEB05E4C303A65902E57A2C570465BD3993E5BCBB295BCF06AD167997A175247

RSA public keys

----BEGIN PUBLIC KEY-----

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAowPQz4fof3j6eFMysdwU s9fXAiij6dRC4yYFm32OSKaYGFSEJLoln+prl7Dx5YCI7MTYa1aAQtTkWpSTTgsl tF174pGFl7+ycxbCILAOZiHeCoEmk0463WJaLaNq27nQYSo4tCmdHy8DOH/ZYPwH cNM/8TsHFJjYsagVgrZXGdjdT78nxDDAORy5ey2Fb6tovVdT5oFGn3fJLcpyRzQK 0QL4/GeLis53IC8tlflGjnChT2R5rbecKk1d4b+AkhawwmOgsCsZJtgh+MpSyVAH KMqk5F30u7QjsxP50mw4hzDyEPTFgNvtlekQuyYeutU8/45p933tD08jJviqqOpi 8wIDAQAB

----END PUBLIC KEY-----

----BEGIN PUBLIC KEY-----

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAq+wyxqkGYGSrU6wbMz5y I2FfwmsFM8MbhU/73LSo/U0TYcecX7CGtT+yi+aG0Bddox4GKFebPYHqnOr0fVaB FxfjyDx359qgrtfbeGH4qUlVCheHWt890urb8m5SntqGN3m1PniK5u7Vvg8vvihw 20Z/FXUPjzvf99w45FeBSSj/FoOp7wHGA+HLArbpfRHHadj6MaxrCGqCXJZHFLYy 6j3Y0LFtRkw6dOGAfgSM4w10NLbzR1037zUZ4kkJ1itnNDBzfDJlx4IbZo82ale+ fFiHYywxwIlLpbPzqcKHCwpx6rrz6t/Twp7FGNldRxV4AjlGxLyevQnMeCAnKgjg WwIDAQAB

-----END PUBLIC KEY-----

----BEGIN PUBLIC KEY-----

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArFYkoqZrdAm4zDI/hYT5 p2WfqOSzrD+bdr6VuZYYa1CInG1xciAwAZcDZi+wB8bFoCWIsVf7v6d6eTJ84D08 e8av4RKt26mXwWglm0znmWuGEGsrsmbBNjWJgl5jI2qb0hcMtLPc4LV/ngz8KWbE qxwPfnFYrhhYgCgFjWbJ1A8azRyOYWLo7qhFYeO+Fmvb480/9lZ4hYEhPRzKL3nW sTR8WFiDNKuMh9CxAFBsialMHKUCC9V06J8T1mYC1f5e7Gj4Aaa+puEHtervPUdZ ZhYURbAr8fEmH65gIat8iZVkUAqdRJetVpWnXypaoxP9ZeXc7T7vPMYzpf4l7l9u zQIDAQAB

----END PUBLIC KEY-----

----END PUBLIC KEY-----

----BEGIN PUBLIC KEY-----

MIIBIjANBqkqhkiG9w0BAQEFAAOCAQ8AMIIBCqKCAQEA4rdRiBiEW7RKi+nGuQkQ zzy6B5M1patDsz8CplezC5ML4TGVjzU1IVpe/qzVoaYoDH5snjRuzcTWc4+BZlE3 2+HAVzySXkTxIm6tBPR10NIj2xUMKyoKlswS45iSbnFyDfwRdoKTVy1PoFQ27HSR DB355xIba7reQf3p44ByLoNByG19Qp0GNpKxY7x74pRJDe8rYF9rcqNjEGjhy4Pv wnzK/EItumjT9wuMcngHggG1T5hjgZQoLjVhAJ8zyNOmcVNsL9/DSUf1zg5SdGjv DV5/xFQGFeKegFKcZLukKA3et8y92NS2KLzQj3SIJVxFrSOK+rGfwy0pkr6qW00a TQIDAQAB

----END PUBLIC KEY-----

LWIDAQAB

----BEGIN PUBLIC KEY-----MIIBIjANBqkqhkiG9w0BAQEFAAOCAQ8AMIIBCqKCAQEA1Kqs1+tUSi1Vh1zvJjcC D9Yj7X8vLBwf82XQ8be6qsew3SD3mrTQUBv5PaQWer6w5iJpkM0FLwx2kvLuLKtz x1cUWFcN2mSKCMHeJpdBPCEM7CwBjGjp9euSG+ZpvpOobvv/eggUVa68wm2HfNAk +WkpR/XKyOCBOQ7UzU7nHXka7MmdDLNuKrH+CSMGEK1VIuz6CjMPJkAwGpL1r1g3 a7nr4ihoLO2+r/dZUXqf7oyPLQdUiPH4I8eCzJVCMLi7kD7AqHvyOF/F9yYJ9DKO LEZrxG5wDnTKM+3Iwti6ULqCQwd/krlcSZkm+kwxN5htMwk2chmEONS6y7fwcMLW

----END PUBLIC KEY-----

WWIDAQAB

----BEGIN PUBLIC KEY-----MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvg5VRwJbmGuWCjGKBJFE KBGdvXxdak4HnUzf+YiQmeqCeJVeR0fmcpf+Mvs1kKMT6LEpqzI3GknQca8V0Y+ vHe22NOZYqAVmHVZWgtg9Le2tOCqBLaTeRHO3SejzT3fwxcWshpvOM1QSXx6WP5V OlNw3XqOM7D1t/MZwo4hi+RaqQPgPLWXigtqDoQuuv/tP7vGOiKfVpjv6oQ4AlAX yTauktpktB4Q51nmm3hxy/pET2WjXpwPOXrl+Cpny5lZC44KRKvX8yume97zXZpg gf8rebKSvwipIIRVeD0ihpazjQ9V8AzPSYpbmk8oF8TUB/hFyYWQRq8+hKZ6IlKC

----END PUBLIC KEY-----

QwIDAQAB

----BEGIN PUBLIC KEY-----MIIBIjANBqkqhkiG9w0BAQEFAAOCAQ8AMIIBCqKCAQEAru8YYbvLSqvQQTvDfNSU ADtZ3K2aFtyurvHZcOipzm3BA9RC9YS3HV3WLIo6bH2TzO9mJA5piBtYWNiSasDu 0zzS8TtrCWiBBOkdiimrrNBoJ4KeWJnFEjL1Md8sZmpNUoqvUY2RwEl7J3Fg2+z/ d9hm23EX2LMLqDRDGFPKadD3GHCw9ZFoVbiUGzq2anEO3UebXHcWEZztktrRVcA0 U4w1LBzBkbonwDYY1VJ98W2BO3qtXiUWmsFRBas88nPO9B4fCR0Urz92ocOf8WBh JPMCkM7qPxW+mnOL7LHhZMpFcLb1lKzNXvUBUJ6T3DvZ5eO28WGoD8AUMHF6J1jx

Applications targeted by Screengrabber plugin

Process name substrings:

&RQ, 7ZFM, 7ZG, AGENTVKONTAKTE, AVANT, BROSIX, BROWSER, CHEAPVOIP, CHROME, DRAGON, EXCEL, FACEBOOKVIDEOCALLING, FINEREADER, FIREFOX, FIREFOXPORTABLE, FLOCK, GAJIM, GGAPP, HAMSTERARC, ICQ, IEXPLORE, INFIUM, LF30, MAGENT, MAXTHON, MIRANDA32, MOZILLA, MSIMN, MSMSGS, MSNMSGR, MULTIFON, MYIE, NERO, NICHROME, NIMBUZZ, NONOH, NOTEPAD, NOTEPAD++, OOVOO, OPERA, OUTLOOK, PGPKEYS, PGPTOOLS, PIDGIN, PIDGIN-PORTABLE, POWERPNT, PSI, PUFF, QIP, QQ, QUTIM, R&Q, RUN, SAFARI, SEAMONKEY, SKYPE, TBB-FIREFOX, TEAMVIEWER, THEBAT, THUNDERBIRD, TRILLIAN, TRUECRYPT, TWEETDECK, TWITTER-WIN8, VIBER, VISIO, W8VK, WEBMONEY, WINMAIL, WINRAR, WINWORD, WINZIP32, WLMAIL, WORDPAD, WUALA, WWAHOST, X-LITE, YAHOOM, YAHOOMESSENGER

Window title substrings:

************************, AOL MAIL, ATTACHMENT.GOOGLE, BLOGGER, BLOGSPOT, DEPOSIT FILES, FACEBOOK, FROMRU, GMAIL, GOOGLE+, HIDE MY ASS!, HIDEMYASS.COM, HOTBOX, HOTMAIL, HUSHMAIL, IMO, LIVEJOURNAL, MAIL.RU, MAIL.YANDEX, MESSENGER, MICROSOFT EXCEL WEB APP, MICROSOFT ONENOTE WEB APP, MICROSOFT POWERPOINT WEB APP, MICROSOFT WORD WEB APP, MSN, POCHTA, RAMBLER, RAPIDSHARE, SKYDRIVE, SKYPE, TURBOBIT, TWITTER, WEBMAIL, YAHOO! MAIL, BAM СООБЩЕНИЕ, ОДНОКЛАССНИКИ, ПРИГЛАШЕНИЕ ДРУЖИТЬ, ЯНДЕКС.ПОЧТА

Tactic	ID	Name	Description
	<u>T1106</u>	Execution through API	Attor's dispatcher uses CreateProcessW API for execution.
	<u>T1129</u>	Execution through Module Load	Attor's dispatcher executes additional plugins by loading the respective DLLs.
Execution	<u>T1085</u>	Rundll32	Plugin 0x10 schedules rundl132.exe to load the dispatcher.
	<u>T1053</u>	Scheduled Task	Plugin 0x10 schedules rundl132.exe to be executed on each boot/logon, and subsequently to load the dispatcher.
	<u>T1035</u>	Service Execution	Attor's dispatcher can be executed as a service.
	<u>T1037</u>	Logon Scripts	Attor's dispatcher can establish persistence via adding a Registry key with a logon script: HKEY_CURRENT_USER\Environment "UserInitMprLogonScript"
			Attor's dispatcher can establish persistence by registering a new service.
Persistence	<u>T1050</u>	TI050 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.
	<u>T1053</u>	Scheduled Task	Plugin 0x10 schedules a new task that loads the dispatcher on boot/logon.

7 MITRE ATT&CK TECHNIQUES

Tactic	ID	Name	Description
	<u>11140</u>	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.
	<u>T1107</u>	File Deletion	The collected files and log files are deleted after exfiltration by plugin 0x05.
	<u>T1158</u>	Hidden Files and Directories	The attributes of log files and directories are set to HIDDEN/SYSTEM/ ARCHIVE (or combination of those).
	<u>T1036</u>	Masquerading	Attor's dispatcher disguises itself as a legitimate task (i.e., the task name and description appear legitimate).
Defense	<u>T1112</u>	Modify Registry	Attor's dispatcher can modify the Run registry key.
Evasion	<u>T1055</u>	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.
		Redundant Access	Both 32-bit and 64-bit versions of Attor's dispatcher are executed; also they are injected into almost all processes.
	<u>T1108</u>		There is a watchdog component, implemented in the dispatcher or as a separate plugin, that reinstalls Attor if it has been removed.
	<u>T1099</u>	Timestomp	The time of last access to files and registry keys is manipulated after they have been created/modified.
	<u>T1497</u>	Virtualization/Sandbox Evasion	Attor can detect whether it is executed in some virtualized or emulated environments. If detected, it terminates itself immediately.
Credential Access	<u>T1056</u>	Input Capture	User credentials can be collected by plugin 0x07 via capturing keystrokes.
	<u>T1083</u>	File and Directory Discovery	Plugin 0x01 enumerates files with specific extensions on all hard disk drives and stores file information in encrypted log files.
Discovery	<u>T1120</u>	Peripheral Device Discovery	Plugin 0x01 collects information about inserted storage devices, modems and phone devices.
	<u>T1082</u>	System Information Discovery	Attor monitors the free disk space on the system.
	<u>T1123</u>	Audio Capture	Plugin 0x03 is capable of recording audio using available input sound devices.
	<u>T1119</u>	Automated Collection	Attor automatically collects data about the compromised system.
Collection	<u>T1115</u>	Clipboard Data	Plugin 0x07 collects data stored in the Windows clipboard by using the OpenClipboard and GetClipboardData APIs.
Concetion	<u>T1074</u>	Data Staged	Collected data is staged in a central upload directory prior to exfiltration.
	<u>T1056</u>	Input Capture	Plugin 0x07 captures keystrokes pressed within the window of the process where Attor is injected.
	<u>T1113</u>	Screen Capture	Plugin 0x02 captures screenshots of target applications.
	<u></u>	Screen Capture	

Tactic	ID	Name	Description
	<u>T1043</u>	Commonly Used Port	Attor uses port 21 for C&C communication.
	<u>T1188</u>	Multi-hop Proxy	Attor uses Tor for C&C communication.
Command	<u> 71079</u>	Multilayer Encryption	Attor sends encrypted traffic using Tor, which itself uses multiple layers of encryption.
and Control	<u>T1105</u>	Remote File Copy	Attor can download additional plugins, updates and other files.
	<u>T1071</u>	Standard Application Layer Protocol	FTP protocol is used for C&C communication.
	<u>T1032</u>	Standard Cryptographic Protocol	A combination of Blowfish-OFB and RSA is used for data encryption.
	<u>T1020</u>	Automated Exfiltration	Exfiltration of the collected data and log files is done automatically by plugin 0x05.
Exfiltration	<u>T1022</u>	Data Encrypted	Attor encrypts data with a combination of Blowfish and RSA ciphers before sending it to the C&C server.
	<u>T1041</u>	Exfiltration Over Command and Control Channel	Attor exfiltrates data over the C&C channel.

ABOUT ESET

For 30 years, <u>ESET®</u> has been developing industry-leading IT security software and services for businesses and consumers worldwide. With solutions ranging from endpoint and mobile security, to encryption and two-factor authentication, ESET's high-performing, easy-to-use products give consumers and businesses the peace of mind to enjoy the full potential of their technology. ESET unobtrusively protects and monitors 24/7, updating defenses in real time to keep users safe and businesses running without interruption. Evolving threats require an evolving IT security company. Backed by R&D centers worldwide, ESET becomes the first IT security company to earn 100 Virus Bulletin VB100 awards, identifying every single "in-the-wild" malware without interruption since 2003. For more information, visit www.eset.com or follow us on LinkedIn, Facebook and Twitter.

