Lazarus’ MacOS Dacls RAT Shows Multi-Platform Ability

blog.trendmicro.com/trendlabs-security-intelligence/new-macos-dacls-rat-backdoor-show-lazarus-multi-platform-attack-capability/

May 11, 2020

Malware

We found an application sample in April called TinkaOTP, and our investigation showed the application bearing a striking resemblance to Dacls remote access trojan (RAT), a Windows and Linux backdoor discovered in December 2019.

By: Gabrielle Joyce Mabutas, Fujisawa Kazuki May 11, 2020 Read time: (words)

A one-time password (OTP) system involves the use of a generated password that can only be used once to log in and access specific online services. Often managed by a third-party provider, this rolling password system aims to reduce unauthorized intrusions to systems via compromised accounts.

We found an application sample in April called TinkaOTP that seemed like a normal OTP authentication tool. However, further investigation showed the application bearing a striking resemblance to Dacls remote access trojan (RAT), a Windows and Linux backdoor that 360 Netlab discovered in December 2019. The strings that the trojan’s command and control (C&C) server uses to communicate with the storage samples links it to previous deployments by the cybercriminal group Lazarus.

The Lazarus link

After running the TinkaOTP application, the sample appeared as a Disk iMaGe (DMG) application bundle meant to be mounted and run. Launching the app initially didn’t reveal any suspicious behavior, but it dropped a hidden file *~/Library/.mina* in the Library folder, along with a LaunchDaemon */Library/LaunchDaemons/com.aex-loop.agent.plist* set to run the hidden file on startup.

Figure 1. TinkaOTP dropping a hidden file

Another comparison showed that the main mach-o executable of the bundle —*/Contents/MacOS/TinkaOTP* — was a repackaged version of another OTP tool called MinaOTP, an open-source authentication tool available at GitHub.

Figure 2. TinkaOTP (left) and MinaOTP (right, image taken from GitHub)

Figure 3. Disassembly of TinkaOTP’s main executable showed direct references to MinaOTP modules, as well as the shell command for copying the malicious hidden file

Tracing the origin of the hidden .mina file showed that it is a copy of an included resource, renamed SubMenu.nib, from the application bundle and where the main backdoor functions were contained. It also has the same links to Lazarus’ Windows and Linux predecessors: the presence of the hardcoded strings *c_2910.cls* and *k_3872.cls*. Both strings were previously used during C&C communication to the domain thevagabondsatchel[.]com as the sample storage of the cybercriminal group, as reported by 360 Netlab researchers.

Figure 4. Strings used to communicate with the C&C

Figure 5. Same HTTP Post message format as Windows and Linux backdoor discovered

We also found another variant with an executable that drops the main backdoor payload. The SubMenu.nib is nowhere to be found, but it downloads the payload from a hardcoded address and extracts it into the hidden payload to be run. A look into the address, loneeaglerecords[.]com, revealed that the domain has been registered and existing for 11 years, but the issued HTTPS certificate is fairly recent. The domain is assigned to 50[.]87[.]144[.]227, an IP investigated to have hosted different kinds of Windows malware such as ADWIND and URSNIF. Despite the differences, the hidden payload downloaded is the same in both variants.

Figure 6. Payload address of another variant

The group appeared to have made this version as a quick follow-up to the Windows/Linux Dacls RAT variants. This was confirmed by matching the latest IP assignment and HTTP certificate from the download address of the new variant.

Backdoor and persistence routines

The backdoor installation sequence shows that it’s meant for persistence via */LaunchAgents/com.aex-loop.agent.plist* and */Library/LaunchDaemons/com.aex-loop.agent.plist*. It initiates the configuration file */Library/Caches/com.applestore.db* to set the C&C server IP and for remote session information. Loading the bot plugins, this enables connection to the server to open and wait for commands, update the configuration file based on the commands received, and encrypt the file via AES CBC. If the configuration file already exists, it will decrypt once a new session starts.

Figure 7. Related code disassembly for persistence

Once installed, it checks if an existing configuration file exists. If not, it creates a config file and writes data related to the setup such as C&C server addresses and other C&C session information. The initial C&C server IPs — 67[.]43[.]239[.]146 and 185[.]62[.]58[.]207 — the bot connects to are hardcoded in the backdoor file and written to the config file. The config file will then be dropped as */Library/Caches/com.applestore.db* and encrypted. Once completed, the backdoor proceeds to load the plugins responsible for specific functions.

Figure 8. Initial server IPs written on the config file. The file may have the IPs rewritten to an updated IP address of a new server.

Figure 9. Dropping the config file and encrypting with AES CBC

A look into the server IPs also showed that the domain hosts legitimate traffic; blocking the said domains may have side effects for other sites.

Attack routines upon infection

During the initial analysis of the disassembled codes, the strings used to load the IPs and plugins appear to be missing characters. However, a closer inspection of the first characters showed that these strings were intentionally separated — likely an effort to evade detection.

Figure 10. Evading detection

After receiving a response from the server, it will check the headers to determine the plugins to run, with each plugin templated to load their respective functions on initialization. Comparing the plugins used for MacOS and its Windows/Linux variants showed some differences and additional capabilities.

Figure 11. Scanning headers

Plugin Name	String	Functions
Plugin_CMD	/bin/bash	· Execute received commands · Capability of Reverse Shell

Plugin_FILE	plugin_file	· · ·	Scan directories Download files Open, write, and delete
Plugin_PROCESS	plugin_process	· · ·	Collect data on running processes Create process Terminate process
Plugin_TEST	plugin_test	·	Check network access to specified address issued by server
Plugin_RP2Pv	plugin_reverse_p2p	·	Setup network proxy between bot and server
Plugin_LOGSEND	logsend	· · ·	Connect to log server Scan system Send collected logs
PLUGIN_SOCKS	plugin_socks	·	Start Socks4 Thread to setup SSL connection

Table 1. Plugin templates and functions

The *bash/cmd* plugin is used for executing shell commands in the form of appending them as a bash script and then running it on the terminal. Based on the arguments set in the packet received, it also has the capability to run a reverse shell.

Figure 12. The bash/cmd plugin

The *file* plugin runs the same functions as the Windows and Linux variants: it can read, write, delete, and download files, as well as scan a directory for a specific file.

Figure 13. The file plugin

Looking into the arguments shows that the *process* plugin comes from the received packet to execute functions such as collecting process information, running a new process, and terminating a running one. The process information collected includes the username, user ID, group ID, and process parent ID of the target process.

Whereas the other plugins may directly call and execute for the arguments' function passed by the server, the process plugin differs in that the server indirectly calls the function from the plugin itself; the location of the process plugin functions' addresses are first called prior to the execution.

Figure 14. The process plugin

Figure 15. Process plugin formatting the collected information before sending

The *test* plugin attempts to connect to a provided address to check access to the network. Meanwhile, the *reverse P2P* plugin creates a proxy server to bridge the C&C and the client. This creates another connection to another C&C specified in the commands to act as a proxy, redirecting traffic from the infected machine to the real C&C server.

Figure 16. The test plugin

Figure 17. The reverse P2P plugin

The *logsend* plugin collects system information by scanning the system using function *start_scan_worm* and send data to the log server specified. Meanwhile, the *socks* plugin creates a connection via socks4 for Secure Sockets Layer-related (SSL) transactions. This plugin was not observed in the Windows/Linux variants.

Figure 18. The logsend plugin

Figure 19. The socks plugin

Conclusion

The discovery of the Windows/Linux variants and analysis of this routine has shown Lazarus' range of expertise. As mentioned in a previous Lazarus discovery involving MacOS, this shift in focus towards attacking multiple operating systems indicate an expansion of targets. It also shows that they're experimenting for future-related cases, highlighted by the additional plugin that has not been observed in similar routines.

To note, shortly after our first discovery of Lazarus' interest in MacOS in 2019 via poisoned spreadsheets, the group was able to follow up with fileless AppleJeus, showing a rapid development in research. Related to this routine, the dropper from the first variant's only implemented evasion method was the use of copying the payload into a hidden file; the second variant improved on this by downloading the payload instead. While considerably not as sophisticated yet, given how the AppleJeus malware quickly followed with a fileless version, we could expect the group to deploy a similar execution of this routine soon.

We also suspect that the group may be targeting specific users for bot distribution, taking advantage of users' needs for layered security. OTP authentication tools have also been used to manage cryptocurrency wallets and exchanges, another common target for fraud by the cybercriminal group. The group might also be expanding to mobile platforms. Considering that the original MinaOTP source contains a separate project called MinaOTP-iOS, the group might be planning to rebuild a repurposed version of this for mobile. We will continue monitoring Lazarus deployments and activities.

To protect systems from this type of threat, users should only download apps from official and legitimate marketplaces. Users can also consider multilayered security solutions such as Trend Micro Antivirus for Mac, which provides comprehensive security and multi-device protection against cyberthreats.

Enterprises can take advantage of Trend Micro's Smart Protection Suites with XGen™ security, which infuses high-fidelity machine learning into a blend of threat protection techniques to eliminate security gaps across any user activity or endpoint.

Indicators of Compromise (IOCs)

SHA256	Filename	Detection
846d8647d27a0d729df40b13a644f3bffdc95f6d0e600f2195c85628d59f1dc6	/Contents/Resources/Base.lproj/SubMenu.nib	Backdoor.MacC
d3235a29d254d0b73ff8b5445c962cd3b841f487469d60a02819c0eb347111dd	TinkaOTP.dmg	Backdoor.MacC
e5b842784cc3e9bc0376915d2d823c3e4e076d29b5fb98ea69ff9a56b0f4a54a		
216a83e54cac48a75b7e071d0262d98739c840fd8cd6d0b48a9c166b69acd57d		
7e8a086319a218732dde5a749afdd9813d3047eaeef511e0374ca64fd8d0d033		
899e66ede95686a06394f707dd09b7c29af68f95d22136f0a023bfd01390ad53		
fea0bd961d8d72642a3e1cb92b6ac084a9680eaef816ad414e282f6ea87d52c6	TinkaOTP.app	Backdoor.MacC
7b8792025aacff5dacb3a9121ec2f5bfa33d5932d1f43b9ad0d518c55c6e1298	/Contents/MacOS/TinkaOTP	Backdoor.MacC
90fbc26c65e4aa285a3f7ee6ff8a3a4318a8961ebca71d47f51ef0b4b7829fd0		

URL

https[://]loneeaglerecords[.]com/wp-content/uploads/2020/01/images[.]tgz.001	Malware accomplice

MITRE ATT&CK Framework

Those highlighted in yellow are characteristics observed during analysis, while those in green are possible actions based on observed C&C server commands. MITRE_framework_Lazarus_Dacls_RAT_backdoor_MacOS

Tags

Malware | Endpoints | Research