Lazarus Campaign Uses Remote Tools, RATANKBA, and More

blog.trendmicro.com/trendlabs-security-intelligence/lazarus-campaign-targeting-cryptocurrencies-reveals-remote-controller-tool-evolved-ratankba/

January 24, 2018

Updated the detection names on January 25, 2018, 9:47 PM PDT

Few cybercrime groups have gained as much notoriety—both for their actions and for their mystique—as the Lazarus group. Since they first emerged back in 2007 with a series of cyberespionage attacks against the South Korean government, these threat actors have successfully managed to pull off some of the most notable and devastating targeted attacks—such as the widely-reported 2014 Sony hack and the 2016 attack on a Bangladeshi bank—in recent history. Throughout the Lazarus group's operational history, few threat actors have managed to match the group in terms of both scale and impact, due in large part to the wide variety of tools and tactics at the group's disposal.

The malware known as RATANKBA is just one of the weapons in Lazarus' arsenal. This malicious software, which could have been active since late 2016, was used in a recent campaign targeting financial institutions using watering hole attacks. The variant used during these attacks (TROJ_RATANKBA.A) delivered multiple payloads that include hacking tools and software targeting banking systems. We analyzed a new RATANKBA variant (BKDR_RATANKBA.ZAEL-A), discovered in June 2017, that uses a PowerShell script instead of its more traditional PE executable form—a version that other researchers also recently identified.

We identified a number of servers Lazarus used as a backend system for temporarily holding stolen data. We were able to access this backend, which provided us with valuable information about this attack and its victims.

Around 55% of the victims of RATANKBA's Powershell version were located in India and neighboring countries. This implies that the Lazarus group could be is either collecting intelligence about targets in this region, or is at an early stage of planning. They could have also been performing exercises in preparation for an attack against similar targets.

The majority of the observed victims were not using enterprise versions of Microsoft software. Less than 5% of the victims were Microsoft Windows Enterprise users, which means that currently, RATANKBA mostly affects smaller organizations or individual users, not larger organizations. It's possible that Lazarus is using tools other than RATANKBA to target larger organizations.

Lazarus' backend logs also record victim IP addresses. Based on a reverse WHOIS lookup, none of the victims can be associated with a large bank or a financial institution. However, we did manage to identify victims that are likely employees of three web software development companies in India and one in South Korea.

Infection Flow

Figure 1. RATANKBA Infection Flow

RATANKBA is delivered to its victims using a variety of lure documents, including Microsoft Office documents, malicious CHM files, and different script downloaders. These documents contain topics discussing either software development or digital currencies. The growth of cryptocurrencies may be a driving force behind the use of cryptocurrency-related lures.

An example of a lure used in a RATANKBA attack can be seen below:

Figure 2. Malicious CHM file used as RATANKBA lure

Once the lure's recipient opens and executes the file, a backdoor will be dropped into the victim's system. This RATANKBA backdoor is what is used to communicate with RATANKBA's Command-and-Control (C&C) server. We have observed two initial conversations with the C&C server (all are done via HTTP GET or POST to the server):

- HTTP POST to {script}.jsp?action=BaseInfo&u=XXX: Sends the victim information to the backend server
- HTTP GET to {script}.jsp?action=What&u=XXX: Checks if there are any pending jobs for the backdoor

This means that the backdoor is responsible for both uploading victim information, as well as executing any tasks that the controller has assigned to it, which includes the following:

- Killkill: Stops the backdoor's activities
- interval: Changes the interval in which the backdoor retrieves jobs; the default interval is set at 120 seconds
- cmd: Executes shell commands
- exe:Reflectively injects a DLL downloaded from a specific URL

In addition to the backdoor's modus operandi, the attackers will use a Microsoft WMI command-line tool to list the compromised system's running processes, which are sent to the C&C server:

- "C:\Windows\system32\cmd.exe" /c "wmic process get processid,commandline,sessionid | findstr SysWOW"

- “C:\Windows\system32\cmd.exe” /c “wmic process get processid,commandline,sessionid | findstr x86”

Technical Analysis

During our analysis, we collected a copy of the RATANKBA malware’s Lazarus Remote Controller tool. The remote controller provides a user interface that allows attackers to send jobs to any compromised endpoint. The controller gives the attackers the ability to manipulate the victims’ host by queueing tasks on the main server. RATANKBA retrieves and executes the tasks, and retrieves the collected information.

Figure 3. RATANKBA communication diagram

The RATANKBA malware has a control model that does not use real-time communication between the backdoor and the attacker. Instead, both the remote controller and the backdoor connect to its main communication control server to push or pull pieces of information. The controller uses a graphical UI interface and can be used to push code to the server, while the backdoor regularly connects to the server to check for pending tasks. The controller downloads the victim profiles from the server. If the profiles are already downloaded by the controller, they are deleted from the server side. The controller can post victim-specific tasks as well as global specific tasks to the server. Below are the various functionalities of RATANKBA’s controller:

Command Name	Function
get_time	Retrieves the server time
delete_inf	Deletes the downloaded victim profiles
delete_con	Deletes the connection log files if they were already downloaded
Kill:	Posts a job to kill the backdoor
inject	Posts a job for DLL injection
Interval	Changes the sleep interval
Cmd	Posts a job for command shell execution
delete_cmd	Retrieves the job results and deletes the posted job
broadcast_cmd:	Posts a job for all the backdoors connecting to the server

Figure 4. RATANKBA main console interface

Figure 5. RATANKBA host manipulation console

RATANKBA's controllers use the "*Nimo Software HTTP Retriever 1.0*" user-agent string for its communication. The communication protocol format for the controller and backdoor is as follows:

> · <domain>/<jsp filename>.jsp?action=<corresponding actions plus additional needed parameters>`

One of most notable changes on the new RATANKBA variant is that the new version was written in Powershell, whereas the original variant was in PE form. The shift from PE to Powershell makes it more difficult for antivirus solutions to detect. The screenshot below shows the conversion from C/C++ code to Powershell, while the protocol remained unchanged.

Figure 6. C/C++ version of RATANKBA

Figure 7. Powershell version of RATANKBA

Profile of the Attackers

While we do not have any knowledge of who the actual Lazarus attackers are, the data collected from the backend systems gives us some insights into the internet usage patterns of systems likely owned by Lazarus group members. Clues regarding the profiles of the attackers was also found, including those connected to developers and at least one operator. All of them appear to be native Korean speakers, or at least have Korean language proficiency that is at the near-native level. We believe at least one of them also understands Chinese.

We also observed clues that the attackers are interested in cryptocurrencies such as Bitcoin (BTC) and Ant Share (NEO). One of them transferred shares of NEO at a good market price.

Figure 8. Empty cryptocurrency wallet of the attacker

Figure 9. An attacker transfers 594 NEO to another wallet, with the money going to a mixer

Figure 10. An attacker mining Ant Share

Defending against RATANKBA

Given Lazarus' use of a wide array of tools and techniques in their operations, it's reasonable to assume that the group will continue to use ever-evolving tactics in their malicious activities. Overall, an organization will need multilayered security strategies, as Lazarus and other similar groups are experienced cybercriminals who employ different strategies to get past organizational defenses.

The impact of this malware can be mitigated with proven mitigation techniques such as routinely scanning the network for any malicious activity to help prevent the malware from entering and spreading through an organization. In addition, educating employees and other key people in an organization on social engineering techniques can allow them to identify what to look out for when it comes to malicious attacks.

Other mitigation strategies include a multilayered approach to securing the organization's perimeter, which includes hardening the endpoints and employing application control to help prevent malicious applications and processes from being executed.

Trend Micro™ Deep Security™ provides virtual patching that protects endpoints from threats such as malicious redirections to malware-hosting URLs as well as those that exploit unpatched vulnerabilities. Trend Micro™ Deep Discovery™ provides detection, in-depth analysis, and proactive response to attacks using exploits and other similar threats through specialized engines, custom sandboxing, and seamless correlation across the entire attack lifecycle, allowing it to detect these attacks even without any engine or pattern update.

A detailed timeline of the Lazarus group's operations can be seen here.

Indicators of Compromise (IoCs):

Hashes detected as BKDR_RATANKBA.ZAEL-A

- 1768f2e9cea5f8c97007c6f822531c1c9043c151187c54ebfb289980ff63d666
- 6cac0be2120be7b3592fe4e1f7c86f4abc7b168d058e07dc8975bf1eafd7cb25
- d844777dcafcde8622b9472b6cd442c50c3747579868a53a505ef2f5a4f0e26a
- db8163d054a35522d0dec35743cfd2c9872e0eb446467b573a79f84d61761471
- f7f2dd674532056c0d67ef1fb7c8ae8dd0484768604b551ee9b6c4405008fe6b

Hashes detected as CHM_DLOADER.ZCEL-A

- 01b047e0f3b49f8ab6ebf6795bc72ba7f63d7acbc68f65f1f8f66e34de827e49
- 030b4525558f2c411f972d91b144870b388380b59372e1798926cc2958242863
- 10cbb5d0974af08b5d4aa9c753e274a81348da9f8bfcaa5193fad08b79650cda
- 650d7b814922b58b6580041cb0aa9d27dae7e94e6d899bbb3b4aa5f1047fca0f
- 6cb1e9850dd853880bbaf68ea23243bac9c430df576fa1e679d7f26d56785984
- 6d4415a2cbedc960c7c7055626c61842b3a3ca4718e2ac0e3d2ac0c7ef41b84d

- 772b9b873100375c9696d87724f8efa2c8c1484853d40b52c6dc6f7759f5db01
- 9d10911a7bbf26f58b5e39342540761885422b878617f864bfdb16195b7cd0f5
- d5f9a81df5061c69be9c0ed55fba7d796e1a8ebab7c609ae437c574bd7b30b48

Hashes detected as JS_DLOADER.ZBEL-A

8ff100ca86cb62117f1290e71d5f9c0519661d6c955d9fcfb71f0bbdf75b51b3

Hashes detected as X97M_DLOADR.ZBEL-A

972b598d709b66b35900dc21c5225e5f0d474f241fefa890b381089afd7d44ee

Hashes detected as VBS_DLOADR.ZAEL-A

4722138dda262a2dca5cbf9acd40f150759c006f56b7637769282dba54de0cab

APT & Targeted Attacks

We analyzed a new RATANKBA variant that uses a PowerShell script instead of its more traditional PE executable form. In this entry, we provide in-depth analysis of the malware, as well as a detailed examination of its remote controller.

By: CH Lei, Fyodor Yarochkin, Lenart Bermejo, Philippe Lin, Razor Huang January 24, 2018
Read time: (words)

Content added to Folio