Booz Allen | CYBER4SIGHT (https://blog.cyber4sight.com/)

Analysis of Malware Used Watering-Hole Attacks Against Polish, Other Financial Institutions

by CYBER4SIGHT

UPDATE: *This report was updated on February 21 with clarifications to the executive summary.*

Executive Summary

Cyber4Sight has analyzed the malware distributed via the compromised Polish Financial Supervision Authority webpage and used in targeted attacks against a number of large banks and telecommunication companies. Cyber4Sight has identified a potential link to Russian developers, although this could easily be a false flag (http://baesystemsai.blogspot.com/2017/02/lazarus-false-flag-malware.html), and has created detection logic in the form of YARA rules for known and potentially new but related samples of the malware. Finally, although some researchers have claimed a connection between the malware used in this campaign and code used in attacks by the Lazarus Group, Cyber4Sight assesses that there are a number of theories that could account for this, and at this point, the campaign cannot be attributed with any real accuracy to either the Lazarus Group or to Russian developers.

Background

In early February 2017, a webpage belonging to the Polish Financial Supervision Authority (PFSA, knf.gov.pl) was compromised, and malicious JavaScript (JS) was injected into the page in the form of an iframe (https://niebezpiecznik.pl/post/jak-przeprowadzono-atak-na-knf-i-polskie-banki-oraz-kto-jeszcze-byl-na-celowniku-przestepcow/). This malicious JS was designed to examine the IP addresses of visitors to the site and compare those addresses to a prepopulated list of IP-address ranges that belong to several dozen large banks, credit card providers, and telecommunications companies.

If a site visitor's IP address matched one of these ranges, the JS redirected the user to a legitimate but compromised domain that hosted exploit code that then attempted to subvert Adobe Flash Player and Microsoft Silverlight browser extensions. If the exploit attempts were successful, a remote access trojan (RAT) was downloaded and installed on the victim's machine. Reports claim that the exploit code did not leverage any zero-day vulnerabilities, meaning that it targeted older, unpatched and susceptible instances of these browser extensions.

This same attack is believed to have occurred against a Mexican financial regulator, the Comision Nacional de Bancaria y Valores, and an as-yet unnamed Uruguayan state bank (most likely the Banco Central de Uruguay, which serves a similar function to the U.S. Federal Reserve Bank, and would be a site likely to be visited by bank employees, making it a good potential watering-hole site).

Technical Analysis

The malware consists of several components:

The encrypted payload—fdsvc.dll, MD5: 9cc6854bc5e217104734043c89dc4ff8

The loader, which decrypts the payload and injects it into memory—fdsvc.exe, MD5: 9914075cc687bdc352ee136ac6579707

The decrypted 64-bit payloads running in memory—MD5: 5994A8FD8C68DD1CC51CE7CA0D9C2749, 889E320CF66520485E1A0475107D7419, and 25200d3fe30785f3c90a91faf8ebf1b5

The decrypted 32-bit payload running in memory—MD5: 40e698f961eb796728a57ddf81f52b9a

Note: the 32-bit version was compiled on 2016-07-08, and the 64-bit version was compiled on 2016-08-26.

In order to decrypt the payload (fdsvc.dll, MD5: 9cc6854bc5e217104734043c89dc4ff8), the loader (fdsvc.exe, MD5: 9914075cc687bdc352ee136ac6579707) XORs each byte of the file using the previous byte before RC4 decrypting using a hardcoded key. The loader then injects the decrypted dropper into a process specified in its command-line arguments. The dropper possesses the ability to communicate with its command-and-control server (specified via the same command-line arguments issued to the dropper) via HTTPS requests.

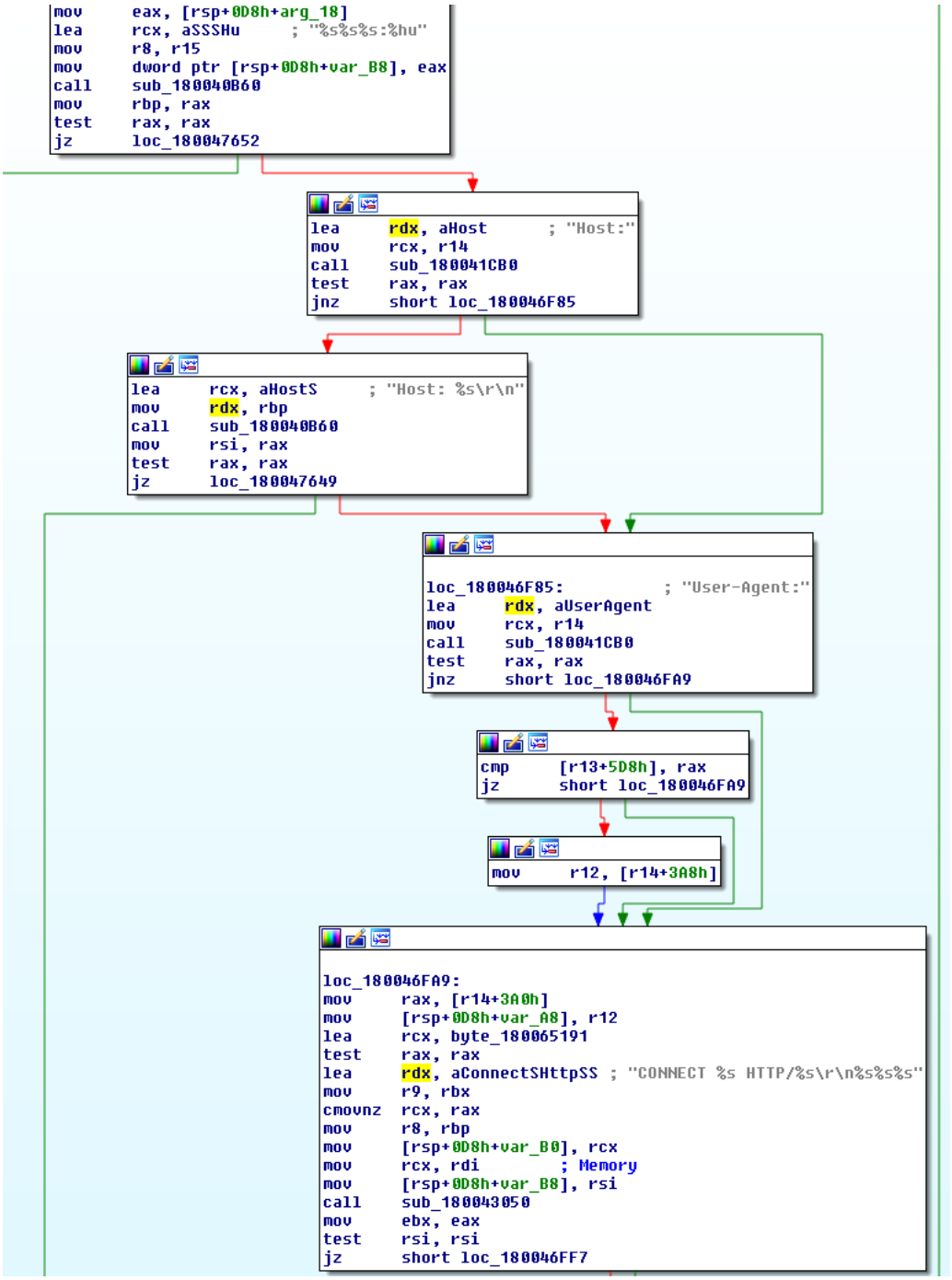

Figure 1: HTTP request-generation snippet.

The decrypted payloads (MD5: 889E320CF66520485E1A0475107D7419 and 5994A8FD8C68DD1CC51CE7CA0D9C2749) are designed to parse commands written in Russian, which is uncommon even for malware known to originate in Russia. These commands include the following:

- Ssylka ("link")
- Ustanavlivat ("install")
- Poluchit ("get")

- Pereslat ("send")
- Derzhat ("keep")
- Vykhodit ("exit")
- Nachalo ("start")

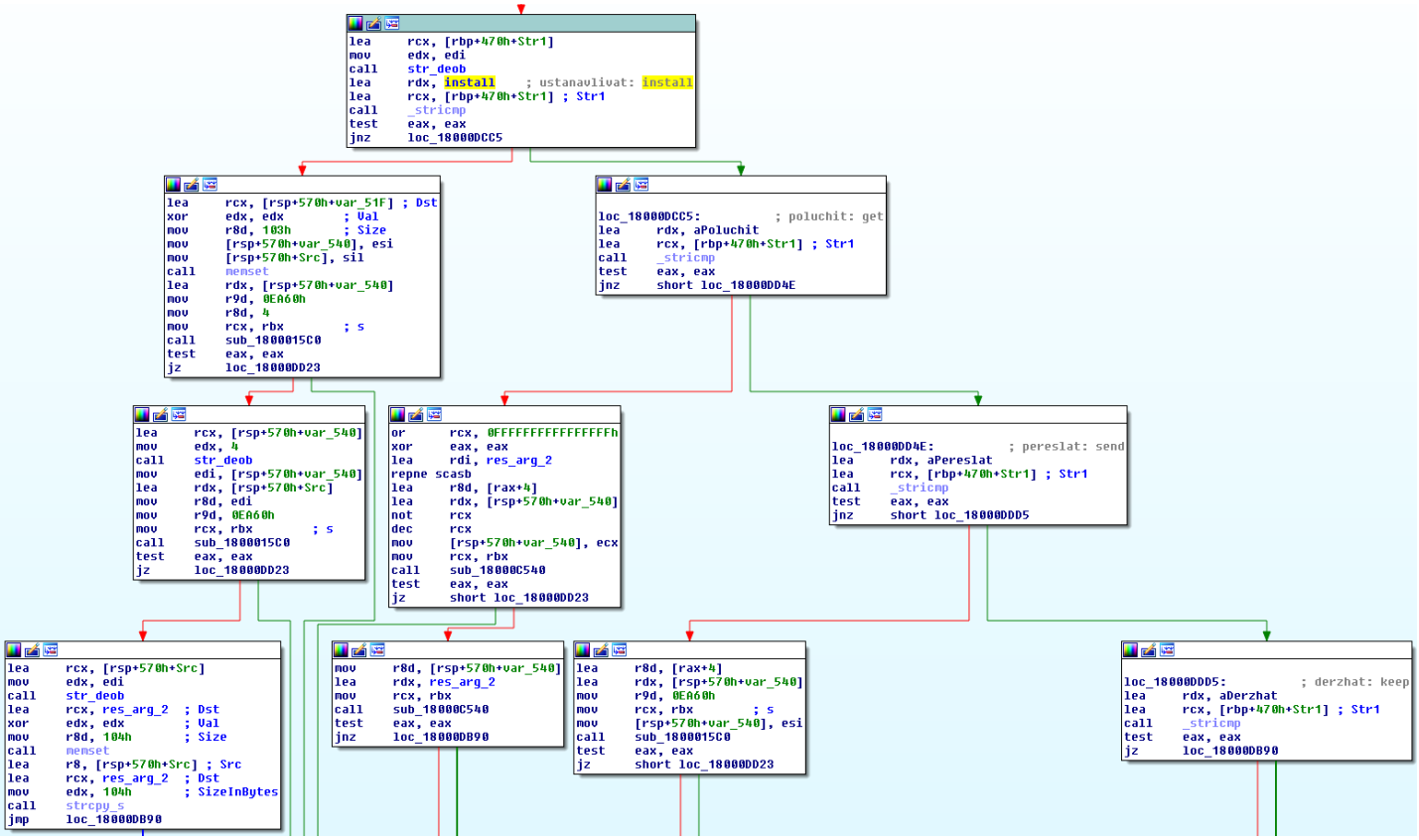

Figure 2: Logic parsing commands written in transliterated Russian.

Five malware components were identified by researchers; however, not all of these are available for analysis. Cyber4Sight only analyzed the samples it could obtain and the decrypted files that were dropped on the system by the loader.

- MD5: e29fe3c181ac9ddbb242688b151f3310 – Windows service persistence mechanism DLL (srservice.dll) [VT: 33/57 as of 2017-02-10]
- MD5: 9216b29114fb6713ef228370cbfe4045 – Injected by persistence mechanism into lsass.exe (srservice.chm)
- MD5: 9914075cc687bdc352ee136ac6579707 – Loader (fdsvc.exe) [VT: 19/56 as of 2017-02-09]
- MD5: 9cc6854bc5e217104734043c89dc4ff8 – Decrypted and injected into specified process by loader (fdsvc.dll) [VT: 2/54 as of 2017-02-07]
- MD5: 889E320CF66520485E1A0475107D7419 – Dropper (fdsvc.dll – decrypted, only extant in memory)

Attribution

A preliminary report by Symantec noted that some of the code in the "Hacktool" (which we assume is the payload, which is essentially a RAT) shares code that has been used in attacks attributed to the Lazarus Group (https://www.symantec.com/connect/blogs/attackers-target-dozens-global-banks-new-malware-0).

The Lazarus Group (aka "Who Am I?") reportedly has been active since at least 2009 (https://operationblockbuster.com/), when it conducted the MYDOOM malware campaign targeting government agencies in the United States and South Korea. The group has been linked to several high-profile attacks (https://securelist.com/blog/incidents/73914/operation-blockbuster-revealed/), including against Sony Pictures Entertainment (SPE) in November 2014 and against South Korean banks and broadcasters in the 2013 DarkSeoul and Operation Troy campaigns. The group was also implicated in the Hangman attacks on the South Korean government, and in the use of the Duuzer malware against various other South Korean organizations.

Although the Lazarus Group's attacks have predominately targeted South Korea and the United States, the group has also reportedly infected machines in Brazil, Saudi Arabia, Turkey, Iran, Russia, India, Bangladesh, China, Taiwan, Malaysia, Indonesia, and Vietnam. Notably, reports stop short of firmly attributing the group's attacks to the North Korean government. However, it is thought to have North Korean connections due to its heavy targeting of South Korea, as well as the following characteristics:

- Based on the timestamps in malware used by the group, and its communications with command-and-control servers sinkholed by security researchers, the Lazarus Group seems to operate during North Korean working hours.

- Nearly 62 percent of the portable executables reportedly contain resources citing a Korean location or written in the Korean language.

In May 2016, security researchers linked (http://baesystemsai.blogspot.co.uk/2016/05/cyber-heist-attribution.html) multiple cyber thefts against global banks (https://www.symantec.com/connect/blogs/swift-attackers-malware-linked-more-financial-attacks) via fraudulent SWIFT transfer messages to the Lazarus Group (https://www.nytimes.com/2016/05/27/business/dealbook/north-korea-linked-to-digital-thefts-from-global-banks.html), based on similarities in wiper malware code used in the thefts and previously suspected North Korean attacks. However, ultimately the evidence is too slim to conclusively determine such a connection.

Lazarus Group's motivations have appeared to be cyber espionage and data destruction. Purported ties to the North Korean government suggest that the group is acting in the government's foreign policy interests.

The group's attacks appear to focus on reconnaissance, data destruction, and data exfiltration. In past attacks, the group has used social-engineering and spear-phishing tactics to obtain credentials and infiltrate the victim's systems. Reports also suggest that the group uses network-mapping tools to further expose and move laterally inside victim networks. In the case of the 2014 SPE attack, the Lazarus Group attackers exfiltrated data from the company's network and then used wiper malware that resulted in the destruction of the Master Boot Record (MBR) of multiple machines. Eventually, a hacker group calling itself "Guardians of Peace" claimed responsibility for the attack and leaked the stolen data.

Cyber4Sight stresses that malware authors commonly borrow code from other malware variants or families when refining their own tools. Code overlap is routine, and by itself should not be the basis for determining that this campaign is the work of the Lazarus Group. This is especially true when considering that the malware analyzed by Cyber4Sight contains a number of commands written in Russian, which would be somewhat surprising if the Lazarus Group is, as some suggest, tied to the North Korean government. Several explanations are possible: the attacks may be the work of the Lazarus Group, which is part of the North Korean government but has purchased or borrowed code written by Russian malware authors; the Lazarus Group includes Russian speaking malware authors; or the attacks are the work of a different group entirely, and has borrowed code that has also been used by the Lazarus Group. It is also possible that the Russian terms were included as a false flag, to shift scrutiny towards Russian criminal actors. To that point, it is notable that the Russian terms in the malware are not written in Cyrillic characters, but are transliterated in Roman characters.

That the attackers specifically targeted several ISPs is interesting and conforms with attacks attributed to the Lazarus Group. Given similarities between the attack chain, the target set, and the Russian-language commands, it is also possible that a Russian group such as Carbanak or Buhtrap are responsible. At this point, attribution cannot be determined with any real accuracy.

Yara Rules:

The following four YARA rules can be used to identify PolishBankRAT infections:

```
rule PolishBankRAT-srservice_xorloop {
meta:
author = "Booz Allen Hamilton Dark Labs"
description = "Finds the custom xor decode loop for <PolishBankRAT-srservice>"
strings:

$loop = { 48 8B CD E8 60 FF FF FF 48 FF C3 32 44 1E FF 48 FF CF 88 43 FF }

condition:
(uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) and $loop
}

rule PolishBankRAT-fdsvc_xor_loop {
meta:
author = "Booz Allen Hamilton Dark Labs"
description = "Finds the custom xor decode loop for <PolishBankRAT-fdsvc>"

strings:

$loop = {0F B6 42 FF 48 8D 52 FF 30 42 01 FF CF 75 F1}

condition:
(uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) and $loop
}

rule PolishBankRAT-fdsvc_decode2 {
meta:
author = "Booz Allen Hamilton Dark Labs"
description = "Find a constant used as part of a payload decoding function in PolishBankRAT-fdsvc"

strings:
$part1 = {A6 EB 96}
$part2 = {61 B2 E2 EF}
$part3 = {0D CB E8 C4}
$part4 = {5A F1 66 9C}
```

```
$part5 = {A4 80 CD 9A}
$part6 = {F1 2F 46 25}
$part7 = {2F DB 16 26}
$part8 = {4B C4 3F 3C}
$str1 = "This program cannot be run in DOS mode"

condition:
(uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) and all of them
}

rule decoded_PolishBankRAT-fdsvc_strings {
meta:
author = "Booz Allen Hamilton Dark Labs"
description = "Finds hard coded strings in PolishBankRAT-fdsvc"

strings:

$str1 = "ssylka" wide ascii
$str2 = "ustanavlivat" wide ascii
$str3 = "poluchit" wide ascii
$str4 = "pereslat" wide ascii
$str5 = "derzhat" wide ascii
$str6 = "vykhodit" wide ascii
$str7 = "Nachalo" wide ascii

condition:
(uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) and 4 of ($str*)
}
```

Feature image courtesy of Flicker photo stream of Erwin T licensed under [Creative Commons (https://creativecommons.org/licenses/by/2.0/)](https://creativecommons.org/licenses/by/2.0/).

#banks (https://blog.cyber4sight.com/tag/banks/), #financial industry (https://blog.cyber4sight.com/tag/financial-industry/), #malware analysis (https://blog.cyber4sight.com/tag/malware-analysis/), #polish banks (https://blog.cyber4sight.com/tag/polish-banks/), #technical analysis (https://blog.cyber4sight.com/tag/technical-analysis/), #watering hole attacks (https://blog.cyber4sight.com/tag/watering-hole-attacks/)

Related articles

January 4, 2017 February 21, 2017 December 19, 2016

FTC to Award USD 25,000 For Tech Solution to OutdatedMalicious PowerShell Scripts Analysis Suggests Shamoon Actors Used Pupy RATMarked Increase in WordPress Brute-Forcing Attacks From Obscure Uk

In what they've dubbed "the Internet of Things Home Inspector Challenge," the… analysis of Shamoon-related spear-phishing documents, researchers from IBM… Between 24 November and the publication of a 16 December report, th

(https://blog.cyber4sight.com/2017/01/ftc-to-award-usd-25000-for-tech-solution-to-outdated-vulnerable-iot-devices/)(https://blog.cyber4sight.com/2017/02/malicious-powershell-…)(https://blog.cyber4sight.com/2016/12/marked-increase-in-wordpress-brut

Share this article

SHARE ON FACEBOOK (HTTP://WWW.FACEBOOK.COM/SHARE.PHP?U=HTTPS://BLOG.CYBER4SIGHT.COM/2017/02/TECHNICAL-ANALYSIS-WATERING-HOLE-ATTACKS-AGAINST-FIN

SHARE ON TWITTER (HTTPS://TWITTER.COM/SHARE?URL=HTTPS://BLOG.CYBER4SIGHT.COM/2017/02/TECHNICAL-ANALYSIS-WATERING-HOLE-ATTACKS-AGAINST-FINANCIAL-INS

SHARE ON PINTEREST (HTTP://PINTEREST.COM/PIN/CREATE/BUTTON/?URL=HTTPS://BLOG.CYBER4SIGHT.COM/2017/02/TECHNICAL-ANALYSIS-WATERING-HOLE-ATTACKS-AGAI

← **Previous Post** (https://blog.cyber4sight.com/2017/02/finding-a-better-metaphor-staring-into-the-cloud-and-what-to-expect-at-rsa-2017/)

All Posts (https://blog.cyber4sight.com/2017/02/technical-analysis-watering-hole-attacks-against-financial-institutions/)

Next Post → (https://blog.cyber4sight.com/2017/02/strong-correlation-between-cards-in-jokers-stash-cashflow-and-arbys-locations/)

(https://twitter.com/cyber4sight) (https://www.linkedin.com/company/cyber4sight) (https://mail.google.com/mail/?view=cm&fs=1&tf=1&to=support@cyber4sight.com)