ner-success) Blog (https://www.anomali.com/blog) Forum (https://forum.anomali.com/) EN

2869291aeb94595.1584552311223.1584552311223.1584552311223.1&__hssc=41179005.1.1584552311224&__hsfp=824236545) (https://ui.threatstream.com/login?__hstc=41179005.6a853dd8017d4c4f12869291aeb9459

ANOMALI (https://www.anomali.com/)

CYBER THREAT INTELLIGENCE (/BLOG/CATEGORY/CYBER-THREAT-INTELLIGENCE)

MALWARE (/BLOG/CATEGORY/MALWARE)

Evidence of Stronger Ties Between North Korea and SWIFT Banking Attacks

May 27, 2016 | Aaron Shelmire

Five new additional pieces of malware code discovered that contain unique portions of code related to the the SWIFT attacks.

Recently, malware analysts at Symantec (http://www.symantec.com/connect/blogs/swift-attackers-malware-linked-more-financial-attacks) discovered two subroutines that were shared amongst North Korea's Lazarus' groups Operation Blockbuster malware and two samples of malware from the recent SWIFT attacks.

The shared subroutines are displayed as evidence to relate the SWIFT intrusion activity to the Lazarus group. Symantec's analysis was utilized in the The New York Times story (http://www.nytimes.com/2016/05/27/business/dealbook/north-korea-linked-to-digital-thefts-from-global-banks.html?_r=0) on May 27, 2016. Their findings supported a claim that these were the only two pieces of software with this shared code.

The Anomali Labs team has conducted deeper research into a very large malware data repository. This process utilized the yara signature below to search for the shared subroutines. At first, we believed it would produce a lot of false positives. Instead, this search not only failed to result in any false positives, but also turned up five other pieces of malware which share this code. We see this as a possible attribution of the Lazarus group attacks to other attacks that involved these same five pieces of malware code.

Malware Family	Md5 hash	Notes
SWIFT BanSwift	5d0ffbc8389f27b0649696f0ef5b3cfe	evchk.bat dropper
SWIFT Fake Foxit Reader	0b9bf941e2539eaa34756a9e2c0d5343	A Fake Foxit Reader submitted to Virustotal from Vietnam in December 2015 (similar sample detailed at https://blogs.mcafee.com/mcafee-labs/attacks-swift-banking-system-benefit-insider-knowledge/ (https://blogs.mcafee.com/mcafee-labs/attacks-swift-banking-system-benefit-insider-knowledge/))
SMBWorm	558b020ce2c80710605ed30678b6fd0c	Known North Korean Malware
Memory dump with SMBWorm	96f4e767aa6bb1a1a5ab22e0662eec86	
Unknown “hkcmd” tool	b0ec717aeece8d5d865a4f7481e941c5	1st Submitted from Canada, likely from an AV organization. 2016/04/22. PE Build Date of December 2010.
imkrmig.exe	5a85ea837323554a0578f78f4e7febd8	An unknown backdoor posing as a Korean sample of Microsoft Office 2007.

Table 1. Malware families and samples known to include the Lazarus Wipe File routine.

Our approach to code comparison was to utilize Position Independent Code function hashes to compare the samples against one another. This process utilizes cryptographic hash values derived from the instruction mnemonics within the binary code. By performing this comparison, we can see the direct overlap of these shared functions between the various samples.

Figure 1: The function overlap viewed from ae086350239380f56470c19d6a200f7d251c7422c7bc5ce74730ee8bab8e6283 as veiwed within IDAPro

Additionally, there are other function hashes (seven) that are shared amongst the Trojan.Filmis and various SWIFT-related malware samples. Anomali LABS is unsure of how rare these functions are at this point.

Investigative Process

We began by taking a look at the two subroutines that are reported to be unique by Symantec. We retrieved the API names and added those to a yara signature. In some cases, the APIs are MoveFileExA instead of MoveFileEx.

We then took a look at the code used. There is a small portion of code where a file name consisting of randomly generated lowercase letters is created. This was used as part of the criteria.

Using this criteria, we began a search of a large malware database starting on Thursday night. On Friday morning, we thought we'd be faced with a sea of false positives. But it only returned 10 matches! Four of those were known samples of the SWIFT malware, and one sample was a zip file that includes a known SWIFT sample. The other five samples are detailed above.

Appendix

Additional Samples related to the SWIFT intrusions (ref: http://baesystemsai.blogspot.com/2016/04/two-bytes-to-951m.html (http://baesystemsai.blogspot.com/2016/04/two-bytes-to-951m.html))

Filename	**md5**	**AntiVirus Name**
evtsys.exe	5d0ffbc8389f27b0649696f0ef5b3cfe	BanSwift
evtdiag.exe	24d76abbc0a10e4c977a28b33c879248	BanSwift
nroff_b.exe	1d0e79feb6d7ed23eb1bf7f257ce4fee	BanSwift
gpca.dat	f7272bb1374bf3af193ea1d1845b27fd	
mspdclr.exe	909e1b840909522fe6ba3d4dfd197d93	BanSwift

Other previously known Lazarus Group samples:

```
138464214c78a73e3714d784697745acbf692ef40419d31418e4018e752cb92b
bdcfa3b6ca6b351e76241bca17e8f30cc8f35bed0309cee91966be9bd01cb848
ddebee8fe97252203e6c943fb4f9b37ade3d5fefe90edba7a37e4856056f8cd6
4d4b17ddbcf4ce397f76cf0a2e230c9d513b23065f746a5ee2de74f447be39b9
e2ecec43da974db02f624ecadc94baf1d21fd1a5c4990c15863bb9929f781a0a
eff542ac8e37db48821cb4e5a7d95c044fff27557763de3a891b40ebeb52cc55
f6cb8343444771c3d03cc90e3ac5f76ff9a4cb9cd41e65c3b7f52b38b20c0c27
```

```
rule AnomaliLABS_Lazarus_wipe_file_routine {
 meta:
    author = "aaron shelmire"
    date = "2015 May 26"
    desc = “Yara sig to detect File Wiping routine of the Lazarus
group”
 strings:
    $rand_name_routine = { 99 B9 1A 00 00 00 F7 F9 80 C2 61
88 16 8A 46 01 46 84 C0 }
    /* imports for overwrite function */
    $imp_getTick = "GetTickCount"
    $imp_srand = "srand"
    $imp_CreateFile = "CreateFileA"
    $imp_SetFilePointer = "SetFilePointer"
    $imp_WriteFile = "WriteFile"
    $imp_FlushFileBuffers = "FlushFileBuffers"
    $imp_GetFileSizeEx = "GetFileSizeEx"
    $imp_CloseHandle = "CloseHandle"
    /* imports for rename function */
    $imp_strrchr = "strrchr"
    $imp_rand = "rand"
    $Move_File = "MoveFileA"
    $Move_FileEx = "MoveFileEx"
    $imp_RemoveDir = "RemoveDirectoryA"
    $imp_DeleteFile = "DeleteFileA"
    $imp_GetLastError = "GetLastError"
condition:
    $rand_name_routine and (11 of ($imp_*)) and ( 1 of
($Move_*))
}
```

About the Author

Aaron Shelmire

Aaron began work in the security field after machines he was responsible for were compromised in the 2004 Stakkato Intrusions. At this point he went to graduate school at Carnegie Mellon Universities Heinz College for Information Assurance, where he currently holds an adjunct position teaching Network Security Analysis. He has been a security researcher at the Software Engineering Institutes CERT/CC initiative and Dell SecureWorks, with a focus on responding to and analyzing threat intelligence.

You might also be interested in...

(https://www.anomali.com/blog/
attack-when-the-herd-is-

(https://www.anomali.c
threat-actors-that-may

distracted)

Blog

Wolves Attack When the Herd Is Distracted (https://www.anomali.com/ attack-when-the-herd-is-

increase-hostile-activit to-elimination-of-irania general-quassem-suleir

Blog

APTs & Threat Actor

Get the latest threat intelligence news in your email.

Company Email

SUBSCR

Privacy Policy (https://www.anomali.com/privacy-policy)

Terms of Use (https://www.anomali.com/terms-of-service)

3rd Party Vendor Policy (https://www.anomali.com/3rd-party-vendor-policy)