Updated Blackmoon banking Trojan stays focused on South Korean banking customers

proofpoint.com/us/threat-insight/post/Updated-Blackmoon-Banking-Trojan

January 18, 2016

Updated Blackmoon banking Trojan stays focused on South Korean banking customers

January 19, 2016 Proofpoint Staff

First analyzed in early 2014 [1] [2], the Blackmoon banking Trojan targets a user's online banking credentials using a type of pharming that involves modifying or replacing the local Hosts file with one that redirects online banking domain lookups to an IP address controlled by the attacker. Blackmoon has been observed targeting primarily customers of South Korean online banking sites and services, and is usually distributed via drive-by download.

Like other banking Trojans targeting online banking users in other countries, Blackmoon continues to evolve both their distribution and delivery technique. Proofpoint threat researchers recently observed a sample of the Blackmoon Korean banking Trojan (7e67216628d9a171be0ce18c51fda8ce) retrieving encoded configuration information from the "lofter[.]com" blogging platform.

Figure 1: Encoded configuration block found on the lofter[.]com blogging platform

Executable Process

The malware arrives on the system as an executable. (The original delivery method of this sample is unknown, but previous analyses have found it spreading via download from an infected site.)

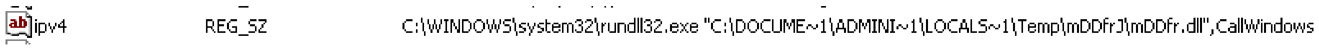
Figure 2: Information from PEStudio for 7e67216628d9a171be0ce18c51fda8ce

Once executed, the dropper extracts a DLL from itself and launches the DLL via rundll32.exe, calling the dropper filename as a parameter that deletes the original dropper from disk. The DLL and the folder it resides in are named using 6 random alphabetical characters.

Figure 3: BlackMoon DLL (84E2D574085C77F47E801F5326E83D73) launching via rundll32.exe

Figure 4: Blackmoon strings present in 84E2D574085C77F47E801F5326E83D73

The malware sets persistence by running the command



```
ipv4    REG_SZ    C:\WINDOWS\system32\rundll32.exe "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\mDDfrJ\mDDfr.dll",CallWindows
```

Figure 5: Blackmoon DLL persistence setting in registry key

The malware calls out to a hardcoded website address to retrieve the encoded configuration block. It parses the page looking for “###” and extracts the data until a subsequent “###”: this is the encoded configuration block (Fig. 6).

Figure 6: The encoded Blackmoon configuration block.

The malware makes use of JavaScript to handle the encoding and decoding of strings. The decoding can be described as case-swapped base64 with a substituted padding character of ‘@‘. (A python script for decoding is included at end of this post.).

Figure 7: Calling decoding routine in malicious JavaScript

Figure 8: Decoded Configuration Block

The malware also utilizes this encoding to register the infected host.

Figure 9: Blackmoon infected client check-in

Figure 10: Decoded SID parameter for infected client

The malware clears the DNS resolver cache by running the command ipconfig.exe /flushdns, and then sets the values found in the [Dns] section of the config block to the infected client’s DNS server settings. In this case it is 127.0.0.1 with the Google primary public DNS server of 8.8.8.8 as a backup.

With the primary DNS set to the loopback address, the malware rewrites the infected client’s Hosts file with search engine and banking sites that will resolve to the attacker's server. The list of domains to redirect is hardcoded in the malware. The value in the [Host] field of the configuration block replaces “IP#“ when it is written to the Hosts file.

Figure 11: The Modified HOSTS file before (right) and after (left) the IP from the Configuration Block is replaced.

The malware also searches for a folder “\NPKI\” containing .cer or .der files on the infected client. If found, the malware extracts a command line RAR executable from itself and saves it to “\Documents and Settings\Administrator\Local Settings\Temp” as “zip.tmp”. Then the malware archives the NPKI folder with the following command:

```
zip.tmp a -hp@#@2016999# "D:\NPKI.z" "D:\NPKI"
```

The malware will then send the password protected archive to the value of the [Upload] field in the configuration block via HTTP POST. It is likely that the content of the NPKI folder are used to impersonate an end-user in order to access their online banking information and accounts. [3] [4]

Figure 12: The Blackmoon DLL sending NPKI RAR archive

The configuration block also includes a [Time] value that indicates how often the malware checks for a new configuration block.

Once the infected client’s Hosts file has been updated with the redirected domains, a user visiting any of the search engine sites with the infected client will often see the following message:

Figure 13: Message displayed when user of infected client visits a domain listed in modified Hosts file

This message roughly translates to:

Financial Supervisory Service is conducting an authentication process did you install the security certificate for this PC?

※ Certificate must verify the security and privacy of information leakage incidents in the auction using Internet banking Guests prevent financial fraud, please see below.

※ You cannot access the Internet Banking more safely receive the security certification process.

※ Please click the bank name that you use to proceed to the secure authentication procedures.

A user who clicks on one of these online banking site names in order to login would be presented with the following sequence pages:

Figure 14: Fake online banking web site using stolen branding

While this page appears legitimate, clicking on any element on the page brings up an alert.

Figure 15: Fraudulent security notification to end user

This roughly translates to:

Safer internet 03.24.2014 (Will) for banking using internet banking, smart Banking, Phone Banking To The use of all services (private. Company) can then use additional authentication.

Clicking the OK button brings up a brief loading message:

Figure 16: Loading message displayed when user accepts warning

Which is then followed by a two-stage credential theft phish.

Figure 17: First page of credential theft phishing, with stolen branding

Figure 18: Second page of credential theft phishing, with stolen branding

The phishing pages include user input validation capabilities that surpass those typically found in phishing pages, such as validating that the user has entered their name using the proper character set, as well as numerous checks to ensure that a valid "Social Security number" (officially called a Resident Registration Number in South Korea) has been entered.

Figure 19: Phishing page form input validation for character set

Figure 20: Phishing page form input validation for South Korean equivalent of Social Security Number

This infection chain is consistent with examples documented in early 2014, with two noteworthy updates to distribution and delivery technique:

- Distribution via the lofter[.]com blogging platform, with frequent changes to the backend web servers.
- Addition of encoding for C2 information in payload download.

These changes are consistent with updates Proofpoint researchers have recently observed in other malware, from highly targeted malware such as the Operation Arid Viper payload [5] to broad-based campaigns such as those distributing Dridex [6]. Organizations can expect to see continued variation in obfuscation techniques and distribution as attackers attempt to stay ahead of evolving defenses.

References

[1] http://training.nshc.net/ENG/Document/virus/20140305_Internet_Bank_Pharming_-_BlackMoon_Ver_1.0_External_ENG.pdf

[2] https://zairon.wordpress.com/2014/04/15/trojan-banking-47d18761d46d8e7c4ad49cc575b0acc2bb3f49bb56a3d29fb1ec600447cb89a4/

[3] http://www.hikorea.go.kr/pt/PublicCertificate_en.pt

[4] http://grrrltraveler.com/countries/asia/korea/expat-life/online-banking-korea/

[5] http://www.proofpoint.com/us/threat-insight/post/Operation-Arid-Viper-Slithers-Back-Into-View

[6] http://www.proofpoint.com/us/threat-insight/post/Not-Yet-Dead

Blackmoon DLLs

Name: yQkUz.dll

MD5: 84e2d574085c77f47e801f5326e83d73

SHA256: ad062b7cba8f149a585018938b45f65698dde3a049a6f50fd4e355e68b562fc3

Compile Time: 2016-01-07 18:08:45

Name: pkNQy.dll

MD5: 9be8a5edc5f0a57d09b733c18a3740c7

SHA256: 4e94d38c1939ca7c6928da062b01e381e7a925ae4c66945f598f090c8d79a6a0

Compile Time: 2016-01-10 05:09:38

Name: OUikm.dll

MD5: 255fd48dd681058d9cb84e4c6dbd92f6

SHA256: 8fbacfa948ba95cbe7e6f44a7974f621259a0c23c43a4a4c3d8e3e163604388a

Compile Time: 2016-01-09 09:43:13

Name: uyonu.dll

MD5: 949482b0aa3ecc019d63d10a46539302

SHA256: df821948e3362a5accdefc444b4bdf8e370f77af65fabbbd371cd95d1c181347

Compile Time: 2016-01-07 18:08:45

Blackmoon EXE Droppers

Name: sa.exe (Analyzed)

MD5: 7e67216628d9a171be0ce18c51fda8ce

SHA256: f0dd2eeaaeb85ab98f0d2d04151b7a56fa3d0c427e9356049cbc4f41bcfabf72

Compile Time: 2016-01-07 18:25:05

Name: smss.exe

MD5: 86a16809fe21cc389740866dfc73abe3

SHA256: 6f250727e69716776f3bb594715a7e10bea65e35556f1dc3a922b63f40611b39

Compile Time: 2016-01-11 06:09:40

Name: smss.exe

MD5: 091bb8f755f7eda753e53b0b6501dcb2

SHA256: af777fe3a147a48185c65ecd750be0863caa6fbcc51a75a4fb944a651c875006

Compile Time: 2016-01-11 06:09:40

Name: smss.exe

MD5: b67e98a8c3f2b46207e9b9d4785bbe4cSHA256:

SHA256: fa2ddd90683ddcc968d5349edfd85d81dd0035daf7f0d2d7556c8c609fb78554

Compile Time: 2016-01-11 06:09:40

Name: smss.exe

MD5: 80abc3ad344c4999f33948c8a241223c

SHA256: c52c5dde2071754b54414fe0035d28145e212aa116917f8d2794169b5def2966

Compile Time: 2016-01-11 06:09:40

Name:

MD5: 3fe1d163b22c619d8e9dd865d83d9b05

SHA256: 8189b3e021be392d4a731d68c5c73d2bafb8168b70351be7475806b8978304aa

Compile Time: 2016-01-11 06:09:40

Name: smss.exe

MD5: c973ac06f36f1b52a08c51faf79fade2

SHA256: e3ccb1c511a18b0b95f51ca54b2bde109eb689b9ef23ad9325ab7c58fd3bd857

Compile Time: 2016-01-11 06:09:40

Name: smss.exe

MD5: 34f4257bba25546aaf486132c27c40d5

SHA256: 5881f66242ceb03f85731dafbab272e88545609bb2542a4328a2060c4ecedc85

Compile Time: 2016-01-11 06:09:40

Name: smss.exe

MD5: 79fee38ebc1c6db755f3da38287349f9

SHA256: c5d95003eb571199ddb6f5c181ab6fc326115c55ff2637673657d946bf314f87

Compile Time: 2016-01-11 06:09:40

Name:

MD5: 7d091ae970c41b85e9a281308fab6985

SHA256: 8b36c161d720926a91e1d2324fe075b740b782100833d01c7905e4ccef5befc6

Compile Time: 2016-01-11 06:09:40

Name:

MD5: 4f21078383c7fff2ad3dbe8b77de7f3d

SHA256: 43bac8196a8410b09e0ab1a2926ad9419b32ea7caa8371585db26748f09418b0

Compile Time: 2016-01-09 09:26:10

Name:

MD5: dd01534e1a78913f440d30bf03d99462

SHA256: f07d0ceb105b5454d8037283667f4103e19413b3297885d38db94031cc14c258

Compile Time: 2016-01-09 09:26:10

Name:

MD5: fe0fca87d2a1ef1b7d0c57414dee32be

SHA256: b6340b9f2433bd20246719e92870e3f1ec01d42a0e22606f27ee53b7fe0adafe

Compile Time: 2016-01-09 09:26:10

Name:

MD5: 371b63fb512513c066e541a13f3ed79a

SHA256: e4b8adcf2974abbe236813b02b507280bca61f8a0795e3901c8718999c661cd5

Compile Time: 2016-01-09 09:26:10

Name: dll.exe

MD5: a4a2d0a47aa3c1bc4382997a197e3aeb

SHA256: 437c8a5639149fd97943f01ee88aa96f131b9755172c77a42a57c014d0158fbe

Compile Time: 2016-01-09 09:26:10

Name: ser.exe

MD5: 4c8f4bd321ebde0698576c4b1a788773

SHA256: f8aa625dd544f3e49412f9a2acea411c8cb4b6f346e04800ff711c9cd9a45d92

Compile Time: 2016-01-09 09:26:10

Name: dll.exe

MD5: 59596e9c4c94ebd7d5a692a782623560

SHA256: 16e922193fb53d58c44e8cb012fe1d19bfba391807db964fe2c6cde06a436aa1

Compile Time: 2016-01-09 09:26:10

Name: sa.exe

MD5: 255e5a9dfc352e9abdfe67e00e6d34ef

SHA256: 8d6b2be9180972274d8111a47e0a15d5158bfd352cd5738a665d14c71a8406e9

Compile Time: 2016-01-07 18:25:05

Name: sa.exe

MD5: 5fac43273dc8a7bed3a005220d32da1d

SHA256: ada60b73629c135592fef0f7257cd1dac8e0cb4a448141b2b2e3e1bba02c5eab

Compile Time: 2016-01-07 18:25:05

Name: sa.exe

MD5: 35732507edc006ce63066f59cee041b8

SHA256: 0fc932e2dae7219cc5a14a224e76385ba7e15e15fa0fb4054206efbf983cea00

Compile Time: 2016-01-07 18:25:05

Name:

MD5: ab9278dbc583d4829524e68f101c0de1

SHA256: 46e572338ea5c1c691ab60984abfc38007c7cfd7b6e77adf26a6bbaa22451d73

Compile Time: 2016-01-07 18:25:05

Name:

MD5: 25e02fe76649535abed4c3f1340ba88c

SHA256: abe2f051bc9339d2e0c29ee75027879d465d644de8a3159ada1db72412a551b8

Compile Time: 2016-01-07 18:25:05

Name: sa.exe

MD5: c967d619404bd371a75ba4c5ca2a650a

SHA256: eb6e0e39bc2c379e18076cae7da2dbfb23233294ebd24b40a01180dc768e092e

Compile Time: 2016-01-07 18:25:05

IP Addresses

100.43.129[.]107

98.126.19[.]178

174.139.200[.]164

174.139.200[.]165

174.139.203[.]180

Emerging Threats Coverage

2815665 - ETPRO TROJAN W32.Blackmoon Checkin 1

2815676 - ETPRO TROJAN W32.Blackmoon Checkin 2

2815733 - ETPRO TROJAN W32.Blackmoon Bank Phishing Page - Compromised Host M1

2815734 - ETPRO TROJAN W32.Blackmoon Bank Phishing Page - Compromised Host M2

2815735 - ETPRO TROJAN W32.Blackmoon Bank Phishing Page - Compromised Host M3

2815736 - ETPRO TROJAN W32.Blackmoon Bank Phishing Page - Compromised Host M4

2815737 - ETPRO TROJAN W32.Blackmoon Bank Phishing Page - Compromised Host M5

2815738 - ETPRO TROJAN W32.Blackmoon Bank Phishing Page - Compromised Host M6

2815769 - ETPRO TROJAN W32.Blackmoon Uploading Stolen Certificates

Yara Rule

```
rule BLACKMOON_BANKER {
    meta:
        author = "Proofpoint Staff"
        info = "blackmoon update"
        strings:
                $s1 = "BlackMoon RunTime Error:" nocase wide ascii
                $s2 = "\\system32\\rundll32.exe" wide ascii
                $s3 = "cmd.exe /c ipconfig /flushdns" wide ascii
                $s4 = "\\system32\\drivers\\etc\\hosts.ics" wide ascii
        condition:
                all of them
}
```

Python Script to Decode

```
#!/usr/bin/env python2

from base64 import b64decode

from binascii import unhexlify

import sys

def main():

    if len(sys.argv) < 2:

        print 'Usage: ' + sys.argv[0] + ' [data_to_decode]'

        exit(-1)

    data = sys.argv[1].strip('#')

    data = data.replace('@','=')

    data = data.swapcase()

    data = b64decode(data)

    try:

        int(data, 16)

        print "\n" + unhexlify(data)

    except ValueError:

        print "\n" + data + "\n"

if __name__ == '__main__':

    main()
```

Subscribe to the Proofpoint Blog