APT Down: The North Korea Files

AUTHOR: Saber / cyb0rg
5bc524352881851934d4a88eb8c1682c

Table of Contents

0 Introduction

This article analyses the dump of data from a APT's workstation. In particular the data and source code retrieved from the workstation belonging to threat actor actively targeting organizations in South Korea and Taiwan.

We believe this to be a member of North Korea's "Kimsuky" group [#14].

"Kimsuky is a North Korean state-backed Advanced Persistent Threat that targets think tanks, industry, nuclear power operators and government for espionage purposes. It is being designated pursuant to E.O. 13687, for being an agency, instrumentality, or a controlled entity of the Government of North Korea."

We refer to this particular member as "KIM" for the sake of this article.

KIM is not your friend.

The dump includes many of Kimsuky's backdoors and their tools as well as the internal documentation. It shows a glimpse how openly "Kimsuky" cooperates with other Chinese APTs and shares their tools and techniques.

Some of these tools may already be known to the community: You have seen their scans and found their server side artifacts and implants. Now you shall also see their clients, documentation, passwords, source code, and command files...

As a freebie, we also give you a backup of their VPS that they used for spear-phishing attacks.

This article is an invitation for threat hunters, reverse engineers and hackers, - Enjoy.

The meat of the article is split into 3 parts:

-- 1.x The dumps, log files, history files, password lists,
-- 2.x Their backdoors, tools, payloads,
-- 3.x OSINT on the threat actor

The dump is available at:

ONION

PROTON

F Dear Kimsuky, you are no hacker

What defines a Hacker? Somebody clever, extremely clever, who enjoys using technology beyond its intended purpose and who does so without causing harm, is free of any political agenda and has no monetary incentives. They take no money and no rewards. They follow nobody and have no goal beyond expressing their creativity.

An artist.

Kimsuky, you are not a hacker. You are driven by financial greed, to enrich your leaders, and to fulfill their political agenda. You steal from others and favour your own. You value yourself above the others: You are morally perverted.

I am a Hacker and I am the opposite to all that you are. In my realm, we are all alike. We exist without skin color, without nationality, and without political agenda. We are slaves to nobody. I hack to express my creativity and to share my knowledge with other artists like me. To contribute, share, and further the knowledge of all man kind. For the beauty of the baud alone.

You hack for all the wrong reasons.

1 The Dumps

>>> Be mindful when opening files from the dump. <<<
>>> You have been warned. <<<

This paragraph gives a short overview of the dumps and then takes a closer look at three initial findings:

- Logs showing an attack against The Defense Counterintelligence Command
- Access to the South Korea Ministry of foreign Affairs
- Access to internal South Korean Gov network
- ...and many more files we did not had the time yet to look at. #ENJOY

The first dump is from KIM's guest VM and the second is from his public VPS. Both dumps were retrieved around the 10th of June 2025.

The first dump:

- A screenshot of his Desktop (kim_desktop.jpg).
- Linux Dev System (VM, running Deepin 20.9 Linux).
- The guest VM had the host's C:\ mounted (hgfs). Dumped included.
- A listing of all files can be found in ./file-lists.
- About 20,000 entries in the Brave & Chrome history. Revealing many email
- addresses (jeder97271@wuzak.com, xocaw75424@weiby.com, ..), sites KIM
- visited and tools KIM downloaded. All Chrome extensions such as spoofing
- the User-Agent, Proxy SwitchyOmega, a Cookie Editor and many others.
- The file `ko 图文编译 .doc` is a manual how to operate one of their backdoors. There is also a very officially sounding statement(translated): "it is forbidden to use the backdoor outside of its designated use".
- Lots of passwords in `mnt/hgfs/Desktop/fish_25327/vps20240103.docx`.
- Including E-Mail and VPS passwords (working).
- root / 1qaz2wsx
- dysoni91@tutamail.com / !QAZ4rfv!@#$
- https://sg24.vps.bz:4083 / center2025a@tutamail.com / H4FHKMWMpX8bZ
- https://monovm.com / dysoni91@tutamail.com / dr567h%a"G6*m
- See fish-url.txt & generator.php to learn about password re-use patterns.

The second dump:

- Server name: vps1735811325, hosted at vps.bz
- Server was used for various spearphising campaigns

- Noticeable are the SSL certificates and auth.log. The source code for phishing attacks are discussed further below.

1.1 - Defense Counterintelligence Command (dcc.mil.kr)

Drop Location: vps/var/www/html/

The Defense Counterintelligence Command (DCC) is an intelligence organization of the South Korean Armed Forces. The DCC is primarily responsible for intelligence missions such as clandestine and covert operations, and counterintelligence.

The logs show a phishing attack against the dcc.mil.kr as recently as three days ago.

The same logs contain The Supreme Prosecutor Office (spo.go.kr), korea.kr, daum.net, kakao.com, and naver.com. It should be noted that the Admin-C for dcc.mil.kr is registered to hyuny1982@naver.com.

```
grep -Fhr 'dcc.mil.kr' log | uniq
jandy3912@dcc.mil.kr_amFuZHkzOTEyQGRjYy5taWwua3I=
di031111@dcc.mil.kr_ZGkwMzExMTFAZGNjLm1pbC5rcg==
didcdba@dcc.mil.kr_ZGlkY2RiYUBkY2MubWlsLmty
jhcgod88@dcc.mil.kr_amhjZ29kODhAZGNjLm1pbC5rcg==
chanchan0616@dcc.mil.kr_Y2hhbmNoYW4wNjE2QGRjYy5taWwua3I=
yib100@dcc.mil.kr_eWliMTAwQGRjYy5taWwua3I=
Dsc808@dcc.mil.kr_RHNjODA4QGRjYy5taWwua3I=

[...]
```

The tools used in this attack are discussed under 2.1 (Generator).

1.2 - Access to South Korea Ministry of foreign Affairs repository

A copy of South Korean Ministry of foreign affairs email platform was found inside a file named: mofa.go.kr.7z. The source code was likely taken very recently:

```
 1923 Apr  1 07:15 .gitignore
   96 Apr  1 07:15 .gitmodules
 4096 Apr  1 07:15 kebi-batch/
 4096 Apr  1 07:15 kebi-core/
 4096 Apr  1 07:15 kebi-resources/
 4096 Apr  1 07:15 kebi-web-admin/
 4096 Apr  1 07:15 kebi-web-archive/
 4096 Apr  1 07:15 kebi-web-mail/
 4096 Apr  1 07:15 kebi-web-mobile/
 4096 Apr  1 07:16 kebi-web-parent/
 7528 Apr  1 07:16 pom.xml
14099 Apr  1 07:15 README.txt
```

Given the format of the files, this is probably a dump from a GitHub repository which appears to be parts of an email server. The source code contains multiple references to government domains:

```
./kebi-web-parent/mail/document/info.txt
/home/ksign/agent
http://email.mofa.go.kr:8080/mail/sso?type=login
http://mail.mofa.go.kr:8080/mail/sso?type=unseenMails
http://email.mofa.go.kr:8190/mail/sso?type=login
http://mail.mofa.go.kr:8080/mail/sso?type=unseenMails
```

1.3 - Access to the internal South Korean Gov network

It appears that KIM maintains access to internal South Korean Government Network systems. There is a project named onnara_auto, which contains several interesting files.

The project appears to be tools to query internal government servers. For instance, a file named: /onnara_auto/log/log-20250511.log has the following entries:

```
[horedi179] get onnara9.saas.gcloud.go.kr at 11/05/2025 19:41:23

[horedi179] main_job:Session 6112b9bc-5a2a-4abd-a907-aaec4b19e2ed does not exist at 11/05/2025 19:41:23
[horedi179] get onnara9.saas.gcloud.go.kr at 11/05/2025 19:41:23
[horedi179] get https://onnara9.saas.gcloud.go.kr/ at 11/05/2025 19:45:37
[horedi179] main_job:Session 0c446a8c-e913-467d-a9b9-3f08abfb6f7a does not exist at 11/05/2025 19:45:37
[horedi179] get https://onnara9.saas.gcloud.go.kr/SSO.do at 11/05/202...
```

The corresponding code:

```
drives = instanceManger(config_hub)
client = Client(config_hub)
plugins = PluginManager()
try:
  onnara = onnara_sso("horedi79", "", "", "1250000","onnara9")

  klass = plugins.load(os.path.join(os.getcwd(),
        "scripts", target_project, "onLaunch.py"),
         opts={'onnara':onnara,'drives': drives, "client": client})
```

The hostname 'onnara9.saas.gcloud.go.kr' is not accessible from the public Internet, however the domain name appears in some documents mentioned as an internal government portal. KIM seems to have access to this network.

1.4 Miscellaneous

His origin IP was 156.59.13.153 (Singapore). The IP has SSHD running on port 60233 and port 4012 shows a TLS certificate with CN=*.appletls.com.Fofa shows around 1,100 uniq IP addresses with that certificate. Most (>90%) are located in China and HK. These may be some VPN proxy network or Operational Relay Boxes (ORBs). (Similar to "Superjumper" and [#15])

On the 13th of June 2025, KIM registered webcloud-notice.com. We believe this to be in preparation for a future phishing attack.

There is a cert and private key for rc.kt.co.kr, South Korea Telecom's Remote Control Service.

Lots of passwords in mnt/hgfs/Desktop/111/account/account.txt from "LG Uplus" (LGU), a South Korean mobile operator. The favicon-search indicates that KIM first hacked into SECUREKI, a company supplying MFA and password services to LGU and from there pivoted into LGU's internal network.

His google search history deserves a closer look. Especially around chacha20 and arc4. The chrome temp files should get some attention.

He seems to download his Dev Tools from [#16] and stole his IDA Pro license from a now disused TOR address [#17].

The Google Chrome configuration files contain these links. Does KIM use (his?) google creds to access these sites? Is wwh1004 his GitHub account?

Did he use google-pay to pay for the three VPN services?

```
"https://accounts.google.com:443, https://[*.]0x1.gitlab.io":
"https://accounts.google.com:443, https://[*.]aldeid.com":
"https://accounts.google.com:443, https://[*.]asawicki.info":
"https://accounts.google.com:443, https://[*.]devglan.com":
"https://accounts.google.com:443, https://[*.]edureka.co":
"https://accounts.google.com:443, https://[*.]johnwu.cc":
"https://accounts.google.com:443, https://[*.]majorgeeks.com":
"https://accounts.google.com:443, https://[*.]maskray.me":
"https://accounts.google.com:443, https://[*.]namecheap.com":
"https://accounts.google.com:443, https://[*.]qwqdanchun.com":
"https://accounts.google.com:443, https://[*.]rakuya.com.tw":
"https://accounts.google.com:443, https://[*.]redteaming.top":
"https://accounts.google.com:443, https://[*.]reversecoding.net":
"https://accounts.google.com:443, https://[*.]shhoya.github.io":
"https://accounts.google.com:443, https://[*.]sparktoro.com":
"https://accounts.google.com:443, https://[*.]tutorialspoint.com":
"https://accounts.google.com:443, https://[*.]wiseindy.com":
"https://accounts.google.com:443, https://[*.]wwh1004.com":
"https://accounts.google.com:443, https://[*.]wwh1004.github.io":
"https://pay.google.com:443, https://[*.]purevpn.com":
"https://pay.google.com:443, https://[*.]purevpn.com.tw":
"https://pay.google.com:443, https://[*.]zoogvpn.com":
```

KIM uses Google-translate to translate error messages to Chinese. A number of Taiwan government and military websites appear in his Chrome history.

The certificate of South Korean's citizens require a deeper look and why he has segregated university professors specifically.

The work/home/user/.cache/vmware/drag_and_drop/ folder contains files that KIM was moving between his Windows and Linux machines. These files include cobalt strike loaders and reverse shells written in powershell. A compiled version of Onnara code as well as Onnara modules for proxying into the government network and more.

In the directory work/home/user/.config/google-chrome/Default/ are many interesting files (.com.google.Chrome*) which give us some insights on interests, search habits, and accessed websites by "KIM". From these we can learn that he is often concerned with cobalt strike (CS) survival, wondering why Kunming is in the Center of Central Inspection Team, and is a big fan of a variety of GitHub projects. He also frequents freebuf.com, xaker.ru, and uses Google translator to read accessibility-moda-gov-tw.translate.goog (translating from taiwanese).

The file voS9AyMZ.tar.gz and Black.x64.tar.gz need a closer look. The binary hashes are not known to virustotal but the names look inviting:

```
2bcef4444191c7a5943126338f8ba36404214202  payload.bin
e6be345a13641b56da2a935eecfa7bdbe725b44e  payload_test.bin
3e8b9d045dba5d4a49f409f83271487b5e7d076f  s.x64.bin
```

His bash_history shows SSH connections to computers on his local network.

Pete Hegseth would say "He is currently clean on OPSEC"

2 The Artifacts

This section analyzes six of Kimsuky's backdoors and artifacts. This work is neither complete nor finished. It is a start to get you excited and learn how Kimsuky operates and what tools they are using.

2.1 Generator vs Defense Counterintelligence Command

Drop Location: vps/var/www/html/

The phishing tool exposes a https website (the phishing-website) under a domain name similar to one that the victim knows and trusts. The victims at dcc.mil.kr are then sent a link to the phishing-website. The attacker then hopes that the victim will enter their login credentials into the phishing-website.

The final redirection of the victim is away from the phishing-website and to an URI on the legitimate website. It is an URI that always throws a login-error. This is a targeted attack and the attacker had to find such an URI on the legitimate website of https://dcc.mil.kr.

The benefit of this "trick" is that the victim will see an error from https://dcc.mil.kr (which he knows and trusts) even though his credentials were submitted to the phishing-website.

config.php:

Contains a long IP black list (and other blacklists) so that companies like Trend Micro and Google are unable to find the phishing site.

generator.php:

This is the remote admin interface to administrate the phishing attack. It is accessible via a configurable password. However, the cookie is hardcoded and the admin-interface can be accessed without a password and by setting the cookie instead:

```
curl -v --cookie "HnoplYTfPX=x" https://phishing-site/generator.php
```

It's trivial to scan the Internet and find phishing results:

```
curl -v --cookie "HnoplYTfPX=x" https://phishing-site/logs.php
```

2.2 Tomcat remote Kernel Backdoor

Drop location: mnt/hgfs/Desktop/tomcat20250414_rootkit_linux234/

This is a kernel level remote backdoor. It allows an attacker to access a host remotely and hide. The drop contains the client (tcat.c), the server side LKM (vmwfxs.mod.c) and userland backdoor (master.c).

The client communicates with the victim's server via (direct) TCP. The LKM sniffs for any TCP connection that matches a specific TCP-SEQ + IP-ID combination (see below). The LKM communicates via `/proc/acpi/pcicard` with its companion master.c userland backdoor.

The master password is `"Miu2jACgXeDsxd"`.

The client uses `"!@nf4@#fndskgadnsewngaldfkl"`.

The script `tomcat20250414_rootkit_linux2345/config.sh` dynamically creates new secret IDs and strings for every installation and saves them to install.h. The master password is hardcoded and does not change.

work/common.c:

Compiled into the client and the master. It contains many old private keys. The newer backdoor generates these keys dynamically (see `install_common.c`).

lkm - vmwfxs.mod.c:

The is the "stub" of the LKM to hook the needed kernel functions.

lkm - main.c:

Process, network-connection, and file hiding takes place here.

lkm - hkcap.c:

Creates /proc/acpi/pcicard to communicate with the userland:

```
echo -n "${DECODEKEY}" > /proc/acpi/pcicard
```

The kernel module intercepts every new TCP connection and checks if the secret TCP-SEQ and IP-ID is used (on any port!). This check is done in `syn_active_check()`. The TCP window size field is used to set the backdoor-protocol (SYN_KNOCK or SYN_KNOCK_SSL mostly).

If this condition is met, it triggers these two steps:

1. Start a userland master.c process (and passes MASTER_TRANS_STRAIGHT_ARGV as parameter to the command line option -m).

2. It redirects the TCP stream to the userland master.c process (and thus stealing it from the intended service).

The master.c then serves the bidding of the attacker.

master - master.c:

The userland companion runs as a hidden process on the victim's server. It handles the SSL handshake and comes with a standard functionality to spawn a root shell or proxy a connection into the internal network.

The main routine is in master_main_handle().

client - tcat.c:

Contains all the functionality to "knock" a victim's LKM (backdoor) via TCP-SEQ+IP-ID and establish an SSL connection to the master.c process started (by the LKM) on the victim's server.

client - kernel.c:

It contains the pre-defined and secret TCP-SEQ numbers and IP-IDs. Any combination can be used to "knock" the remote backdoor. These are not dynamically generated and are identical for every installation.

client - protocol.c:

Contains various stubs and static strings to access the backdoor via SMTP, HTTP, or HTTPS (TLS) protocol.

```
char smtp_e1[] = "250-example.com\r\n250-STARTTLS\r\n250 SMTPUTF8\r\n";
char smtp_tls1[] = "220 Ready to start TLS\r\n";
char smtp_starttls[] = "starttls\r\n";
char smtp_hello[] = "HELO Alice\r\n";
```

It is trivial to detect the LKM locally.

Detecting the LKM remotely might be trivial as well but further testing is needed:

Password authentication is done _after_ the SSL handshake

Thus it should be possible to "knock" the backdoor with a TCP connection (SEQ=920587710 and ID=10213) and port number to a service that normally does not support SSL (like port 80, port 22, or port 25).

1. Establish a TCP connection

2. Send a TLS-CLIENT-HELLO

3. A compromised server will respond with a valid TLS-SERVER-HELLO whereas any other server will not.

2.3 Private Cobalt Strike Beacon

Drop Location: mnt/hgfs/Desktop/111/beacon

This is a custom Cobalt Strike C2 Beacon. This source code was being worked on using Intellij IDEA IDE. beacon/.idea/workspace.xml contains pointers to open files and positions in those files as well as the recent project search history. The last updates in the source code were made in June 2024.

The config.cpp file contains two cobalt-strike config binary blobs. Those are valid blobs that can be parsed with CobaltStrikeParser script from SentinelOne and contains the following settings:

```
BeaconType                  - HTTP
Port                        - 8172
SleepTime                   - 60842
MaxGetSize                  - 1048576
Jitter                      - 0
MaxDNS                      - Not Found
PublicKey_MD5               - c5b6350189a4d960eee8f521b0a3061d
C2Server                    - 192.168.179.112,/dot.gif
UserAgent                   - Mozilla/5.0 (compatible; MSIE 9.0;
                                 Windows NT 6.1; WOW64;
                                 Trident/5.0; BOIE9;ENUSSEM)
HttpPostUri                 - /submit.php
..
Watermark_Hash              - BeudtKgqnlm0Ruvf+VYxuw==
Watermark                   - 126086
```

KIM's version also includes early revision of code that in 2025 was included in the LKM backdoor from above (hkcap.c). However, it is incomplete and missing some key files (like config.h)

The /bak/ subdirectory contains older version of some of the files.

2.4 Android Toybox

KIM is heavily working on ToyBox for Android. It seems to have diverged from ToyBox's official GitHub repository near commit id:

`896fa846b1ec8cd4895f6320b56942f129e54bc9.`

We have not investigated what the many ToyBox modifications are for.

The community is invited to dissect this further.

2.5 Ivanti Control aka RootRot

Drop Location: mnt/hgfs/Desktop/ivanti_control

We present the source code of a client to access a publicly known backdoor.

In 2017, SynAcktiv [#11] mistakenly identified the backdoor as a "vulnerability". It was later found [#12] that this was indeed an implant left behind by the threat actor.

Its name is "RootRot".

This request will reply with "HIT" if the backdoor is running:

```
curl -ksi --cookie "DSPSALPREF=cHJpbnRmKCJISVQiKTsK"
        \ "https://HOST/dana-na/auth/setcookie.cgi"
```

2.6 Bushfire

Drop Location: /mnt/hgfs/Desktop/exp1_admin.py

(The file is also included in ivanti-new-exp-20241220.zip)

This is a Ivanti exploit, possibly for CVE-2025-0282, CVE-2025-0283, or CVE-2025-22457 and the payload installs a backdoor.

Mandiant recently discovered the payload in the wild. They attribute the activity to UNC5221, a suspected China-nexus espionage actor [#13].

The exp1_admin.py uses the same iptable commands that Mandiant discovered in the wild.

The exploit comes with documentation, which, when translated, reads:

>>> **"contact us if the exploit fails"** <<<

It may be an indication that there is code sharing and support happening between these two state actors.

The payload also allows remote access to a compromised system. The interesting part is at line 2219, where the keys/magics are generated:

- The key has 206^4 different combinations only (<31 bit strength).
- The magic has (26*2 + 10)^3 different combinations (<18 bit strength).

The encryption happens at line 85, and is....XOR, using a 31 bit key :>

Line 335, function `detect_door()` can be used to remotely scan for the backdoor.

Notable is that only the magic (but not the key) is used to "knock" the backdoor.

The magic is transmitted in the first 24 bits of the Client-Random in the TLS Client-Hello message. The chances that an ordinary Client-Random has the first 24-bit of this constellation are about 1 in 70.

Meme Alert! There is a "All-your-bases-are-belong-to-us" in the code:

>>>> **"The target doesn't exist backdoor!"** <<<

2.7 Spawn Chimera and The Hankyoreh

Drop Location:
mnt/hgfs/Desktop/New folder/203.234.192.200_client.zip

The client accesses the SpawnChimera backdoor via port knocking. The IP 203.234.192.200 belongs to https://hani.co.kr (The Hankyoreh), a liberal newspaper from South Korea.

The client.py at line 152 shows the port knocking method:

It hides again inside the TLS-Client-Hello, in the 32 byte ClientRandom field, but with a new twist:

The first 4 bytes must be the correct crc32 of the remaining 28 bytes.

```
random = os.urandom(28)
client_hello[15:43] = random
jamcrc = int("0b"+"1"*32, 2) - zlib.crc32(random)
client_hello[11:15] = struct.pack('!I', jamcrc)
```

We invite the community to investigate further.

3 Identifying Kimsuky

The conclusion that the threat actor belongs to Kimsuky was made after a series of artifacts and hints were found, that when analysed revealed a pattern and signature that was too exact of a match to belong to anyone else.

Among these hints is the system's "locale"-setting set to Korean, along with several configuration files for domain names that were previously tied to Kimsuky's infrastructure and attacks. There are similarities between the dumped code and the code from their previous campaigns.

Another recurring detail was the threat actor's strict office hours, always connecting at around 09:00 and disconnecting by 17:00 Pyongyang time.

3.1 - Operation Covert Stalker

[...this section has been shortened for the print release...]

Operation Covert Stalker[#1] is the name given by AhnLab to a months-long spear-phishing campaign conducted by North Korea against individuals (journalists, researchers, politicians...) and organizations in South Korea.

The web server configuration for a domain associated with this attack was found on the threat actor's system.

```
SSLCertificateFile /etc/letsencrypt/live/nid-security.com/cert.pem
```

3.2 - GPKI Stolen Certificates

In early 2024, a new malware written in Go and labelled Troll Stealer was discovered by S2W[#4]. This malware has the ability to steal GPKI (Government Public Key Infrastructure) certificates and keys that are stored on infected devices.

GPKI is a way for employees of the South Korean government to sign documents and to prove their authenticity.

The threat actor had thousands of these files on his workstation.

```
subject=C=KR, O=Government of Korea, OU=Ministry of Unification,
OU=people, CN=Lee Min-kyung
issuer=C=KR, O=Government of Korea, OU=GPKI, CN=CA131100001
```

Drop location: work/home/user/Desktop/desktop/uni_certs && work/home/user/Downloads/cert/

The threat actor developed a Java program to crack the passwords protecting the keys and certificates.

```
136박정욱001_env.key Password $cys13640229
041▒Φ├ó┐╡001_env.key Password !jinhee1650!
041▒Φ├ó┐╡001_sig.key Password ssa9514515!!
[...]
```

Drop location: work/home/user/Downloads/cert/src/cert.java

3.3 Similar Targets

Our threat actor has attacked the same targets that were previously attributed to attacks by Kimsuky.

Naver

Naver Corporation is a South Korean conglomerate offering a wide range of services. A search engine (the most used in the country), Naver Pay (mobile payment service), Naver Maps (similar to Google Maps), email services, and so on.

Naver has a history of being targeted by North Korea. In 2024, Zscaler discovered a new Google Chrome extension called TRANSLATEXT developed by Kimsuky[#8]. This extension can inject arbitrary JS scripts when visiting specific pages. Upon visiting `nid.naver.com` - the Naver login page - the extension injects `auth.js` into the browser to steal the login credentials.

The phishing attack described in section 2.1 uses the domain `nid.navermails.com` as its main URL. This domain has been found to be associated with Kimsuky by Ahnlab[#9].

Ministry of Unification

A regular target of Kimsuky is the South Korean Ministry of Unification. The attacker used the cracked passwords from the GPKI and crafted a custom worldlist for brute forcing.

The log files show that these passwords were tried against the ministry's domain.

```
unikorea123$
unikorea1!!
unikorea100
unikorea625!
[...]
```

Drop location: work/home/user/Downloads/cert/dict/pass.txt

3.4 Hypothesis on AiTM attack against Microsoft users

In the middle of 2022, an AiTM attack was detected and reported by Microsoft[#5] and Zscaler[#6]. The principal of the attack is the use of a web server that acts as a proxy between the legitimate login page and the victim.

The victims were sent an email, inviting them to click on a HTML attachment. When opened, they would be redirected to the proxy via HTTPS. The proxy would then forward any request to the Microsoft server (re-encrypt the data via HTTPS).

Once logged in, the proxy would record the session cookie and redirect the victim to the Microsoft server.

The stolen cookie is valid and can be used by the attacker without any further MFA. The domain used for this campaign was websecuritynotice.com [#7].

While this exact domain was not found on this threat actor's system, a very similar one was used (notice the additional 's'):

```
subject=CN=*.websecuritynotices.com
```

Drop location:
vps/etc/letsencrypt/live/websecuritynotices.com

The Tactics, Techniques, and Procedures (TTPs), the similarity of domain names, and post-exploitation activities (payment fraud, ...) show a strong link to Kimsuky.

3.5 Is KIM Chinese?

KIM uses Google to translates Korean into simplified Chinese. He does seem to understand some (very little) Korean without translating.

KIM follows the Chinese public holiday schedule: May 31st - June 2nd was the Dragon Boat Festival. KIM was not working during this time whereas in North Korea this would have been a normal working day.

However, using https://github.com/obsidianforensics/hindsight, his Chrome settings reveal that KIM is on "Korean Standard Time".

3.6 Fun facts and laughables

In September 2023, "KIM" attempted to purchase the domain name 'nextforum-online.com' at namecheap.com. The payments could be made using Bitcoin, what could go wrong?

A few days later, namecheap.com disabled the domain without given an explanation. When "KIM" asked to have it unblocked, namecheap.com requested the following:

```
In order to verify the legitimacy of the registered
domain(s), please provide us with the following information:
* The purpose of the registration of the domain<br>
* The documentation confirming the authorization to act
  on behalf of Microsoft or a confirmation that the domain(s)
  in question is not associated with it.
```

LOL, afterall, the namecheap.com is not so bulletproof :)

Another fun-fact: In 2020, when websecuritynotice.com was used in a phishing campaign, the owner created several subdomains of realistic URLs for the phishing attacks:

```
login.websecuritynotice.com. IN A 80.240.25.169
wwwoffice.websecuritynotice.com. IN A 80.240.25.169
www-microsoft.websecuritynotice.com. IN A 80.240.25.169
prod-msocdn-25ae5ec6.websecuritynotice.com. IN A 80.240.25.169
prod-msocdn-55e5273a.websecuritynotice.com. IN A 80.240.25.169
prod-msocdn-84311529.websecuritynotice.com. IN A 80.240.25.169
prod-msocdn-c7b8a444.websecuritynotice.com. IN A 80.240.25.169
aadcdn-msauth-84311529.websecuritynotice.com. IN A 80.240.25.169
sts-glb-nokia-346189f1.websecuritynotice.com. IN A 80.240.25.169
res-cdn-office-84311529.websecuritynotice.com. IN A 80.240.25.169
aadcdn-msftauth-25ae5ec6.websecuritynotice.com. IN A 80.240.25.169
aadcdn-msftauth-55e5273a.websecuritynotice.com. IN A 80.240.25.169
aadcdn-msftauth-84311529.websecuritynotice.com. IN A 80.240.25.169
r4-res-office365-55e5273a.websecuritynotice.com. IN A 80.240.25.169
r4-res-office365-84311529.websecuritynotice.com. IN A 80.240.25.169
```

However, in 2025, "KIM" was sloppy and used the main domain only:

```
http://www.websecuritynotices.com/request.php?i=amhraW0xQGtsaWQub3Iua3I=
```

(The "i" parameter is the base64 encoded email of the recipient. In this case 'jhkim1@klid.or.kr'.)

In January 2025, this domain pointed to the IP 104.167.16.97. In March 2025, the domain download.sponetcloud.com resolved to the same IP.

There is its sibling on virustotal: sharing.sponetcloud.com

The following URLs are associated with this domain:

```
https://sharing.sponetcloud.com/logo.png?v=bG1lMjc2MUBzcG8uZ28ua3I=
https://sharing.sponetcloud.com/bigfile/v1/urls/view?\
shareto=aGFudGFlaHdhbkBzcG8uZ28ua3I=
```

The parameters are again base64 encoded, are decode to 'lme2761@spo.go.kr' and 'hantaehwan@spo.go.kr'. Both targets in the South Korean Government Prosecution Office.

The same email addresses (and many more) show up on "KIM's" VPS in the file request_log.txt:

```
hantaehwan@spo.go.kr
paragon74@spo.go.kr
baekdu475@spo.go.kr
[...]
```

Or is this a false-flag threat actor?

"KIM" may have deliberately pointed some of his domains to IP addresses that were previously known to be associated with Kimsuky.

For example, nid-security.com has the following DNS hosting history:

```
nid-security.com. IN A 27.255.80.170 (observation date: 2024-11-05)
nid-security.com. IN A 45.133.194.126 (observation date: <= 2025-05-09)
nid-security.com. IN A 185.56.91.21
nid-security.com. IN A 192.64.119.241
*.nid-security.com. IN A 45.133.194.126
lcs.nid-security.com. IN A 27.255.80.170
lcs.nid-security.com. IN A 45.133.194.126
nid.nid-security.com. IN A 27.255.80.170
nid.nid-security.com. IN A 45.133.194.126
www.nid-security.com. IN A 45.133.194.126
rcaptcha.nid-security.com. IN A 27.255.80.170
rcaptcha.nid-security.com. IN A 45.133.194.126
zwkd3e3wbc.nid-security.com. IN A 45.133.194.126
```

The phishing log on the VPS, dated 2 December 2024, shows this domain:

```
https://nid.nid-security.com/bigfileupload/download?\
h=UJw39mzt3bLZOESuajYK1h-G1UlFavI1vmLUbNvCrX80-\
AtVgL7TIsphr1hlrvKOdOR-dbnMHVV7NJ4N
```

During this month, the domain resolved to 45.133.194.126. Was 27.255.80.170 a red herring?

Last fun-fact. When registering the websecuritynotices.com domain name the "Kimsuky" member had his email address visible in SOA records. lol

```
websecuritynotices.com IN SOA ns4.1domainregistry.com dysoni91.tutamail.com
```

References

[#1] https://image.ahnlab.com/atip/content/atcp/2023/10/20231101_Kimsuky_OP.-Covert-Stalker.pdf
[#2] https://raw.githubusercontent.com/stamparm/maltrail/refs/heads/master/trails/static/malware/apt_kimsuky.txt
[#3] https://www.virustotal.com/gui/ip-address/27.255.80.170/relations
[#4] https://medium.com/s2wblog/kimsuky-disguised-as-a-korean-company-signed-with-a-valid-certificate-to-distribute-troll-stealer-cfa5d54314e2
[#5] https://www.microsoft.com/en-us/security/blog/2022/07/12/from-cookie-theft-to-bec-attackers-use-aitm-phishing-sites-as-entry-point-to-further-financial-fraud/
[#6] https://www.zscaler.com/blogs/security-research/large-scale-aitm-attack-targeting-enterprise-users-microsoft-email-services
[#7] https://raw.githubusercontent.com/BRANDEFENSE/IoC/refs/heads/main/AiTM%20Phishing%20Campaign%20IoC's.txt
[#8] https://www.zscaler.com/blogs/security-research/kimsuky-deploys-translatext-target-south-korean-academia
[#9] https://www.ahnlab.com/ko/contents/content-center/32030
[#10] https://cloud.google.com/blog/topics/threat-intelligence/ivanti-connect-secure-vpn-zero-day?hl=en
[#11] https://www.synacktiv.com/sites/default/files/2024-01/synacktiv-pulseconnectsecure-multiple-vulnerabilities.pdf
[#12] https://cloud.google.com/blog/topics/threat-intelligence/ivanti-post-exploitation-lateral-movement?hl=en
[#13] https://cloud.google.com/blog/topics/threat-intelligence/china-nexus-exploiting-critical-ivanti-vulnerability
[#14] https://home.treasury.gov/news/press-releases/jy1938
[#15] https://cloud.google.com/blog/topics/threat-intelligence/china-nexus-espionage-orb-networks
[#16] https://bafybeih65no5dklpqfe346wyeiak6wzemv5d7z2ya7nssdgwdz4xrmdu6i.ipfs.dweb.link/
[#17] http://fckilfkscwusoopguhi7i6yg3l6tknaz7lrumvlhg5mvtxzxbbxlimid.onion/