Monthly Security Report

2020
September

Cover Story

Reflective DLL Injection
Browser security using Secure Header

IGLOOSECURITY

This report is based on the data collected through the SIEM solution at IGLOO SECURITY's Security Operation Center (SOC). IGLOO SECURITY continuously strives to achieve a 24/7 safe cyber environment throughout the year.

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Cover Story

1. Monthly Security Issue

- Security Issue

2. IGLOO Statistics

- Monthly attack service and trend analysis
- Detailed analysis according to different patterns

3. Tech Note

- Reflective DLL Injection

4. Special Column

- Browser security using Secure Header

5. Focus On IGLOO Security

- This month's IGLOO Security

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Cover Story

CHAPTER 1

Monthly Security Issue

1. Security issue

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

1 Security Issue

European Union implements first “cyber sanctions” against North Korea

- In a press release on July 30, the EU announced that it has decided to implement sanctions against individuals and institutions involved in cyber attacks. This is the first time the EU has implemented cyber sanctions.
- The sanctions included six individuals and three institutions in North Korea, China, and Russia, including the North Korean Chosun Expo joint venture that provided financial, technical, and material support for WarnerCry ransomware attacks and Sony Pictures hacking.
- The EU Council said that their cyber attacks had a "significant" impact on the EU, member states and third countries, and it has decided to implement sanctions, including bans on entry and measures to freeze assets. Funding of sanctions by individuals and institutions within the EU is also prohibited.

Deep fake attack warning using open source artificial intelligence tools

- It is argued that as artificial intelligence algorithms are released as open source, cybercriminals' ability to produce fake images, videos, voices, and texts is strengthening.
- FireEye experts demonstrated “synthetic media” created using artificial intelligence algorithms that anyone can use for free at the “Black Hat USA 2020” event that was recently held online in the United States.
- 'Synthetic medium' refers to a fake medium created by combining various combinations of voice, video, image, and sound. A typical example is the use of deep fake technology to create words and actions that no one has ever said.
- They used open source tools related to natural language processing, computer vision, and speech recognition, and combined various combinations of voice, video, image, and sound.
- FireEye said that “fake media” pretending to be American politicians and reporters were actually spreading on social media, emphasizing the need to prepare defense technologies against such deep fake attacks.

Summer vacation-related phishing email advisory

- The number of phishing e-mails related to summer vacations, such as hotel reservations and automatic replies out of the office, has increased.
- According to AhnLab, there are a number of phishing emails related to summer vacation, such as 'Check the hotel room','Please leave a message (random contact)' and'please contact us (malignant URL)', etc. Appears.
- The attacker induces the user to download a malicious document file by attaching a malicious word or PDF file to the e-mail or by including a malicious URL in the e-mail.
- When executing this document file, information on the user's PC may be stolen and additional malicious code may be downloaded.

Copyright © IGLOO Security, Inc. 2019. All rights reserved.

1 Security Issue

E-mail warning of Emotet malware distribution that exploited COVID-19 fear

- It appears that the Emotet malicious code, which has been running for a long time, is spreading again at the end of August following July.
- AhnLab revealed that it has discovered a phishing email with emotet, a representative banking malware, impersonating a COVID-19 protective equipment company.
- The attacker introduces himself as a company that deals with protective equipment such as thermometers and masks, and if the mail receiver is interested in such protective equipment, the sender induces him to read the attached document.
- The file attached to the e-mail is a word file containing a malicious macro, which induces the use of the macro. When executed, the Emotet malicious code is downloaded.
- Security experts urge you to check the sender of mail and refrain from executing macros contained in suspicious files.

In the first half of 2020, 8.8 million COVID-19-related security threats were discovered

- About 8.8 million COVID-19-related threats were discovered in the first half of 2020, of which 92% were spam mail delivered via email, according to a survey.
- According to a security threat report for the first half of 2020 released by Trend Micro, cybercriminals focused on exploiting what the world's attention was focused on in the first half of this year. Cybercriminals were mainly targeting the loopholes of the security system created by many companies introducing remote work.
- Business Email Compromise (BEC) detection rate increased by 18% from the second half of last year. It appears that this is because some telecommuters are more susceptible to social engineering attacks.
- Ransomware now appears to be an always-on threat regardless of the specific period. Although the number of ransomware-related threats decreased, the number of new ransomware attack groups increased by 36% compared to the first half of the previous year.
- Meanwhile, the number of vulnerabilities discovered in the industrial control system (ICS) increased by 16% compared to the first half of last year.

COVID-19 threatens industrial control system safety... Increasing remote access and increasing exploitation

- Due to the COVID-19 outbreak, major industrial control systems also appear to be facing serious cyber threats.
- According to the '2020 First Half ICS Risk and Vulnerability Report' released by Claroti, the number of attacks that exploited this increased as mobility was restricted due to COVID-19 and remote access to the ICS network increased.
- According to the report, more than 70% of vulnerabilities discovered in ICS in the first half could be exploited remotely, and it was analyzed that they were particularly affected by serious vulnerabilities in energy, critical manufacturing, and water resources.

Copyright © IGLOO Security, Inc. 2019. All rights reserved.

1 Security Issue

US celebrity Twitter hacker caught... Three people indicted, including 17-year-old youth

- The US Department of Justice said on the 31st (local time) that it had arrested and prosecuted three criminals, including 17-year-old Graham Ivan Clark, in connection with the hacking of a celebrity Twitter account.
- They hacked into a celebrity's Twitter account and posted a post saying, 'If I send $1000 (about 1.2 million won) in bitcoin, I will double the money in 30 minutes.' Over 130 Twitter accounts, including accounts of leading politicians and business people, including former US President Obama, Tesla CEO Elon Musk, and Amazon CEO Jeff Bezos, were targeted.
- It turns out that hackers used the internal login government of Twitter employees to gain access to the Twitter system and reset the passwords of each account.

US government warns of North Korean cyber attacks targeting defense industry

- The Cyber and Infrastructure Security Administration (CISA) and the Federal Bureau of Investigation (FBI) under the Department of Homeland Security (DHS) released a Malware Analysis Report (MAR) on North Korean cyber attacks in the defense sector on the 19th. They named the remote-controlled Trojan horse (RAT) malware caught this time as "blinding can," and pointed to the "Hidden Cobra", a hacking organization supported by the North Korean government, as behind the attack.
- According to the report, "Hidden Cobra" hacked a US government subcontractor earlier this year to gather key defense and energy technology information. It was analyzed that it was designed to leak information when the system was infected by spreading a malicious document pretending to be a job posting in a major defense company.

Copyright © IGLOO Security, Inc. 2019. All rights reserved.

Cover Story

CHAPTER 2

IGLOO Statistics

1. Monthly attack service and trend analysis

   01. Top 10 monthly vulnerability attacks

   02. Event by vulnerability compared to previous month

   03. Top 10 monthly attack service(port)

   04. Monthly attack service(port) cases compared to previous month

   05. Top 10 monthly attack service patterns

   06. Monthly attack service incident patterns compared to previous month

2. Detailed analysis according to different patterns

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

1 Monthly attack service and trend analysis

The IGLOO Security SOC collects data on attacks according to the vulnerability and analyzes the attack. Through this process, future cyber threats can be predicted.

01. Top 10 monthly vulnerability attacks

Zyxel remote code execution (CVE-2020-9054) in the top 10 vulnerability attacks collected over the month of September 2020
, Cross Site Script (XSS), and GPON Router Vulnerability attacks newly entered the rankings. In addition, WordPress sample page access, SQL Injection, and phpMyAdmin sample page attack patterns have entered the top rankings with a rate increase compared to the previous month.

※ Based on the Incident response report data of IGLOO Security SOC

Rank	Pattern	No. of cases	Ratio(%)	Fluctuation
1	WordPress sample page access	617	16.45%	▲3
2	SQL Injection	537	14.32%	▲1
3	phpMyAdmin sample page access	518	13.81%	▲2
4	ZeroShell kerbynet RCE (CVE-2009-0545)	512	13.65%	▼2
5	ThinkPHP Remote Code Execution Vulnerability	334	8.91%	▲1
6	URL extension access control	297	7.92%	▲2
7	Command Injection (Netgear Routers Vulnerability)	280	7.47%	▲2
8	zyxel remote code execution (CVE-2020-9054)	235	6.27%	NEW
9	Cross Site Script(XSS)	246	6.56%	NEW
10	GPON Router Vulnerability	174	4.64%	NEW
	Total	3,750	100.00%	-

[Table2-1] TOP 10 monthly vulnerability attacks

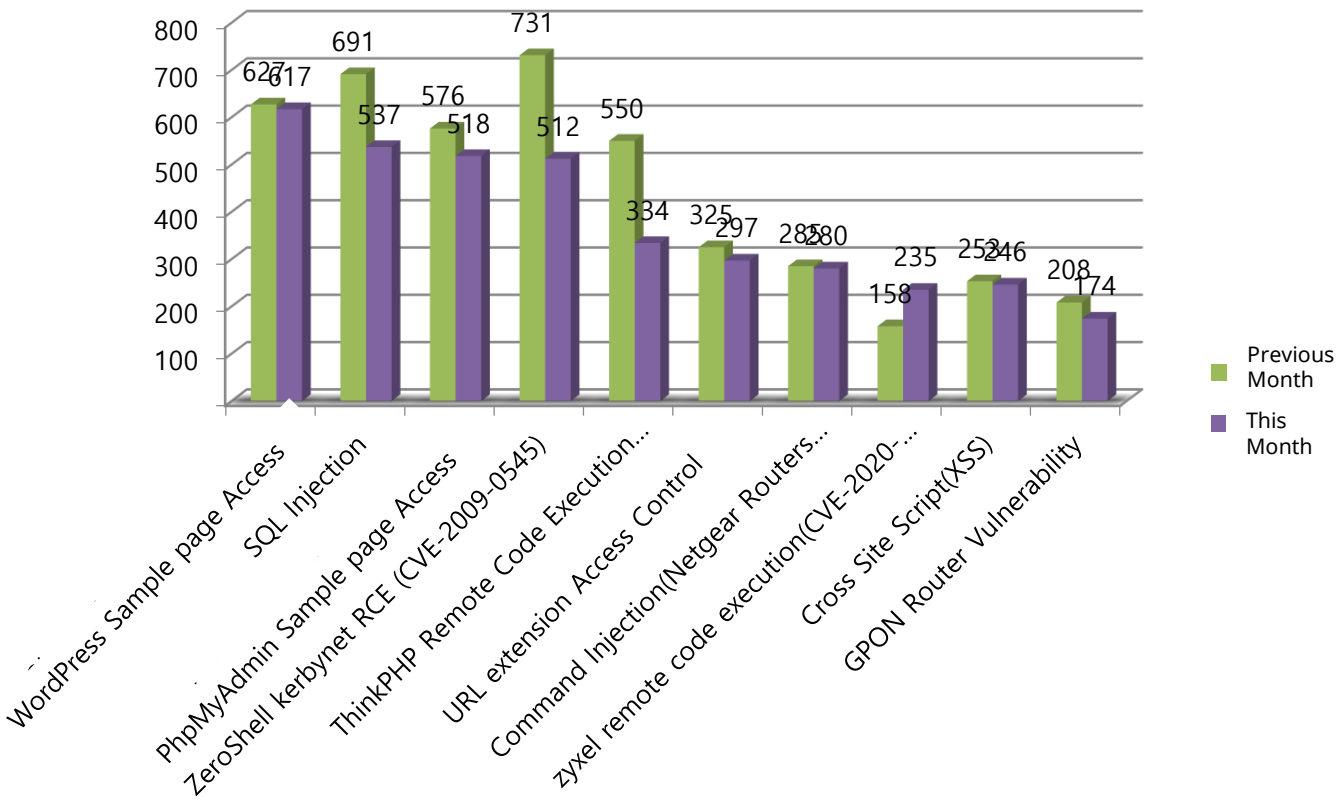
Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

1 Monthly attack service and trend analysis

The IGLOO Security SOC collects data on attacks according to the vulnerability and analyzes the attack. Through this process, future cyber threats can be predicted.

02. Event by vulnerability compared to previous month

As a result of analysis of vulnerability events collected for one month in September 2020, the total number of Top 10 decreased compared to the previous month. However, web vulnerability attacks using web scanners such as WordPress sample page access, SQL Injection, and phpMyAdmin sample page access are continuously occurring, so administrators need to delete unnecessary basic web pages and manage web pages against SQL Injection attacks.



[Figure 2-1] Comparison of previous month by vulnerability

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

1 Monthly attack service and trend analysis

The IGLOO Security SOC collects data on attacks according to the server(port) and analyzes the attack. Through this process, future cyber threats can be predicted.

03. Top 10 monthly attack service(port)

As a result of analysis of events collected for one month in September 2020, the number of events using the Oracle (TCP/1521) port increased by more than 10 million compared to the previous month, raising the ranking by 3 steps.Other Microsoft-DS (TCP/445), MSSQL (TCP/1433), Unsigned (TCP/7178) events using ports decreased slightly compared to the previous month.

※ Based on IGLOO Security SOC SIEM data collection

Rank	Service(Port)	No. of cases	Ratio(%)	Fluctuation
1	HTTPS(TCP/443)	3,874,471,739	69.95%	-
2	DNS(UDP/53)	810,397,299	14.63%	-
3	HTTP(TCP/80)	275,744,463	4.98%	▲1
4	Microsoft-DS(TCP/445)	242,798,101	4.38%	▼1
5	SNMP(UDP/161)	100,530,323	1.81%	-
6	ICMP(0/ICMP)	88,553,400	1.60%	-
7	Oracle(TCP/1521)	38,599,200	0.70%	▲3
8	MSSQL(TCP/1433)	38,001,472	0.69%	▼1
9	Telnet(TCP/23)	35,186,244	0.64%	-
10	Unsigned(TCP/7178)	35,001,499	0.63%	▼2
	Total	5,539,283,740	100.00%	-

[Table 2-2] Monthly attack service (port) TOP 10

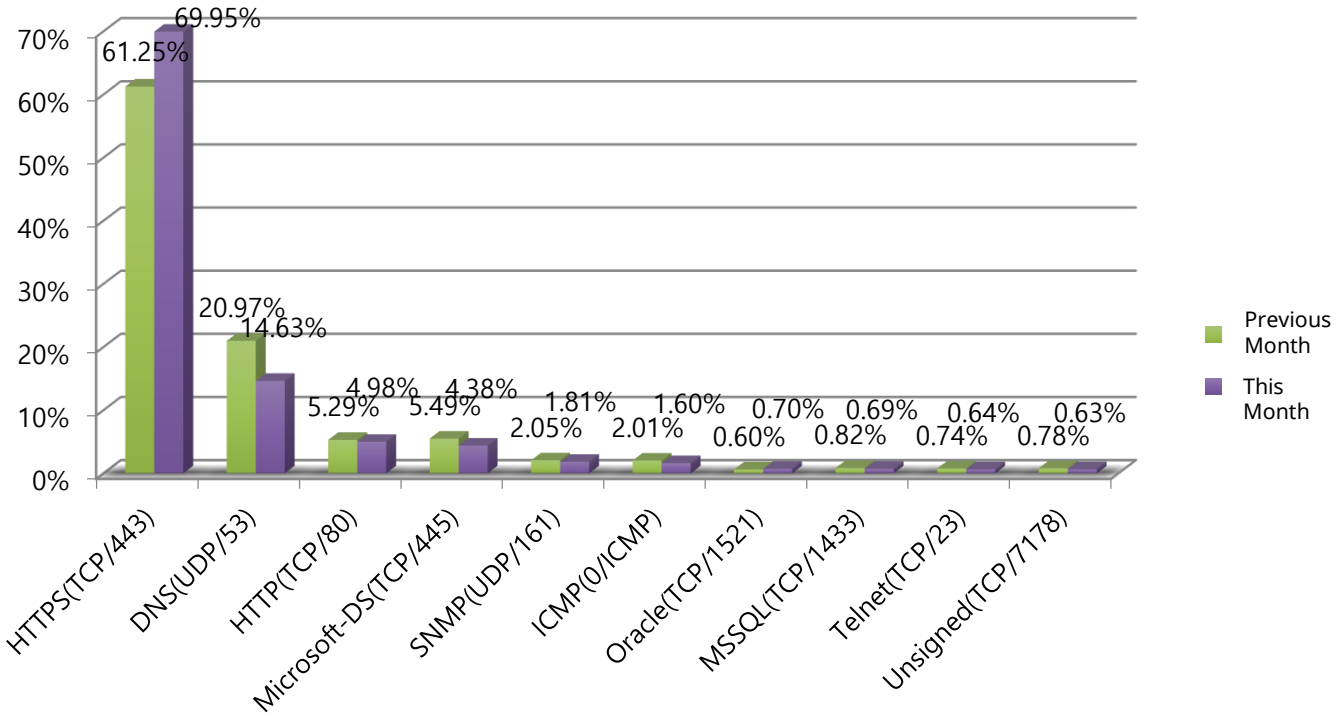
Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

1 Monthly attack service and trend analysis

The IGLOO Security SOC collects data on attacks according to the server(port) and analyzes the attack. Through this process, future cyber threats can be predicted.

04. Monthly attack service(port) cases compared to previous month

As a result of analysis of events collected for one month in September 2020, the number of events continues to increase as the number of newly entered Oracle (TCP/1521) ports in the previous month's TOP 10 increased by more than 10 million this month. When accessing the Oracle DB successfully, an attacker can search, insert, and modify internal system information using SQL query statements, so access control and attention must be exercised to prevent exposure or access to the outside.



[Figure2-2] TOP10 monthly attack port

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

1 Monthly attack service and trend analysis

The trend of attack patterns detected by IGLOO SECURITY SOC can be used to identify the trend of the latest attacks.

05. Top 10 monthly attack service patterns

In the top 10 attack patterns in September 2020, Netbios Scan (Messenger RPC Dcom MyDoom...) (UDP-137)
, TCP Invalid port event ranking has slightly increased, and other MS WINS Server Registration Spoofing Vuln-1[Req](UDP-137), Apache HTTPd DoS(tcp 80) events have newly entered the ranking.

※ Based on IGLOO Security SOC SIEM data collection

Rank	Pattern	No. of cases	Ratio(%)	Fluctuation
1	SMB Service connect(tcp-445)	1,210,488,074	97.38%	-
2	Ack Storm	19,328,194	1.55%	-
3	ACK Port Scan(F/W Scan)	4,445,512	0.36%	-
4	Dcom_TCP_Sweep(MSBlaster Worm Messenger...)	3,688,534	0.30%	-
5	Netbios Scan (Messenger RPC Dcom MyDoom...) (UDP-137)	2,080,810	0.17%	▲1
6	TLS Malformed Handshake DoS	895,278	0.07%	▼1
7	HTTP Login Brute Force	806,462	0.06%	-
8	TCP Invalid port	716,448	0.06%	▲2
9	MS WINS Server Registration Spoofing Vuln-1[Req](UDP-137)	330,848	0.03%	NEW
10	Apache HTTPd DoS(tcp 80)	260,878	0.02%	NEW
	Total	1,243,041,038	100.00%	-

[Table 2-3] TOP10 monthly attack pattern

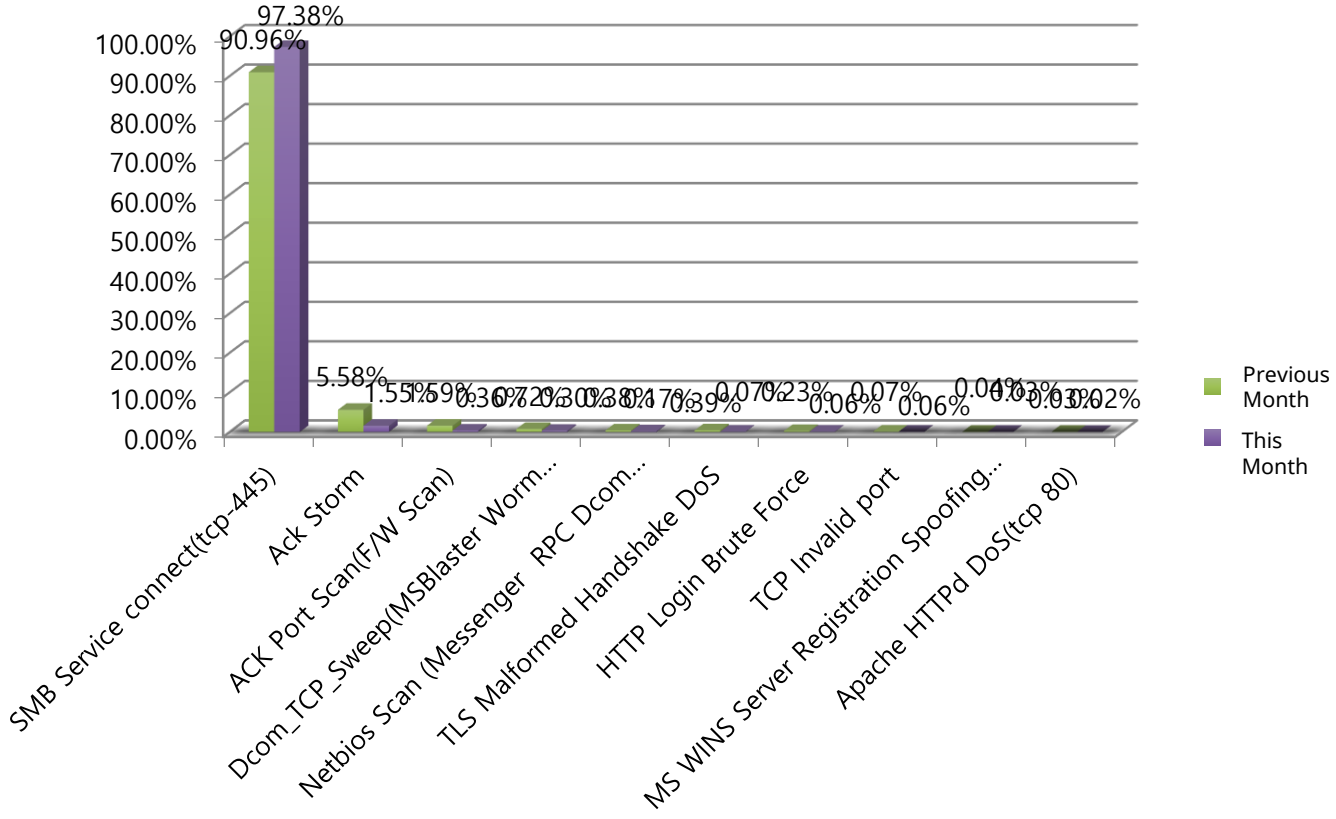
Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

1 Monthly attack service and trend analysis

The attack pattern rankings detected by IGLOO SECURITY SOC can be used to identify the latest trends of attacks.

06. Monthly attack service incident patterns compared to previous month

MS WINS Server Registration Spoofing Vuln-1[Req](UDP-137) and Apache HTTPd DoS(tcp 80) events newly entered the Top 10 attack patterns in September 2020, and 400,000 TCP Invalid port events compared to the previous month. This event can be used as an attack to understand information on the victim server by attempting to communicate using a well-known port in both the source port and destination port, or by sending data to a port that is not normally used. The administrator recommends control and access control for port communications that are not normally used.



[Figure 2-3] Monthly event comparison by Attack(Pattern)

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

2 Detailed analysis according to different patterns

We introduce detailed analysis results according to attack patterns, focusing on TOP 10 among attack patterns that occurred in September.
By referring to the detailed analysis results for each detection pattern, the vulnerabilities of the system must be taken in advance.

Attack Pattern	Detailed Analysis Results
SMB Service connect(tcp-445)	Microsoft Windows uses SMB protocol to share files and printer resources with other computers. Sharing resources using SMB in older versions (95, 98, Me and NT) of Windows used TCP port 137 and 139 and UDP port 138 by executing NETBIOS with NETBIOS over TCP/IP. On older versions of Windows 2000/XP, it was possible to run SMB directly through TCP/IP on TCP port 445. If the files are shared with easily guessable password or passwords are not set, they can suffer from secondary attacks.
Ack Storm	Attacker sends a large amount of TCP / IP Ack signal to the target server, and the target server is an attack method that creates unnecessary load and delays normal service. This attack is also used to Hijacking the packet in which the Session is made.
ACK Port Scan (F/W Scan)	ACK Port Scan (F/W Scan) is a type of scanning attack targeting vulnerable ports which are allowed to be accessed by firewall policy. Attackers can collect information on vulnerable ports by analyzing the returned packet as a response to the packet sent in advance. These attacks are not intended to destroy the system but to collect information for planning a scenario for secondary attack.
Dcom_TCP_Sweep (MSBlaster Worm Messenger...)	W32.Blaster.Worm is a worm that spreads by using the DCOM RPC Buffer Overflow vulnerability. The worm checks whether the TCP / 135 port is enabled and infects the system when it is found. The infected system activates TCP / 4444 port, downloads malicious files from the host server, registers them in the registry, and in this process, traffic from the infected system may increase.
Netbios Scan (Messenger RPC Dcom MyDoom...) (UDP-137)	NetBios checks each other's information through UDP port 137, establishes a session with TCP 139, and exchanges data through TCP 138. An attacker can use this UDP port 137 to establish a session with the target system and scan information about the folders and networks shared on the target system.
TLS Malformed Handshake DoS	TLS Malformed Hadnshake DOS attack is a type of DOS attack using a maliciously manipulated TLS packet. It occurs during the handshake of a maliciously manipulated TLS client. It is an attack that a remote attacker can cause a load on the victim system by sending a maliciously crafted TLS packet to the affected system.
HTTP Login Brute Force	This attack accesses the HTTP WEB service port(80) and repetitively inserts random strings through a Tool-Kit in order to find out the password for a specific ID(root, guest etc). Passwords which are easy to guess or are in an alphabetic/symbolic order are more prone to this attack. This attack can be prevented by using IDs and passwords over 6 letters and by Filtering data from the HTTP port(80/TCP).
TCP Invaild Port	According to the network protocol (RFC 1700), port 0 is a reserved port, so it is not normally used in a public network. Ports less than 1024 are well-known ports. Normally, both the source and destination do not use the well-known port. (An exception is the case that only 20 ftp-data services are provided.) By sending data to a port that is not normally used like this, it may cause an error in the operation of the receiving system or a fingerprint attack that identifies the recipient's OS.
MS WINS Server Registration Spoofing Vuln-1[Req] (UDP-137)	This attack is a vulnerability caused by not properly verifying the NetBIOS communication name while the name is registered in the Windows WINS server. The vulnerability could allow an attacker to redirect Internet traffic to an address of the attacker's choice through spoofing a web proxy.
Apache HTTPd DoS(tcp 80)	Apache is an open source web server running on operating systems such as UNIX, Linux, and MS Windows. This is a vulnerability in ap_get_mime_headers_core(). If the header starts with a blank or tab, Apache allocates as much memory as necessary. A remote attacker sends a maliciously crafted packet and causes the server to allocate a large amount of memory to prevent Apache from running.

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Cover Story

CHAPTER 3

Tech Note

Reflective DLL Injection

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

Security Operation Center
Security Analysis Team

01. Overview

The attack techniques of malicious code attackers are evolving day by day. Various distribution methods are used to attack targets more reliably and accurately, and malicious codes are secretly hidden by using various techniques to avoid detection of security device/software. Among them, methods of hiding malicious codes using Fileless are becoming more and more diverse.

It was found that the Magniber ransomware discovered earlier this year was injected into the Internet Explorer process (Fileless) to encrypt files without leaving malicious files on the user's PC. In addition, in May, a DOC document-type malicious code that impersonated NCNK's “North Korea COVID-19 Situation Interview Document” was discovered, which was a fileless form using the Windows task scheduler.

Like this, malicious codes that use the Fileless technique are appearing without any boredom, and the frequency of use is expected to increase in the future. For attacks, power shell scripts are mainly obfuscated, or techniques such as Process Hollowing and Reflective DLL Injection are used.

On June 22 of this year, malicious code pretending to be a vision protection program was distributed in the Internet community. The malicious code was a file that did not show any suspicious behavior other than executing a paint board (normal file) in the background when executed. However, as a result of detailed analysis, a malicious behavior of reflective DLL injection of arbitrary DLLs to the paint (normal file) was caught. In the background, a log connecting to a random C2 remains, but there is no file that performs malicious behavior. Let's look at how to use Reflective DLL Injection technique in actual malicious code.

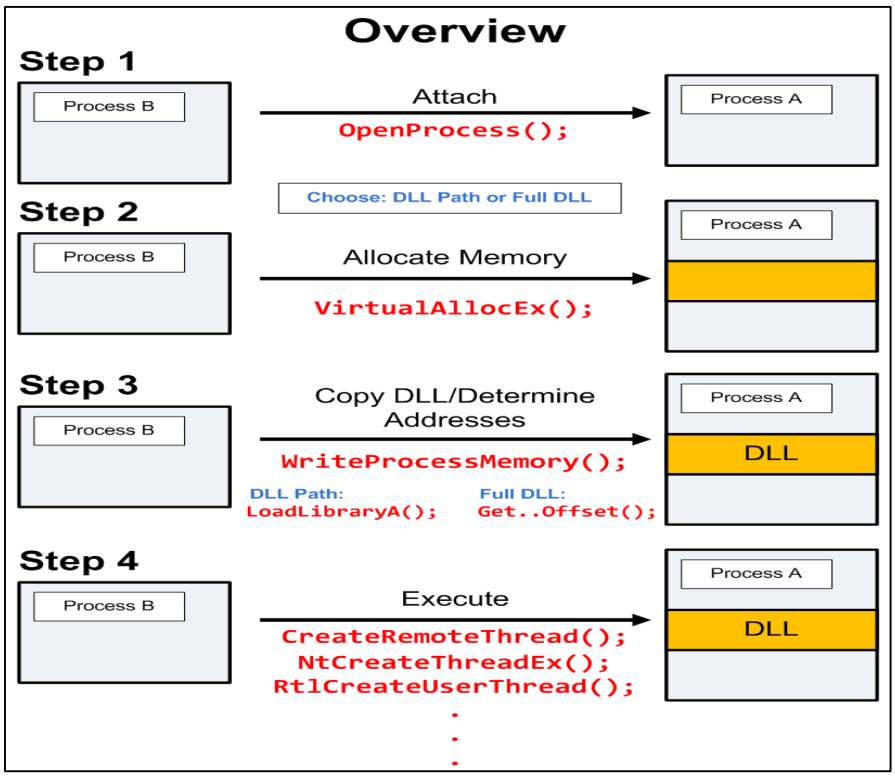
Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

02. DLL Injection and Reflective DLL Injection

1) DLL Injection

DLL injection means a method of forcibly injecting a DLL into the space of another running process.



[Figure 3-1] DLL Injection process (Resource : malwareforensics1.blogspot.com)

1. Using OpenProcess to get the handle of the target process
2. Free up memory space of target process with VirtualAlloc
3. Enter the path of the DLL to be injected into the memory of the target process with WriteProcessMemory.
4. Get LoadLibary API Address (GetModuleHandle, GetProcAddress)
5. Execute injected data of target process with CreateRemoteThread

Because it can inject a DLL in a relatively simple way, it has been widely used in malicious code, but it has a disadvantage that it is easily detected because of the lpStartAddress parameter that is entered when executing the injected DLL of the target process with CreateRemoteThread. In the lpStartAddress argument, the address of the LoadLibrary (or GetProcAddress) API obtained through GetProcAddress is entered. If this part is detected, DLL Injection can be prevented.

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

2) Reflective DLL Injection

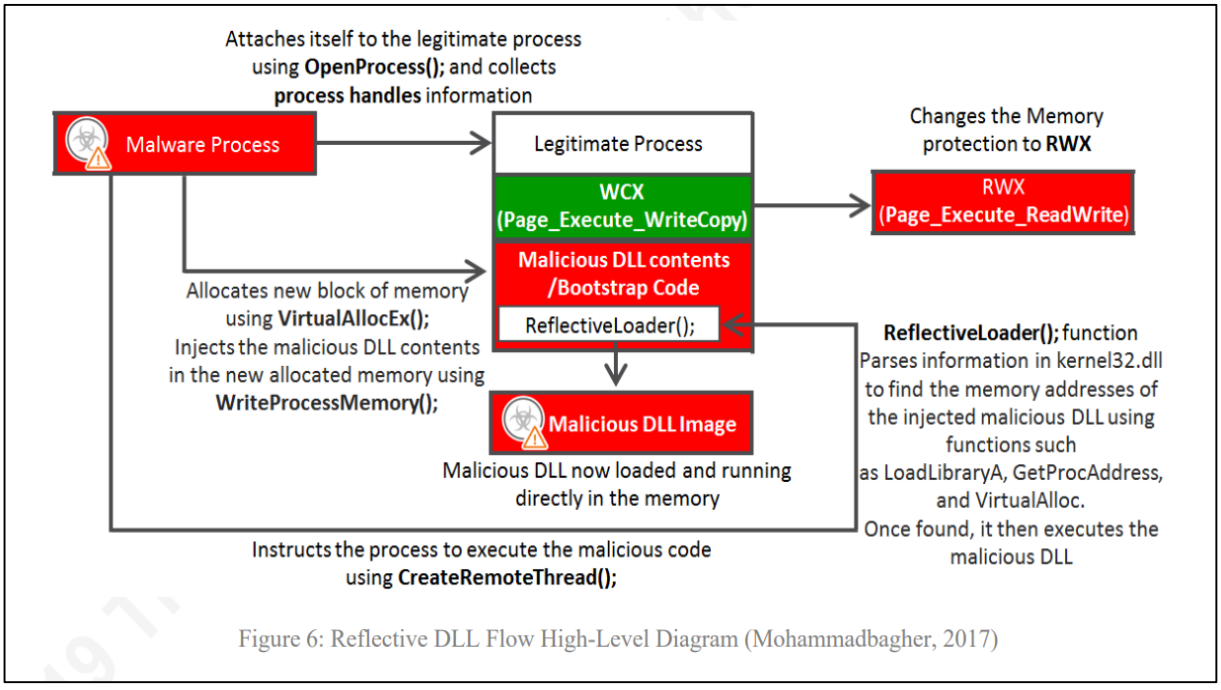
Unlike the existing DLL injection method, **Reflective DLL Injection** operates by inserting data for an arbitrary DLL into the memory of the currently running process and then directly mapping it to execute. DLLs that actually perform malicious actions in this way do not exist in the background and are difficult to detect. In this regard, it is widely used in fileless malicious codes, and was also used in Eternal Blue and Double Pulsar, which were an issue in 2017.

1. Using OpenProcess to get the handle of the target process
2. Free up memory space of target process with VirtualAlloc
3. Enter data of DLL to be injected into the memory of target process with WriteProcessMemory

[Reflective Loader] → Granting memory execution permission (RWX (write, read, execute))

- **4. Mapping DLL to memory**
- **5. New Import Table**
- **6. Ready for relocation and mapped DLL operation**

7. Execute injected data of target process with CreateRemoteThread



[Figure 3-2] Reflective DLL Injection Process (Source: giac.org)

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

3) The difference

The two injection methods differ depending on how the DLL is injected into the process. The difference in DLL Injection is that the executable file (PE) is injected into the process as it is, and in the case of Reflective DLL Injection, the executable file is directly injected into the memory currently being used during injection, and even the tasks that make it usable from the memory are carried out together.

Classification		DLL Injection	Reflective DLL Injection
Injection	Method	Directly insert the path of the DLL to be injected	Directly insert data to be injected into memory and map it to memory (MAPPING)
	Execution	Thread	Thread
How to detect		• Check the registry (AppInIt_DLLs) value • Checking the argument value inside the thread function	• Checking the authority of the memory area that is out of normal operation (as it must be loaded into the memory, RWX authority is granted to an arbitrary memory area)

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

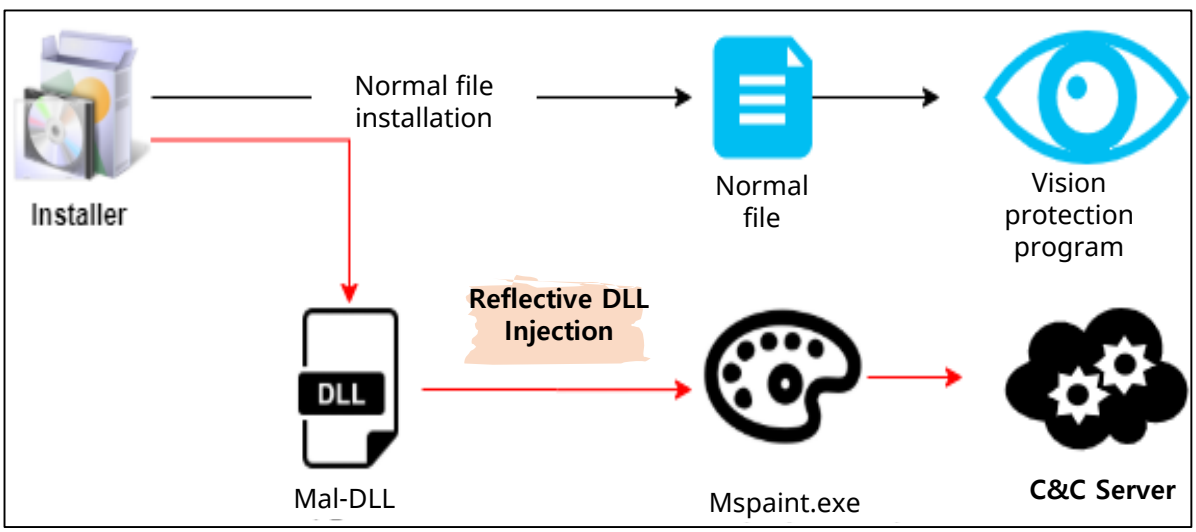
03. The principle of reflective DLL injection as a malicious code

The malicious code used in the analysis is the malicious code that was distributed in the Internet community data room in June by impersonating the vision protection program. It is in the form of an installer and normally downloads the vision protection program when executed, but the paint (mspaint.exe) process runs in the background and transmits PC information to the C2 address. In the case of Paint, the normal process inside System32 was used, and the attackers used the Reflective DLL Injection technique for the attack.

1) Overall configuration of malicious code-Where was it used?

The overall schematic diagram of the malicious code is shown in [Figure 4-3]. At the same time as execution, the malicious DLL (wctE64E.tmp) appears to be dropped and immediately deleted. However, a strange phenomenon occurs in which the paint (mspaint.exe) is running in the background and communicates with the C&C server. The DLL file that is presumed to be performing malicious behavior has disappeared, but the malicious behavior continues through the normal process. How can this be? This is all done in a malicious DLL, and we will look at the details in the next chapter.

Classification	Contents
File name	wctE64E.tmp
MD5	19493F139D74950681354BB881113683
File size	286.50 KB (293,376 byte)



[Figure 3-3] Schematic diagram of malicious code

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

2) Reflective DLL Injection – How it works?

Unlike the existing DLL injection method, Reflective DLL Injection operates by inserting data for an arbitrary DLL into the memory of the currently running process and then directly mapping it to execute. DLLs that actually perform malicious actions in this way do not exist in the background and are difficult to detect. In this regard, it is widely used in fileless malicious codes, and was also used in Eternal Blue and Double Pulsar, which were an issue in 2017.

Pre-work to put data into memory

The malicious code used mspaint.exe (Paint) as a target for injecting malicious DLLs.
Create the process

```
!decAPI_7FEF5795330(&unk_7FEF57D75E0, 24)// kernel32.CreateProcessW
(CreateProcessW = decAPI_7FEF5795330(&unk_7FEF57D75E0, 24),
 !CreateProcessW(0i64, v2, 0i64, 0i64, 0, 0x8000000, 0i64, 0i64, &Dst, &v8, -2i64)) )// v2 = mspaint.exe
```

DLLLoader64_CDEA.exe
mspaint.exe

[Figure 3-4] mspaint.exe process open

It tries to create the same file as the malicious DLL that is running in the same location as the currently running malicious DLL file (wctE64E.tmp), but since the dwShareMode parameter of CreateFileW is set to 0x0, it is set so that the file cannot be created normally in the debugging state. (Anti-debugging effect) By bypassing this part, you can check the part that normally creates the file.

```
CreateFileW = decAPI_7FEF5795330(&unk_7FEF57D75E0, 7);
LODWORD(v42) = 128;
LODWORD(v41) = 3;
v17 = CreateFileW(v4, 0x80000000i64, 0i64, 0i64, v41, v42, 0i64);
```

```
mov     [rsp+160h+var_130], rsi ; hTemplateFile = 0x0
mov     [rsp+160h+var_138], 80h ; dwFlagsAndAttributes = 0x80(FILE_ATTRIBUTE_NORMAL)
mov     [rsp+160h+var_140], 3 ; dwCreationDisposition = 0x3
xor     r9d, r9d        ; lpSecurityAttributes = 0x0
xor     r8d, r8d        ; dwShareMode = 0x0
mov     edx, 80000000h  ; dwDesiredAccess = 0x80000000(GENERIC_WRITE)
mov     rcx, r12        ; lpFileName = 0x1DF240
call    rax             ; kernel32.CreateFileW
```

[Figure 3-5] Collection of handles of malicious DLL files currently being executed

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

The meaning of 'dwShareMode = 0'

If the value of dwShareMode is 0, the file or device cannot be opened when another process requests delete, read, or write access. Therefore, in case of injecting a malicious DLL file to an arbitrary loader for analysis, the file handle created by the corresponding factor cannot be retrieved normally, and further actions are not performed due to this.

After checking the size of the normally created file (the malicious DLL currently being executed), allocate the heap as much as the size. After allocating the heap normally, the data of the created file (the currently executing malicious DLL) is copied to the allocated heap.

```
GetFileSize = here_7FEF5795330(&unk_180475E0, 14);
v19 = GetFileSize(v17, 0i64);                   // kernel32.GetFileSize
                                                // return eax=47A00
if ( v19 - 1 <= 0xFFFFFFFD )
{
  v20 = GetProcessHeap();                       // eax = 250000
  v21 = HeapAlloc(v20, 0, v19);
  if ( v21 )
  {
    if ( here_7FEF5795330(&unk_180475E0, 8) )
    {
      ReadFile = here_7FEF5795330(&unk_180475E0, 8);
      if ( ReadFile(v17, v21, v19, &v51, 0i64) )
```

[Figure 3-6] Process for using malicious DLL data (1)

주소	Hex	ASCII
00000000002667D0	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°
00000000002667E0	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°
00000000002667F0	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°
0000000000266800	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°
0000000000266810	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°
0000000000266820	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°
0000000000266830	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°
0000000000266840	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°
0000000000266850	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°
0000000000266860	0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA	.ð.°.ð.°.ð.°.ð.°

주소	Hex	ASCII
00000000002667D0	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00	MZ..........ÿÿ..
00000000002667E0	B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00	........@.......
00000000002667F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
0000000000266800	00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00	....
0000000000266810	0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68	..°..´.Í!¸.LÍ!Th
0000000000266820	69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	is program canno
0000000000266830	74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20	t be run in DOS
0000000000266840	6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00	mode....$.......
0000000000266850	28 D7 AD 70 6C B6 C3 23 6C B6 C3 23 6C B6 C3 23	(x.pl¶Ã#l¶Ã#l¶Ã#
0000000000266860	2A E7 1C 23 60 B6 C3 23 2A E7 22 23 29 B6 C3 23	*ç.#`¶Ã#*ç"#)¶Ã#

[Figure 3-7] Before memory load (top) and after load (bottom)

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

Open the process (CreateProcess) to inject malicious DLL into the mspaint.exe process.

```
OpenProcess = decAPI_7FEF5795330(&unk_7FEF57D75E0, 50);
v5 = OpenProcess(1082i64, 0i64, v44);// v44 = mspaint.exe의 PID
```

[Figure 3-8] mspaint.exe process open

In order to do Reflective DLL Injection, iAppleCloud function, which is a loader function, is called. The function is declared in the Import Table. Let's take a look at the code inside the function.

Name	Address	Ordinal
DllRegisterServer	0000000180005D50	1
UpgradeToPremium	0000000180013CF0	2
iAppleCloud	0000000180012AE0	3
DllEntryPoint	000000018001E780	[main entry]

```
LABEL_14:
    if ( strstr((v1 + v10), "iAppleCloud") )
      break;
    ++v6;
    ++v9;
    if ( !v8 )
      return 0i64;
  }
  v18 = sub_1800075F0(v5[7], v1);                 // v5[7] : 0x2667D0(iAppleCloud)
                                                  // v1 : 42b88
  return sub_1800075F0(*(v1 + v18 + 4i64 * *v9), v1);
```

[Figure 3-9] Reflective DLL Injection loader function (iAppleCloud) call

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

STEP1 Processing data for use in the loader

Loader function is important for injection. The loader needs information on the system DLLs (ntdll.dll, kernel32.dll) that the currently running malicious DLL is calling. This is because API is required for the function of the loader. In the loader function, the Windows API Hash is used to find the necessary system DLL and API and process it into a form that can be used in the loader.

```
while ( v15 );
if ( v16 == 0x6A4ABC5B )                    // 0x6A4ABC5B : KERNEL32DLL_HASH
  break;
if ( v16 == 0x3CFA685D )                    // 0x3CFA685D : NTDLLDLL_HASH
{
  v28 = v7[4];                              // PLDR_DATA_TABLE_ENTRY->DllBase
  v29 = (v28 + *(*(v28 + 60) + v28 + 136));// PIMAGE_DOS_HEADER->e_lfanew
  v30 = v29[6];                             // PIMAGE_NT_HEADERS->OptionalHeader.DataDirectory
  v31 = (v28 + v29[8]);                     // PIMAGE_EXPORT_DIRECTORY->AddressOfNames
  v32 = (v28 + v29[9]);                     // PIMAGE_EXPORT_DIRECTORY->AddressOfNameOrdinals
```

```
while ( *v34 );
if ( v33 == 0x534C0AB8 || v33 == 0xFF7F061A )// 0x534C0AB8 : NTFLUSHINSTRUCTIONCACHE_HASH
                                             // 0xFF7F061A : RtlExitUserThread
{
  v36 = *v32;
  v37 = v28 + v29[7];
  if ( v33 == 0x534C0AB8 )                  // 0x534C0AB8 : NTFLUSHINSTRUCTIONCACHE_HASH
  {
    v76_NTFLUSHINSTRUCTIONCACHE = (v28 + *(v37 + 4 * v36));
  }
  else if ( v33 == 0xFF7F061A )             // 0xFF7F061A : RtlExitUserThread
  {
    v81_RtlExitUserThread = (v28 + *(v37 + 4 * v36));
  }
```

STEP2 Image load

After getting the header of the PE to be loaded, it allocates memory to load the malicious DLL into memory. When loaded into virtual memory, relocation is performed so that it can be loaded into an executable Imagebase.

```
dwHeaderValue = dwLibraryAddress + *(dwLibraryAddress + 60);// PIMAGE_DOS_HEADER->e_lfanew
dwValueA = dwLibraryAddress + *(dwLibraryAddress + 60);
BaseAddress = v9_VIRTUALALLOC(0i64, *(dwHeaderValue + 80), 0x3000i64, 64i64);// 로드할 DLL을 위한 메모리할당
dwValueB = dwLibraryAddress;
dwValueC = BaseAddress;
```

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

STEP3 Determine which section to load the image

Navigate through the section area to load data into any section.

```
v46 = *(dwHeaderValue + 6);                     // dwHeaderValue)->FileHeader.NumberOfSections
if ( *(dwHeaderValue + 6) )
{
  j_dwValueA = (*(dwHeaderValue + 20) + dwHeaderValue + 44);
  do
  {
    v40 = *(j_dwValueA - 1);
    v47 = dwValueC + *(j_dwValueA - 2);
    dwValueB = dwLibraryAddress + *j_dwValueA;
    --v46;
    if ( *(j_dwValueA - 1) )
    {
      do
      {
        v48 = *dwValueB;
        ++v47;
        ++dwValueB;
        *(v47 - 1) = v48;
        --v40;
      }
      while ( v40 );
    }
    j_dwValueA += 10;
  }
  while ( v46 );
}
```

STEP4 Import Table rewrite

Rewrite the IAT by referring to the file of the new image loaded.

```
dwValueC_1 = (dwValueC + *(dwHeaderValue + 144));// // STEP 4: process our images import table...
v50 = dwValueC_1[3];                            // uiValueC->Name
if ( v50 )
{
  do
  {
    dwLibraryAddress_1 = LOADLIBRARYA(dwValueC + v50, dwValueB, v40, j_dwValueA);
    dwValueA_1 = (dwValueC + dwValueC_1[4]);    // PIMAGE_IMPORT_DESCRIPTOR->FirstThunk
    dwValueD = (dwValueC + *dwValueC_1);        // PIMAGE_IMPORT_DESCRIPTOR->OriginalFirstThunk
    dwLibraryAddress_2 = dwLibraryAddress_1;
    while ( *dwValueA_1 )
    {
      if ( dwValueD && *dwValueD & 0x8000000000000000ui64 )
      {
        v55 = *(*(dwLibraryAddress_2 + 60) + dwLibraryAddress_2 + 136);
        dwValueB = *dwValueD - *(v55 + dwLibraryAddress_2 + 16);
        v56 = dwLibraryAddress_2 + *(dwLibraryAddress_2 + *(v55 + dwLibraryAddress_2 + 28) + 4 * dwValueB);
      }
      else
      {
        v56 = GETPROCADDRESS(dwLibraryAddress_2, dwValueC + *dwValueA_1 + 2);
      }
      *dwValueA_1 = v56;
      ++dwValueA_1;
      if ( dwValueD )
        ++dwValueD;
    }
    v50 = dwValueC_1[8];
    dwValueC_1 += 5;
  }
  while ( v50 );
  dwHeaderValue = dwValueA;
}
```

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

STEP5 Process image relocation

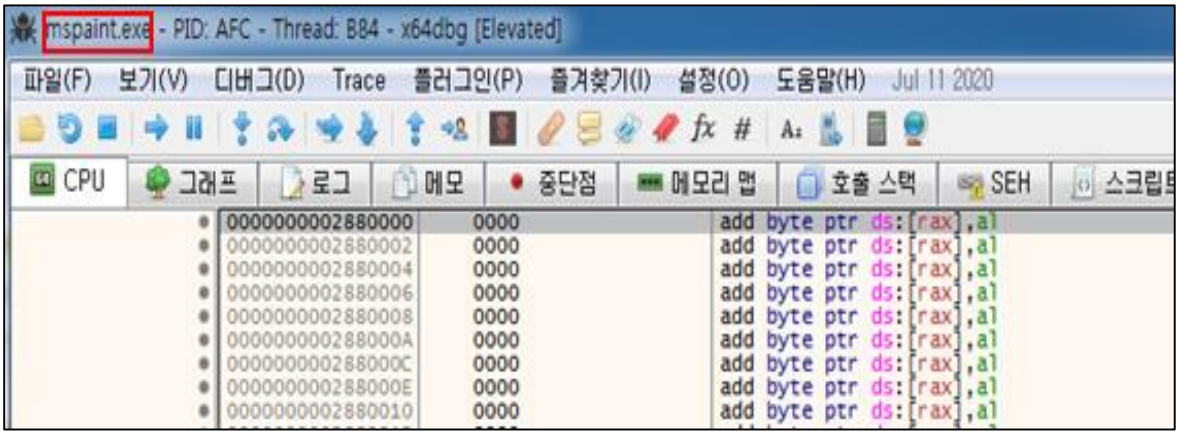
Relocation is done so that the image after all processes can be loaded into memory without collision.

```
uiLibraryAddress = v43v_BaseAddress - *(v38 + 48);// // STEP 5: process all of our images relocations...
if ( *(v38 + 180) )
{
  v58_uiValueC = (v43v_BaseAddress + *(v38 + 176));
  for ( i = v58_uiValueC[1]; i; i = v58_uiValueC[1] )
  {
    v60_uiValueD = v58_uiValueC + 2;
    v61_uiValueA = v43v_BaseAddress + *v58_uiValueC;
    for ( j = ((i - 8) >> 1); j; ++v60_uiValueD )
    {
      j = (j - 1);
      v62 = *v60_uiValueD >> 12;
      switch ( v62 )
      {
        case 10:
          *((*v60_uiValueD & 0xFFF) + v61_uiValueA) += uiLibraryAddress;// IMAGE_REL_BASED_DIR64
          break;
        case 3:
          *((*v60_uiValueD & 0xFFF) + v61_uiValueA) += uiLibraryAddress;// IMAGE_REL_BASED_HIGHLOW
          break;
        case 1:
          *((*v60_uiValueD & 0xFFF) + v61_uiValueA) += WORD1(uiLibraryAddress);// IMAGE_REL_BASED_HIGH
          break;
        case 2:
          *((*v60_uiValueD & 0xFFF) + v61_uiValueA) += uiLibraryAddress;// IMAGE_REL_BASED_LOW
          break;
      }
    }
```

A random space is allocated as much as 0x47AAA (the file size of the malicious DLL being executed) inside the mspaint.exe opened with OpenProcess. (The address of the allocated space varies with each execution)

```
dwSize = v17 + v6 + 64;
v18 = VirtualAllocEx(v8, 0i64, dwSize, 0x3000u, 0x40u);// dwSize=0x47AAA
```

[Figure 3-10] Random space allocation for loading malicious DLL inside mspaint.exe



[Figure 3-11] mspaint.exe-Random space allocated

Copyright © IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

By using WriteProcessMemory, data is inserted into mspaint.exe three times.

NO	Data being copied
1	Malicious DLL currently running loaded in the heap area
2	Absolute path of the currently running malicious DLL
3	The first part of the thread execution

```
WriteProcessMemory_1 = decAPI_7FEF5795330(&unk_7FEF57D75E0, 33);
if ( WriteProcessMemory_1(v8, v18, v7, v30, 0i64) )
{
  v20 = &v18[v16];
  v21 = &v18[v30];
  v31 = &v18[v30];
  if ( a5 && v17 )
  {
    if ( !decAPI_7FEF5795330(&unk_7FEF57D75E0, 33) )
      return v9;
    WriteProcessMemory_2 = decAPI_7FEF5795330(&unk_7FEF57D75E0, 33);
    if ( !WriteProcessMemory_2(v8, v31, a5, v17, 0i64) )
      return v9;
    v20 = &v18[v16];
    v21 = v31;
  }
    if ( decAPI_7FEF5795330(&unk_7FEF57D75E0, 33) )
    {
      WriteProcessMemory_3 = decAPI_7FEF5795330(&unk_7FEF57D75E0, 33);
      if ( WriteProcessMemory_3(v8, v23, &v42, v24, 0i64) )
```

[Figure 3-12] Inserting data using WriteProcessMemory

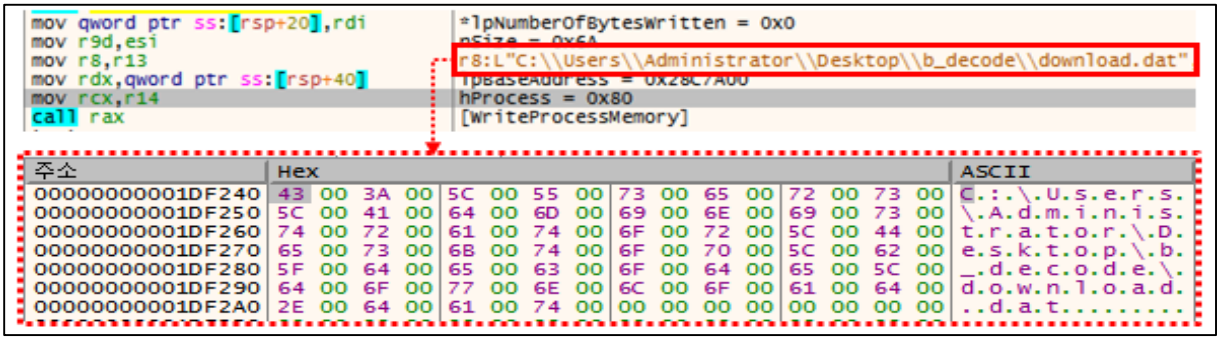
```
mov qword ptr ss:[rsp+20],rdi        *lpNumberOfBytesWritten = 0x0
mov r9d,dword ptr ss:[rsp+40]        nSize = 0x47A00
mov r8,rbx                           lpBuffer = 0x2667D0
mov rdx,r15                          lpBaseAddress = 0x2880000
mov rcx,r14                          hProcess = 0x80
call rax                             [WriteProcessMemory]
```

주소	Hex	ASCII
00000000002667D0	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00	MZ..........ÿÿ..
00000000002667E0	B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00	........@.......
00000000002667F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
0000000000266800	00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00	....
0000000000266810	0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68	..º..´.Í!¸.LÍ!Th
0000000000266820	69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	is program canno
0000000000266830	74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20	t be run in DOS
0000000000266840	6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00	mode....$.......
0000000000266850	28 D7 AD 70 6C B6 C3 23 6C B6 C3 23 6C B6 C3 23	(×.pl¶Ã#l¶Ã#l¶Ã#
0000000000266860	2A E7 1C 23 60 B6 C3 23 2A E7 22 23 29 B6 C3 23	*ç.#`¶Ã#*ç"#)¶Ã#
0000000000266870	2A E7 23 23 FE B6 C3 23 B1 49 13 23 6D B6 C3 23	*ç##þ¶Ã#±I.#m¶Ã#

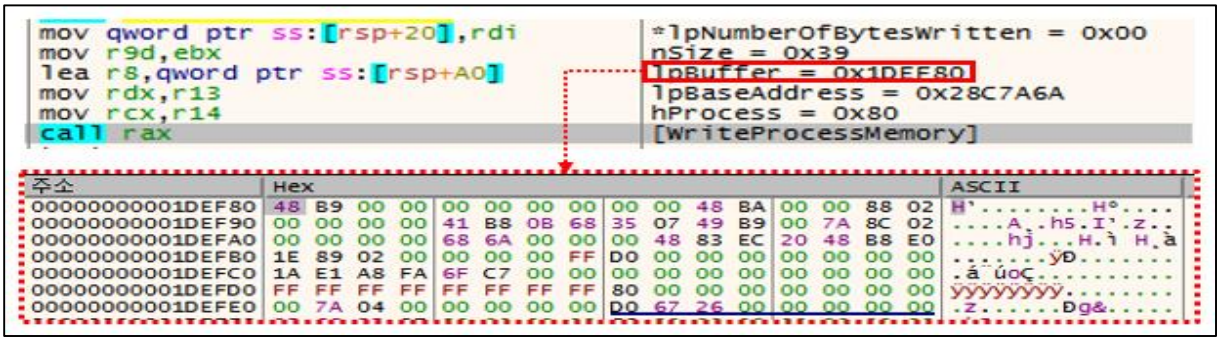
[Figure 3-13] Content inserted through the first WriteProcessMemory

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

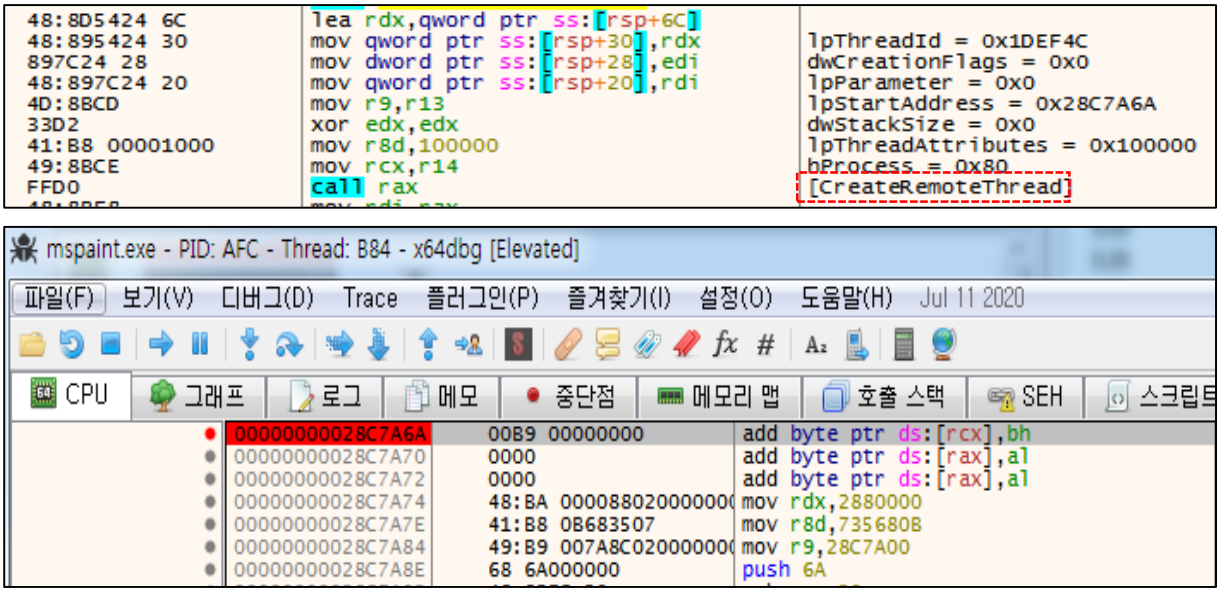


[Figure 3-14] Content inserted through the second WriteProcessMemory



[Figure 3-15] Content inserted through the third WriteProcessMemory

The malicious DLL injected reflective DLL into mspaint.exe is executed as CreateRemoteThread.



[Figure 3-16] Injected DLL data is executed by using CreateRemoteThread

Copyright © IGLOO Security, Inc. 2020. All rights reserved

Reflective DLL Injection

04. Reference

Reflective DLL Injection

- https://0x00sec.org/t/reflective-dll-injection/3080
- https://www.exploit-db.com/docs/english/13007-reflective-dll-injection.pdf
- https://github.com/stephenfewer/ReflectiveDLLInjection
- http://read.pudn.com/downloads443/sourcecode/windows/system/1870992/ReflectiveDllInjection/source/ReflectiveLoader.c__.htm
- https://gitlab.com/kalilinux/packages/metasploit-framework/blob/a77c7a6bdfc7b90b5002a8f84a1ca80c6f37fd6e/external/source/exploits/CVE-2016-0040/Metasploit/Metasploit.h
- http://blog.opensecurityresearch.com/2013/01/windows-dll-injection-basics.html
- https://www.giac.org/paper/gcfa/11563/hunting-ghosts-fileless-attacks/150888
- https://www.microsoft.com/security/blog/2017/11/13/detecting-reflective-dll-loading-with-windows-defender-atp/
- https://zerosum0x0.blogspot.com/2017/07/threadcontinue-reflective-injection.html

Fileless cases

- http://www.dailypop.kr/news/articleView.html?idxno=45150
- https://koya-culture.com/news/article.html?no=123015

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Cover Story

CHAPTER 4

Special Column

Browser security using Secure Header

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Browser security using Secure Header

Security Operation Center
Security Analysis Team

1. Overview

Web Browser, invented in 1990, refers to an environment that provides various HTTP user agents such as desktop and mobile based on a graphical user interface that navigates in a web server and communicates in both directions and outputs HTML documents or files.

According to StatCounter's global browser market share in March 2020, 65.99% used Chrome to become the flagship browser, followed by Safari and Firefox.), Samsung Internet, Edge, and Opera. From these statistics, it can be seen that a large number of users around the world use Chrome browsers in their desktop environment, and Safari, which is mainly Mac users, has a high level of market share.

Looking at the other point of view, website usage by platform, the browser usage rate on mobile worldwide is 51.9%, exceeding the 45.4% usage rate on desktop. The browser share on mobile is also 61.5%, with Chrome the highest, followed by Safari and Samsung Internet.

As such, as the correlation between attack threats is formed by the share of browser types and access environments, browser-based cyber attacks are also increasing. In addition, when building a web site these days, it is possible to build a web service quickly and conveniently because it consists of web pages and scripts created by different developers and is made by combining source codes of third parties, but it is inevitably vulnerable to security management.
For these reasons, many browser manufacturers provide countermeasures to strengthen the security of web vulnerabilities through HTTP security headers.

HTTP header contains additional information such as request/response time and browser used when requesting/responding using HTTP protocol, and can be divided into categories such as authentication, caching, control, cookie, connect, and secure. Some of the headers are summarized in [Table 5-1]. In this issue, let's look at the details and application method of HSTS (HTTP Strict Transport Security) of encryption, X-XSS-Protection of input value verification, and CSP.

Classification	Secure Header	Description
Encryption	**HSTS (HTTP Strict Transport Security)**	Forced use of https protocol when accessing the website
Session management	Same-origin-policy	Loaded only when the imported document or script is from the same source
	Cross-origin-resource-share	Granting permission to request limited resources on web pages from other domains
Information leakage	Referer	Previously visited web page
	X-content-type-options	Limit interpretation to formats other than those sent by the web server
Input value verification	**X-XSS-Protection**	Detect XSS attacks and stop page loading
	CSP(Contents-Security-Policy)	Detect and limit specific attacks, including XSS attacks and data injection attacks
	X-frame-options	Restrict the page from being included in other sites through <frame>, <iframe>, and <object>

[Table 4-1] Part of Secure Header

Copyright © IGLOO Security, Inc. 2020. All rights reserved

Browser security using Secure Header

02. HSTS(HTTP Strict Transport Security) detailed analysis

HTTPS is an encrypted transmission method that compensates for the shortcomings of HTTP protocol that can expose important information to other users by transmitting data in plain text with HTTPS (Hypertext Transfer Protocol Secure), a protocol that supports encrypted communication.
In order to easily distinguish whether encrypted communication is used or not, when you access a financial site, you can see a lock shape. In the case of financial sites, user's important information such as personal information or personal credit information is transmitted, so encrypted communication is used.

So, how does encrypted communication work during data transmission?
When talking about encrypted communication, you may have heard of encrypted communication via SSL (Secure Socket Layer) or TLS (Transport Layer Security).
SSL is used to provide security for the connection between the server and the client, and is characterized by providing the following three services.

- Confidentiality-Prevents other users from intercepting or recognizing information during data transmission
- Integrity-Prevents unintentional modification and corruption during data transmission
- Authentication-protection against man-in-the-middle attacks

TLS was released as a successor version that provided the same service and compensated for vulnerabilities in SSL. Currently, there are many bugs in SSL, so it is recommended to use the latest TLS version.

Classification	Description
HTTP	A protocol that enables data exchange on the web and fetching resources such as HTML
SSL	Encrypt communication between client and web browser
TLS	The next generation version of SSL3.0
HTTS	A protocol that combines HTTP with SSL/TLS certificates to convert data into private form when transmitting

[Table 4-2] Encrypted communication

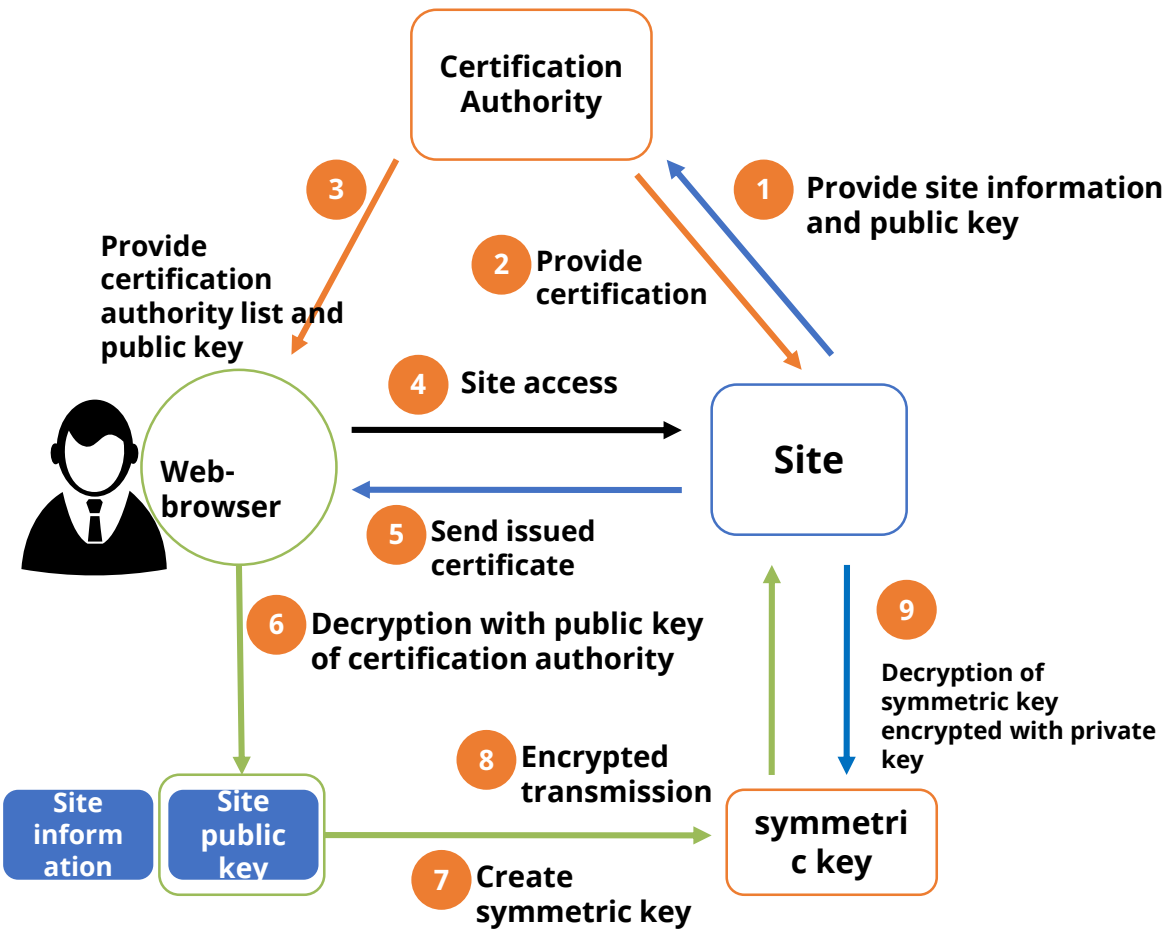
Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Browser security using Secure Header

For encrypted communication, an SSL/TLS certificate must be issued from a certification authority and applied to the web server. The encryption communication method is expressed as [Figure 4-1]. The site provides site information and public key to the certification authority, receives and applies the encrypted certificate from the certification authority, and the web browser receives the certification authority list and each public key from the certification authority.

Accordingly, the web browser has a list of certification authorities and each public key, and the site has a private key that is private.

When a user accesses a site through a web browser, the site transmits the certificate to the web browser, and the web browser decrypts the certificate with the public key of the certificate authority it has to verify that it is a trusted site, and then finds out the site information and public key. The symmetric key is encrypted with the obtained site public key and sent back to the site, and the site decrypts it with a private key, allowing encrypted data to be sent and received.



[Figure 4-1] Encryption communication method

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

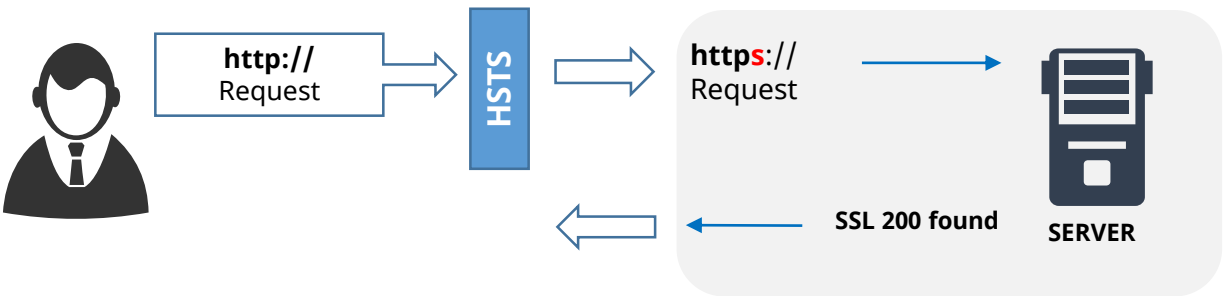
Browser security using Secure Header

However, even if encryption communication is applied in this way, malicious users do not easily give up attempting to steal information. As an example, even though encrypted communication is used, if a request is made by forcibly applying https:// in the address bar to http://, the request is made to http, and data is transmitted in plain text, and information can be stolen.

In general, when HTTPS is forcibly applied, the server can switch to HTTPS, but in this case, the HTTP request from the client is accepted as it is and can be used as an attack vector.

[Figure 4-2] General https enforcement method

The security header created to prevent this is HSTS (HTTP Strict Transport Security), which sets the client to use HTTPS forcibly. HSTS is a method of converting the request sent to the server to HTTPS even if the client makes a HTTP request. Because this is done on the client before going to the server, the HTTP connection itself is not attempted.



[Figure 4-3] HSTS operation method

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Browser security using Secure Header

HSTS can be applied differently according to the content of the indicator as summarized in [Table 4-3], and can be applied to server, web server, PHP, etc. as shown in [Table 4-4]. Also, when I access the site called hstspreload.org, I can check whether the site I want to access uses HSTS.

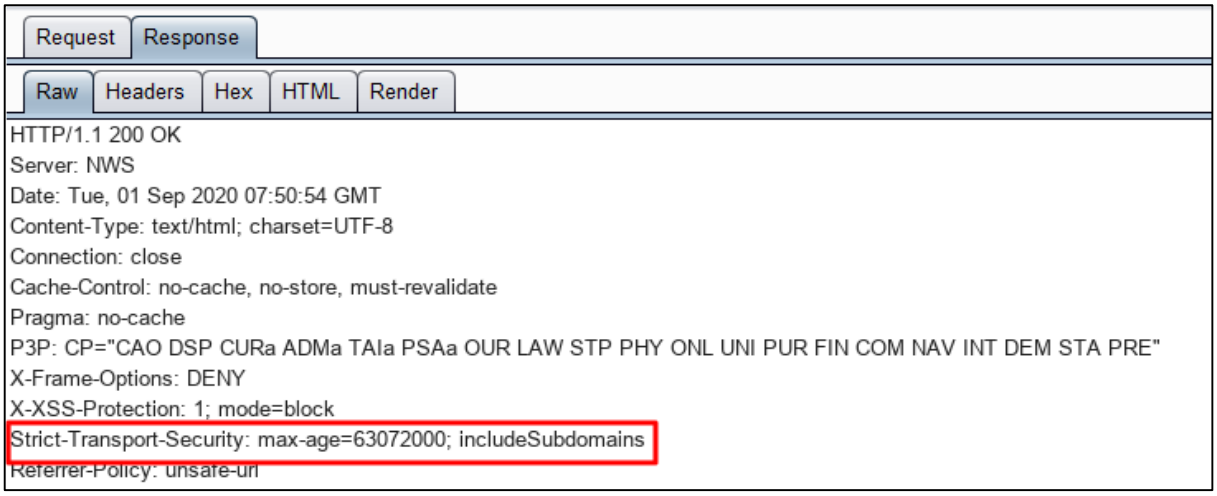
Indicator	Description
Max-age=<expire-time>	Time (in seconds) that HSTT will be set in the browser
includeSubDomains	Extend to subdomain of domain to which HSTS is applied
preload	The browser applies HSTS to the site and adds it to the preload list.

[Table 4-3] HSTS indicator

Target	How to apply
PHP	Header(“Strict-Transport-Security: max-age=86400”);
Webserver (ex, Apache)	Header always set Strict-Transport-Security “max-age=86400; includeSubdomains;”
Server(ex, Nginx)	Add_header Strict-Transport-Security “max-age=86400; includeSubdomains;

[Table 4-4] HSTS applying sample

Looking at the actual HSTS-applied site in [Figure 4-4] below, the site is redirected to HTTPS when a forced connection is attempted with HTTP, as the characteristics of HSTS, and HSTS is used for 63072000 seconds in the browser, and extended to subdomains. You can check what you are using.



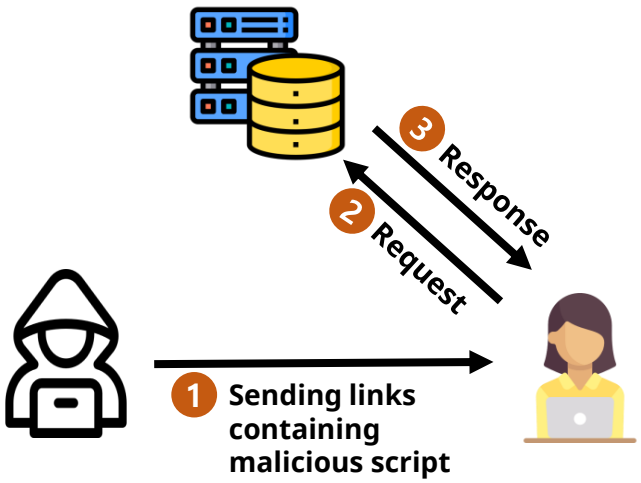
[Figure 4-4] HSTS application site

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Browser security using Secure Header

03. X-XSS-Protection detailed analysis

Cross-Site Scripting (XSS) is one of the most common attacks when starting web hacking. As a basic vulnerability among web vulnerabilities, Reflected XSS attack among the types of XSS attacks is performed by a malicious user sending a URL containing a malicious script to the user, passing it to the web server through the user's access, and inserting the malicious script into the response page of the web server. As a result, user information may be stolen or a website may behave abnormally.



[Figure 4-5] Reflected XSS attack method

In general, in order to prevent an XSS attack, it is prevented by filtering special characters such as <, >, “, etc., which are widely known attack phrases or script languages. This time, we will cover X-XSS-Protection and CSP (Content-Security-Policy), which can enable the Reflected XSS filter in the browser.

X-XSS-Protection supports this header in Explorer, Safari, and Chrome browsers. As shown in [Table 4-5] below, when there are 4 directives and a tag matching the script tag of the URL exists through HTML, a Reflected XSS attack is detected or the page load is stopped.

Indicator	Description
0;	Disable XSS filter
1;	Enable XSS filter, malicious script deletion
1; mode=block	Enable XSS filter and limit browser page load when attack is detected
1; report=<URI>	Enable XSS filter and report when attack is detected

[Table 4-5] X-XSS-Protection indicator

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

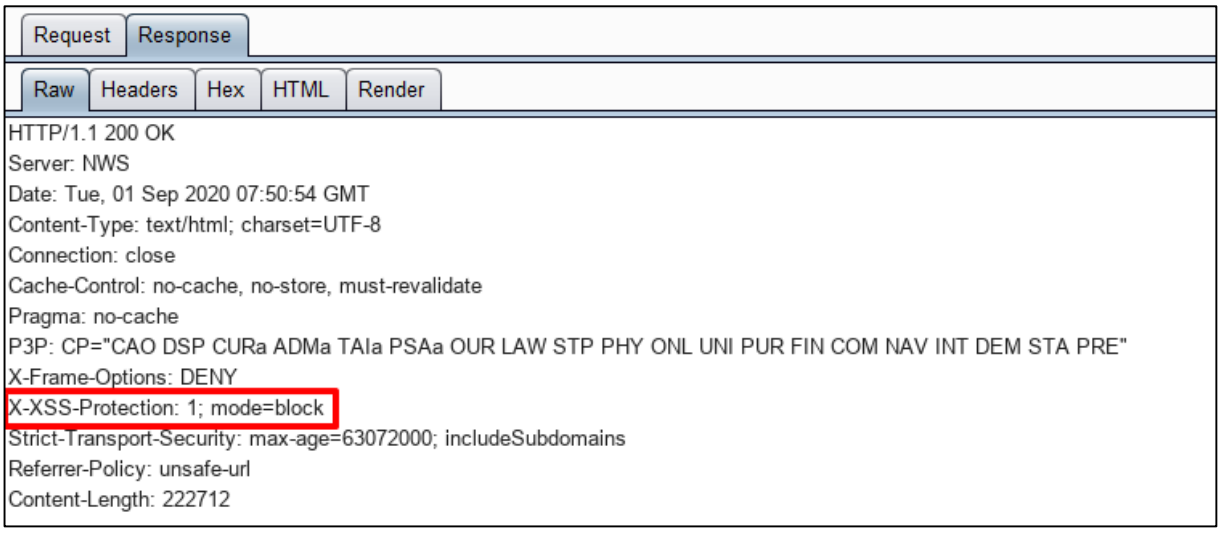
Browser security using Secure Header

The user can directly set it in a browser that supports the function, or it can be applied in PHP, web server, server, etc. when building a web page. [Table 4-6] below is a representative example of each.

Target	How to apply
PHP	Header("X-XSS-Protection: 1;");
Webserver (ex, Apache)	<IfModule mod_headers.c> Header set X-XSS-Protection "1; mode=block" </IfModule>
Server(ex, Nginx)	Add_header "X-XSS-Protection" "1; mode=block";

[Table 4-6] Example of X-XSS-Protection application

Looking at the actual applied site in [Figure 4-6], it can be seen that in the case of the site, the XSS filter is activated and the page load is restricted when an attack is detected.



[Figure 4-6] Application example of X-XSS-Protection

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Browser security using Secure Header

04. CSP(Contents-Security-Policy) detailed analysis

Content-security-policy (CSP) has various web security policies including the function of X-XSS-Protection, the XSS filter discussed above. It is mainly used to prevent attack techniques that insert malicious scripts into web pages such as XSS, data injection, and click jacking, and is a function that protects that only allowed sources can be executed or loaded by designating a trusted content whitelist.

In [Table 4-7] and [Table 4-8], you can see indicators and attribute values for some functions that are frequently used among the many functions of CSP. Blocking range can be specified accordingly.

Indicator	Description
Default-src	Default setting when no other indicators are present
Connect-src	Limit the sources you can connect to
Script-src	Script-related restrictions
Child-src	Web restrictions inserted using iframe and frame tags
Font-src	Limit the sources that can serve web fonts
Img-src	Image loading source restrictions
Reflected-xss	Same function as X-XSS-Protection header
Report-uri	URL to deliver the content when a policy violation case occurs

[Table 4-7] Part of CSP indicator

Attribute value	Description
[Domain]	Domain allowed access
None	Block all
Self	Allow only current domain
Unsafe-inline	Allow inline JavaScript CS in source code
Nonce-[Encrypted character]	Allow inline JavaScript and CSS

[Table 4-8] CSP indicator attribute value

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

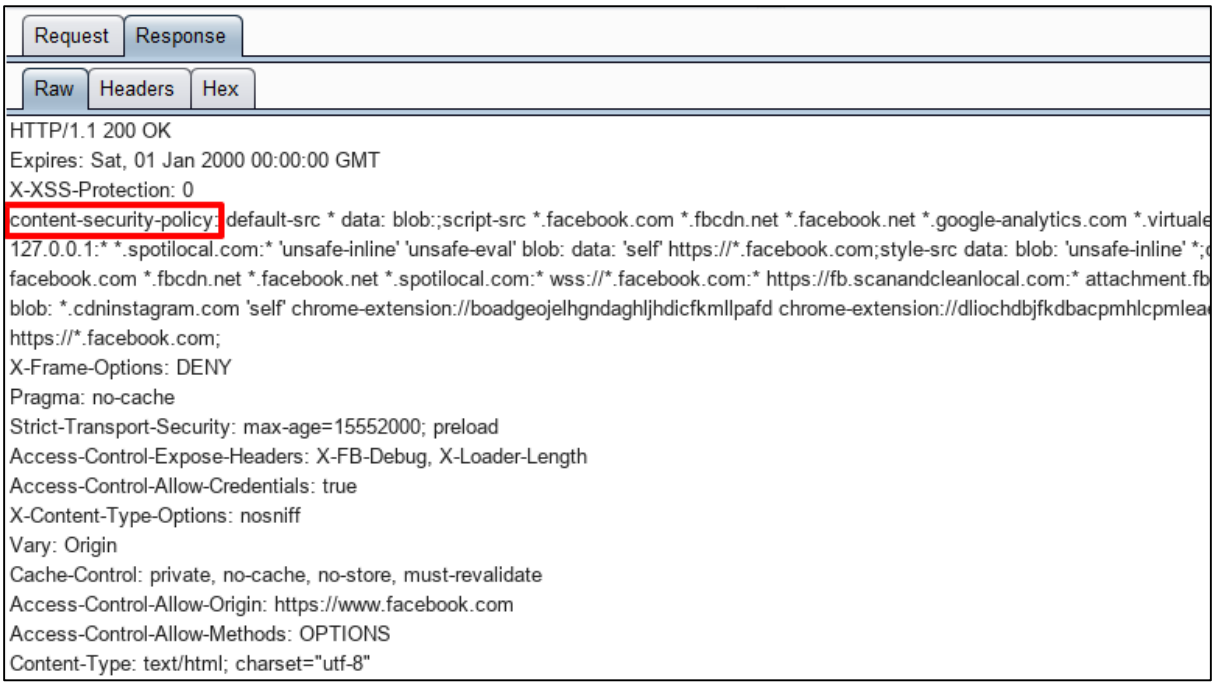
Browser security using Secure Header

The difference between the above-introduced function and CSP is not only having an indicator, but the range of limitations depending on the attribute value of the indicator. CSP can be applied to server, web server, PHP, etc. The following shows typical application methods. Like HSTS, you can check whether CSP is applied on the csp-evaluator.withgoogle.com site.

Target	How to apply
PHP	Header("Content-Security-Policy: default-src 'self';");
Webserver (ex, Apache)	Header set Content-Security-Policy "default-src 'self';"
Server(ex, Nginx)	Add_header Content-Security-Policy "default-src 'self';"

[Table 4-9] CSP application example

Looking at the site actually applied in [Figure 4-7], the source of all content comes from its own domain, but the resource inserted in the same place as the image tag is allowed, but the script tag where the script can be inserted is input only in the allowed domain. You can see that it is set to be possible.



[Figure 4-7] CSP application example

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Browser security using Secure Header

05. Conclusion

The web, which has long been used as a playground for attackers due to the characteristics of a web browser with good accessibility, is constantly generating new threats as the number of cases of building web sites by combining open sources, commercial sources, and libraries, etc. Although there are many ways to prevent and respond to existing threats before these new threats are discovered, there are cases in which companies operating web sites do not incorporate them due to cost or service characteristics.

However, the features introduced above are protected by security headers in supported browsers. It cannot be said that perfect security will be achieved if companies operating web sites use it, but I think it will relieve some of the worries.

In addition, as the use of non-face-to-face services such as distance education, work, and e-commerce is spreading due to the COVID-19, companies building and using web sites do not know which browsers users use a lot and what browser security measures exist You will have to watch with interest.
In addition, browser manufacturers devise various methods to prevent similar attacks such as X-XSS-Protection and CSP, and constantly seek new security measures as Firefox manufacturer Mozila constantly guides them about security measures. It must be done.

06. Reference

[1] Web browser market share status
https://presscat.co.kr/blog/browser-market-share-mar2020/
[2] HTTP headers
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers
[3] HSTS(HTTP Strict Transport Security) concept and setting
https://rsec.kr/?p=315
[4] Difference between HTTP and HTTPS and how they work
https://devdy.tistory.com/14
[5] X-XSS-Protection
https://developer.mozilla.org/ko/docs/Web/HTTP/Headers/X-XSS-Protection
[6] Content Security Policy (CSP)
https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Cover Story

CHAPTER 5

Focus On IGLOO Security

1. IGLOO SECURITY, INC.
   MS Azure Partnership Agreement
2. IGLOO SECURITY, INC.
   First GS certification in the field of AI security monitoring

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Focus on IGLOO Security

IGLOO SECURITY, INC. MS Azure Partnership Agreement

- IGLOO SECURITY expands cloud security business
- Expanding the provision of cloud-based SIEM solutions and security control services that enhance the security of MS Azure environment

[13th Aug 2020] IGLOO Security, INC. is speeding up the cloud security business. IGLOO Security, INC. (CEO Lee Deuk-Choon, www.igloosec.com/en) announced that it has entered into a partnership agreement with Microsoft Azure to strengthen its cloud security business competitiveness. It is planning to focus on providing security monitoring solutions and services that are compatible with Microsoft Azure, which continues to grow rapidly around the world.

IGLOO SECURITY plans to further increase the competitiveness of the cloud security business through this partnership agreement. First, They focus on developing and supplying a cloud-based SIEM (Security Information and Event Management) solution optimized for MS Azure environment. It plans to introduce a SIEM solution that meets the requirements of hybrid and multi-cloud environments that are different from on-premises environments and can also be provided in the form of SaaS (software as a service). In addition, it will build an integrated cloud monitoring system to monitor the cloud environment and provide failure/performance monitoring services.

In addition, it will expand the provision of cloud security monitoring services for Microsoft Azure customers. It is a strategy to provide security monitoring services optimized for various industries such as games, finance, communication, education, etc. by utilizing SPiDER TM AI Edition, the first AI (artificial intelligence) security monitoring solution in Korea. It focuses on finding and responding to security gaps based on machine learning algorithms optimized for threat detection and response.

IGLOO SECURITY CEO Lee Deuk-Choon said, “With COVID-19, as the digital transformation of domestic companies accelerates, the importance of security monitoring solutions and services optimized for the cloud is becoming more prominent. By adding expertise in the cloud to the AI security monitoring capabilities accumulated over the years, we will put more effort into supporting the digital transformation of enterprises.”

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

Focus on IGLOO Security

IGLOO SECURITY, INC. First GS certification in the field of AI security monitoring

- Acquired GS certification, recognized for AI security monitoring technology and value
- Expanding the provision of AI security solutions optimized for cyber security

SPiDER TM AI Edition

[18th Aug 20202] IGLOO SECURITY acquired GS certification for the first time in the field of AI (artificial intelligence) security monitoring, and was recognized for its AI security monitoring technology and value. IGLOO SECURITY, INC. (CEO Lee Deuk Choon, www.igloosec.com/en) announced that the AI security monitoring solution SPiDER TM AI Edition is the first grade of GS (Good Software) certification of Korea Information and Communication Technology Association (TTA) Has been obtained.

GS certification is a national certification system that tests software functions, performance, ease of use, and reliability. The Korea Information and Communication Technology Association (TTA) and the Korea Industrial Technology Testing Institute (KTL) conduct tests under the same conditions as the actual operating environment to objectively evaluate the technical performance and economic value of the software.

SPiDER TM AI Edition is focused on increasing the efficiency of security monitoring and leveling up security monitoring capabilities based on AI technology optimized for cyber security. Depending on the capabilities and experience of the security monitoring personnel, it solves the problems of True-Negative, False-Positive and False-Negative, and provides time to intensively analyze the filtered key information and suspicious clues.

Since the launch of the AI security monitoring solution in early 2019, IGLOO Security has been devoted to R&D and continuously improving the performance of the solution. IGLOO Security has acquired 19 patents related to AI and security monitoring this year. The plan is to accelerate research and development to increase the ability to respond to advanced cyber attacks.

Lee Deuk-Choon, CEO of IGLOO SECURITY, said, “As the untact digital transformation accelerated with COVID-19, the number of attacks that cyber attackers could aim for became wider. Accordingly, the importance of AI security technology for companies and institutions that accelerate digital transformation is expected to become more prominent. As a leading AI security monitoring company, we will put more effort into providing AI security solutions optimized for cyber security.”

Copyright ⓒ IGLOO Security, Inc. 2020. All rights reserved

We Know SECURITY!

Published by IGLOO SECURITY (www.igloosec.co.kr/en)
6 Floor. 7, Jeongui-ro 8-gil, Songpa-gu, Seoul, Korea
T. +82-2-3404-8678
E-mail. overseas@igloosec.com

Distributed by Cyber-Infinity Corp. (www.ci-corp.jp)
株式会社 シーアイシー
東京都千代田区岩本町3-4-3 リードシー秋葉原ビル5階
T. +81-3-5829-5801
E-mail. business@ci-corp.jp

Edited by IGLOO SECURITY Marketing Team
Translated by IGLOO SECURITY Overseas Business Team
© 2020 IGLOO SECURITY, Inc. All rights reserved.
The copyright of this publication is held by IGLOO SECURITY. It is not permissible to reproduce, copy, or distribute any or all of the contents of this publication in any form or by any means without the prior written consent of IGLOO SECURITY. All information contained in the publication can be changed without prior notice.