

1 Infrastructure Security

1.1 Introduction

This whitepaper summarizes incidents to which IIJ responded, based on general information obtained by IIJ itself related to the stable operation of the Internet, information from observations of incidents, information related to services, and information obtained from companies and organizations that IIJ has cooperative relationships.

This volume (Vol.3) covers the period of time from January 1 through March 31, 2009. A number of incidents occurred during this period; we will be addressing the most representative of those in this whitepaper.

Infections of Conficker and its variants were repeatedly reported during the period in question (Conficker first came to prominence last year). A number of other incidents involved the alteration of Web content, facilitated through password theft.

Vulnerabilities that affected a wide range of systems or users have been identified, including OpenSSL vulnerabilities, problems with transparent proxy servers, etc.

The total number of malware specimens on the Internet is in decline, according to IIJ observations. While the number of DDoS attacks has likewise decreased, attacks still happen on a scale that directly affect server performance. SQL injection attacks on Web servers continue at about the same pace as the past periods.

As seen above, the Internet continues to experience many security-related incidents.

1.2 Incident Summary

Here, we discuss the IIJ handling and response to incidents that occurred between January 1 and March 31, 2009. Figure 1 shows the distribution of incidents handled during this period, while Table 1 provides an explanation of categorizations.

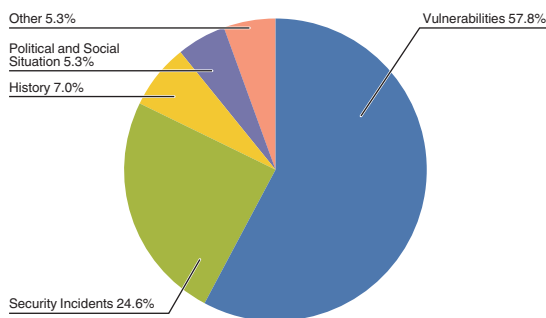


Figure 1: Incident Ratio by Category (January 1 to March 31, 2009)

Table 1: Incident Categories

Category Name	Explanation
Vulnerabilities	Indicate responses to vulnerabilities associated with network equipment, server equipment or software used over the Internet, or used commonly in user environments. Vulnerabilities, information about attacks on vulnerabilities, information from vendors regarding response to vulnerabilities, response steps taken, etc.
Political and Social Situation	Indicates responses to incidents related to domestic and foreign circumstances and international events. Responses to international conferences attended by VIPs, attacks originating in international disputes; measures taken in response to warnings/alarms, detection of incidents, and so forth.
History	Historically significant dates; warning/alarms, detection of incidents, measures taken in response, etc., related to an attack in connection with a past historical fact.
Security Incidents	Unexpected incidents and related response. Wide propagation of network worms and other malware; DDoS attacks against certain websites. Include response to incidents for which the cause was not clearly determined.
Other	Incidents not otherwise categorized. Includes those incidents not directly associated with security problems, including highly concentrated traffic associated with a notable event.

■Vulnerabilities

We noted numerous vulnerabilities related to client operating systems and applications. These included the discoveries or identifications of a significant number of attack methods not relying on specific implementations or versions, including CA spoofing using MD5 vulnerabilities*1, OpenSSL vulnerabilities*2, Intel TXT security structure vulnerabilities*3, host header interpretation problems in transparent proxy servers*4, and Cisco IOS attack methods*5 using Return Oriented Programming*6.

■Political and Social Situation

The 2009 World Baseball Classic and several other international events were held during the period under study. However, IIJ did not note any attacks on IIJ facilities or client networks associated with any of these events. IIJ paid careful attention to political situation during Northern Territories Day (February 7) and Takeshima Day (February 22) [two dates observing controversial events/disputes –Ed.]. IIJ also paid careful attention during the event of the North Korean missile launch in late March. No attacks directly related to these events against IIJ facilities or client networks were detected.

■History

The period in question included several historically significant days on which multiple sites in Japan had been subject to DDoS attacks; however, IIJ did not detect any direct attacks on IIJ facilities or client networks.

■Security Incidents

The largest of unanticipated incidents not linked to political and social situation was the spread of Conficker variants*7 exploiting Microsoft Security Bulletin MS08-067. These variants infect other computers not only via the network using the MS08-067 vulnerability, but also via USB memory devices and other media, resulting in reports of compromised PCs not connected directly to the Internet. Many arguments*8 since have been put forth for disabling the Windows AutoRun feature.

We also noted an increase in alteration of Web content*9 through the theft of user account information (ID/password). See “1.4.2 Alerts Concerning ID/Password Management.”

-
- *1 Intermediate CA certificates can be counterfeited in attacks using this method. However, such attack requires that approximately 8,000 PCs perform one or two days' worth of calculations (<http://www.win.tue.nl/hashclash/rogue-ca/>).
 - *2 OpenSSL has issues with the validation of certain certificates; counterfeit certificates could be viewed as authentic (http://www.openssl.org/news/secadv_20090107.txt).
 - *3 Announcement related to the discovery of a defect allowing the bypass of Intel TXT security protections and related implementation errors (<http://theinvisiblethings.blogspot.com/2009/02/attacking-intel-txt-paper-and-slides.html>).
 - *4 When a transparent proxy exists between the Web browser and Web server, security features such as Same Origin Policy in Java implementation, may be bypassed through the host header manipulation (<http://www.kb.cert.org/vuls/id/435052>).
 - *5 Under this method, the Cisco ROMMON code is utilized to create attack code using Return Oriented Programming, allowing for the creation of version independent exploit code. This method can be a serious threat when a remotely exploitable vulnerability is discovered (http://www.phenoelit-us.org/stuff/FX_Phenoeelit_25c3_Cisco_IOS.pdf).
 - *6 Published materials relating to Return Oriented Programming (http://www.blackhat.com/presentations/bh-usa-08/Shacham/BH_US_08_Shacham_Return_Oriented_Programming.pdf).
 - *7 Information related to Conficker and Downadup variants can be found at, for example, Microsoft Malware Protection Center “Centralized Information About The Conficker Worm” (<http://blogs.technet.com/mmpc/archive/2009/01/22/centralized-information-about-the-conficker-worm.aspx>).
 - *8 Regarding the prevention of malware infections via USB memory by disabling AutoRun: Technical Cyber Security Alert TA09-020A (<http://www.us-cert.gov/cas/techalerts/TA09-020A.html>); Microsoft (<http://support.microsoft.com/kb/967715>).
 - *9 For example, “Web alteration not relying on SQL or RFI” (<http://jvnrss.ise.chuo-u.ac.jp/csn/index.cgi?p=SQL%A4%C8%A4%ABRF%A4%F2%BB%C8%A4%EF%A4%CA%A4%A4Web%B2%FE%E3%E2>). (in Japanese)

“Click-jacking” emerged last year as a malicious technique that involved the inducement of unexpected behavior from user clicking. A technical report^{*10} was released about this technique. Other incidents involved messages disguised as New Year’s or Valentine’s Day greetings, attempting to expose users to malware infections.

■Other

While not directly related to security, a defect^{*11} related to Seagate Technology hard drives received attention due to aspects involving availability. Elsewhere, several BGP implementations were affected operationally due to the distribution^{*12} of routing information having extremely long AS Path attributes. Networks that used these BGP implementations observed interruption or instability of communications; however, there were no incidents that affected networks directly operated by IJ.

1.3 Incident Survey

Of those incidents occurring on the Internet, IJ focuses on those types of incidents that have infrastructure-wide effects, continually conducting research and engaging in countermeasures. In this section, we provide a summary of our survey and analysis results related to the circumstances of DDoS attacks, malware infections of networks, and SQL injections on Web servers.

1.3.1 DDoS Attacks

Today, DDoS attacks on corporate servers are almost a daily occurrence. The methods involved in DDoS attacks vary widely. Generally, however, these attacks are not the type that utilize advanced knowledge of such as vulnerabilities, but rather cause large volumes of unnecessary traffic to overwhelm network bandwidth or server processes for the purpose of hindering services.

*10 JPCERT/CC Technical Notes “About Click-Jacking Defenses” (<http://www.jpCERT.or.jp/ed/2009/ed090001.pdf>). (in Japanese)

*11 See the official Seagate Technology announcement (<http://seagate.custkb.com/seagate/crm/selfservice/search.jsp?DocId=207931>).

*12 See, for example, NANOG Mailing List Archive (<http://www.merit.edu/mail.archives/nanog/msg15468.html>).

Figure 2 shows the circumstances of DDoS attacks handled by the IJ DDoS Defense Service between January 1 and March 31, 2009. This information shows traffic anomalies judged to be attacks based on IJ DDoS Defense Service standards. IJ also responds to attacks perpetrated on clients of other IJ connection services; however, these incidents are excluded from the figure due to the difficulty in accurately ascertaining the facts of each situation.

There are many methods that can be used to carry out a DDoS attack. In addition, the capacity of the environment attacked (bandwidth and server performance) will largely determine the degree of the effect. The statistics in Figure 2 categorize DDoS attacks into three types: attacks on bandwidth capacity^{*13}, attacks on servers^{*14}, and compound attacks (several types of attacks on a single target conducted at the same time).

During the three months under study, IJ dealt with 197 DDoS attacks. This averages to two attacks per day, but represents a decline in average incidents compared to our prior report (September through December, 2008). No bandwidth capacity attacks were noted. Server attacks accounted for 94% of all incidents, and compound attacks accounted for the remaining 6%. The largest server attack was a SYN flood of 90,000pps—an attack of a scale that can seriously compromise a server; however, the maximum traffic volume was around 108Mbps, which was observed in compound attacks. As to the reason behind the infrequency and small scale of bandwidth attacks, we believe that such lies in the fact that an attacker is also affected by heavy traffic loads.

Of all attacks, 74% ended within 30 minutes of commencement, while the remaining 26% lasted anywhere from 30 minutes to up to 24 hours. During the time period under study, IJ did not note any attacks that exceeded 24 hours in length.

In most cases we observed an extremely large number of IP addresses, whether domestic or foreign. We believe this is accounted for by the use of IP spoofing^{*15} and botnet^{*16} usage as a means for carrying out DDoS attacks.

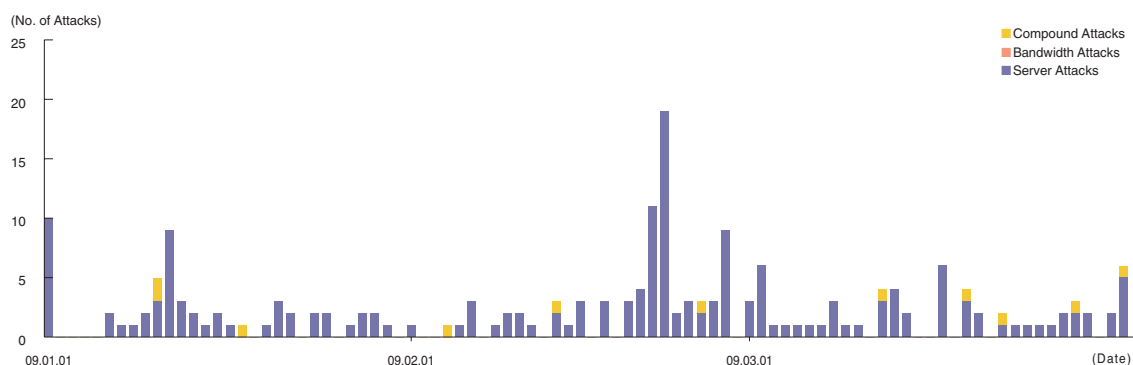


Figure 2: DDoS Attacks

- *13 Attack that overwhelms the network bandwidth capacity of a target by sending massive volumes of larger-than-necessary IP packets and fragments. The use of UDP packets is called a UDP flood, while the use of ICMP packets is called an ICMP flood.
- *14 TCP SYN flood, TCP connection flood, and HTTP GET flood attacks. TCP SYN flood attacks send mass volumes of SYN packets that signal the start of TCP connections, forcing the target to prepare for major incoming connections, causing the wasted consumption of processing capacity and memory. TCP Connection flood attacks establish mass volumes of actual TCP connections. HTTP GET flood attacks establish TCP connections on a Web server, and then send mass volumes of HTTP GET protocol commands, wasting processing capacity and memory.
- *15 Misrepresentation of a sender's IP address. Creates and sends an attack packet that has been given an address other than the actual IP address of the attacker in order to pretend that the attack is coming from a different location, or from a large number of individuals.
- *16 A "bot" is a type of malware that institutes an attack after receiving a command from an infected external C&C server. A network constructed of a large number of bots acting in concert is called a "botnet."

1.3.2 Malware Activities

Here, we will discuss the results of the observations of the Malware Investigation Task Force (MITF)^{*17}, malware activity observation project operated by IIJ. The MITF uses honeypots^{*18}, connected to the Internet in a manner similar to general users in order to observe communications arriving over the Internet. Most appear to be communications by malware selecting a target at random, or an attempt to locate a target for attack.

■ Status of Random Communications

Figure 3 shows trends in the total volumes of communications coming into the honeypot (incoming packets) between January 1 and March 31, 2009. Figure 4 shows the distribution of sender's IP addresses by country.

The MITF conducts observations using numerous honeypots. Here, however, we have taken the average per honeypot, showing the trends for incoming packet types (top ten) over the entire period subject to study. Many are TCP ports used by the Microsoft OS, searching for clients. The MITF also observed searching behavior for 2967/TCP used by Symantec client software.

At the same time, the MITF observed 11075/UDP, 20689/UDP—communications not used by most applications—the goals of which were not clearly identifiable. Attacks on 445/TCP etc. targeting the MS08-067 vulnerability have continued since last October.

Looking at the overall sender distribution by country, we see that attacks sourced to Japan and China, 36.1% and 23%, respectively, were comparatively higher than the rest.

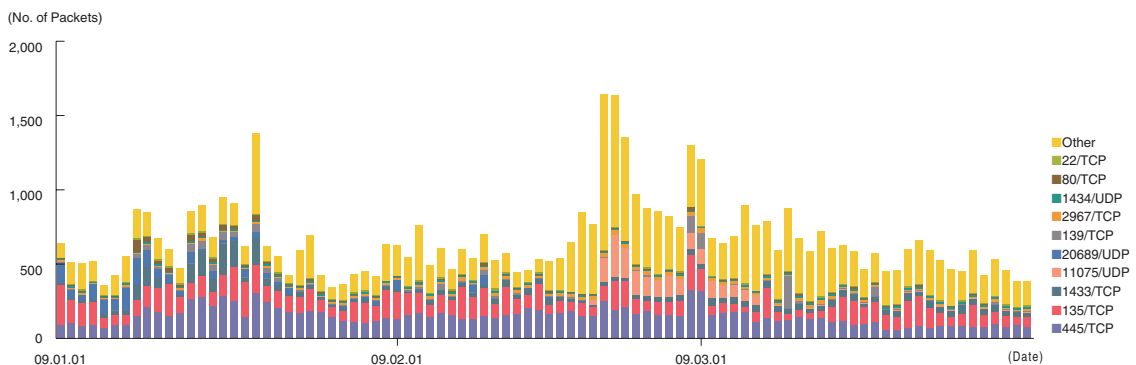


Figure 3: Communications Arriving at Honeypots (By Date, By Target Port, Per Honeypot)

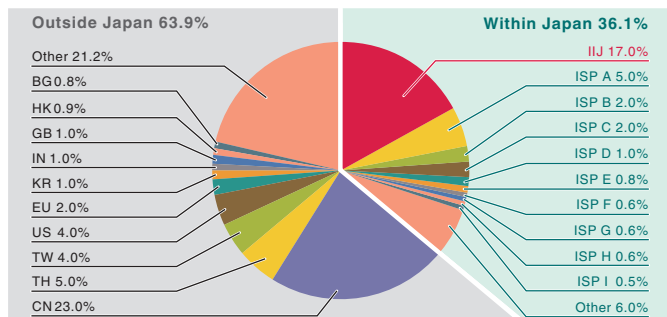


Figure 4: Sender Distribution (Entire Period under Study)

*17 Malware Investigation Task Force (MITF). The MITF began activities in May 2007 observing malware network activity through the use of honeypots in an attempt to understand the state of malware activities, to gather technical information for countermeasures, and to link these findings to actual countermeasures.

*18 A system designed to simulate damages from attacks by emulating vulnerabilities, recording the behavior of attackers, and the activities of malware.

■ Malware Network Activity

Next, let's take a look at the malware activity observed by the MITF. Figure 5 shows trends in the total number of malware specimens*19 acquired during the period under study. Figure 6 shows the distribution of the specimen acquisition source for malware. The trends in the number of acquired specimens show the total number of specimens acquired per day, while the number of unique specimens is the number of specimen variants categorized according to their hash values*20.

A total of 899 specimens were acquired per day on average during the period under study, representing about 44 different malware variants. According to prior statistics, the average daily total for acquired specimens was 2,235, with 55 different variants. Though we see a major decline in the number of specimens acquired, the number of variants remained basically unchanged.

The distribution of specimens according to source country has Japan at 70.1%, with other countries accounting for the 29.9% balance. Of the total, malware infection activity among IIJ users was 33.0%. This shows that malware infection activity continues to be extremely localized.

The MITF prepares analytical environments for malware, conducting its own independent analyses of specimens acquired. The results of these analyses show that, during the period under observation, 14% of the malware specimens were worms, 45% were bots, and 41% were downloaders. In addition, the MITF confirmed the presence of 86 botnet C&C servers*21 and 540 malware distribution sites.

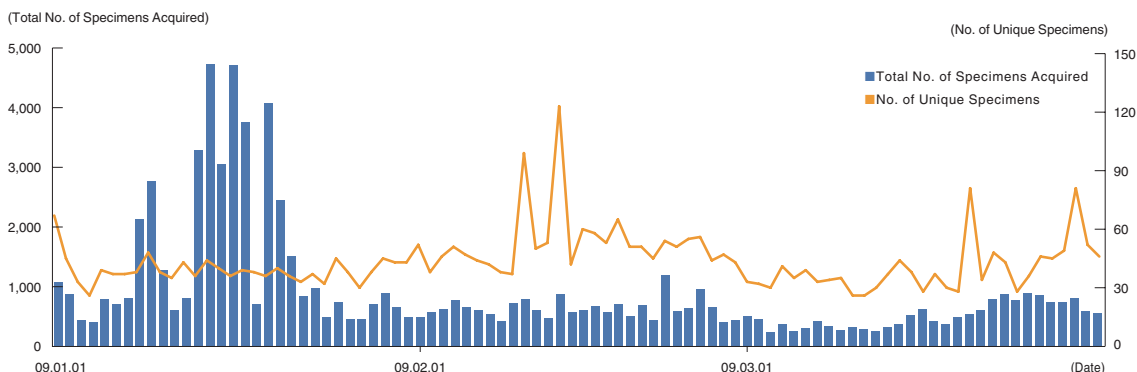


Figure 5: Trends in Number of Malware Specimens Acquired (Total Number, Number of Unique Specimens)

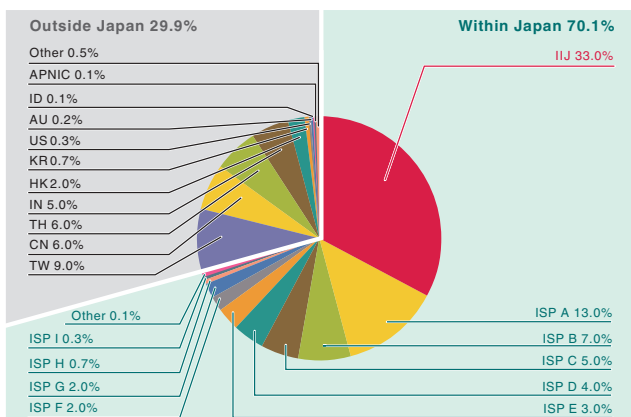


Figure 6: Distribution of Acquired Specimens by Source (Entire Period under Study)

*19 This indicates the malware acquired by honeypots.

*20 Figure derived by utilizing a one-way function (hash function) that outputs a fixed-length value for various input. The hash function is designed to produce as many different outputs as possible for different inputs. While we cannot guarantee the uniqueness of specimens by hash value, given that obfuscation and padding may result in specimens of the same malware having different hash values, the MITF has expended its best efforts to take this fact into consideration when using this methodology as a measurement index.

*21 Abbreviation for Command and Control Server. A server that sends commands to botnets comprised of numerous bots.

1.3.3 SQL Injection Attacks

Of the types of different Web server attacks, IJ conducts ongoing surveys related to SQL injection attacks*22. SQL injection attacks have flared up in frequency numerous times in the past, remaining one of the major topics in the Internet security. SQL injections are known to occur in one of three attack patterns: those that attempt to steal data, those that attempt to delete or rewrite Web content, and those that attempt to gain full control over servers.

Figure 7 shows trends of the numbers of SQL injection attacks against Web servers detected between January 1 and March 31, 2009. Figure 8 shows the distribution of attacks according to source. This data is a summary of attacks detected by signatures on the IJ Managed IPS Service. However, we have not included large scale attacks that have continued since the end of the prior year. Japan accounted as the source for 38.5% attacks, while South Korea and the United States represented 20.3% and 8.3%, respectively, with other countries accounting for the rest. As detailed in our previous report, there was a large-scale SQL injection attack against a few certain Web servers that began at the end of December 2008. These attacks dropped off rapidly beginning in 2009, quieting almost completely by the end of the first week of January.

The attacks above were properly detected and dealt with by the service. However, attack attempts continue, requiring ongoing attention.

See 1.4.1 “SQL Injection Attacks and their Impact” for more information about SQL injection attacks.

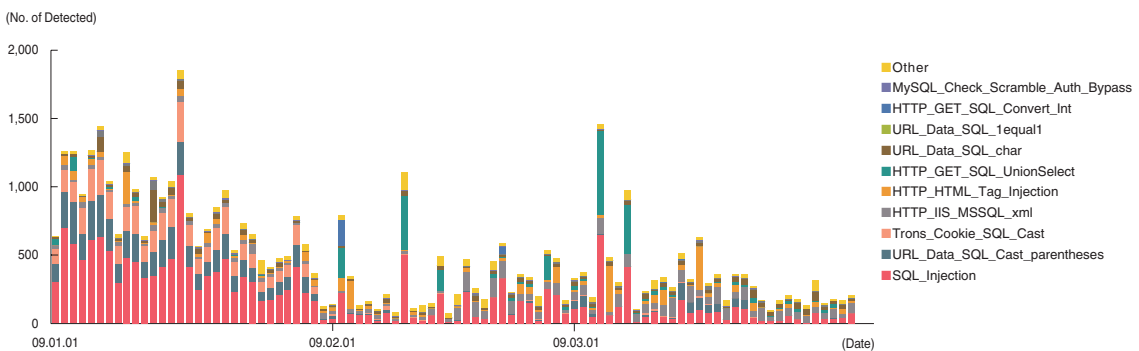


Figure 7: Trends of SQL Injection Attacks (By Day, By Attack Type)

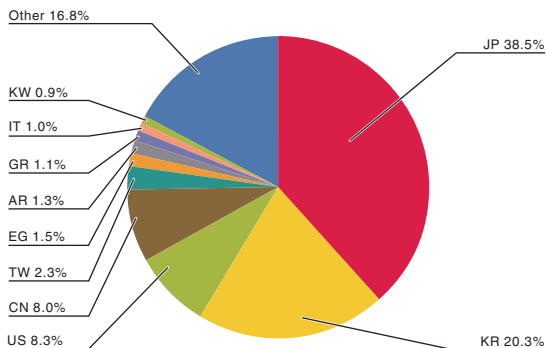


Figure 8: Distribution of SQL Injection Attacks by Source

*22 Attacks accessing a Web server to issue SQL commands, thereby connecting to and manipulating an underlying database. The database content can be accessed or altered without proper authorization; attackers can steal sensitive information, rewrite Web content, or otherwise issue system commands to control the database server.

1.4 Focused Research

Incidents occurring over the Internet change in type and scope almost from one minute to the next. Accordingly, IJ works toward taking countermeasures by performing independent surveys and analyses. In this section, we will discuss SQL injection attacks and their impact, alerts concerning ID/password management, and scareware from among surveys conducted during the period under study.

1.4.1 SQL Injection Attacks and their Impact

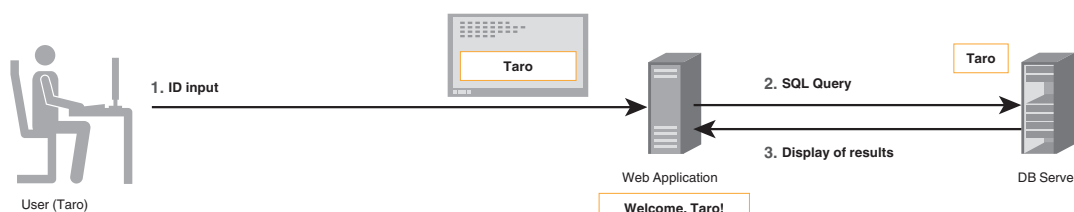
One method used to attack a Website is known as the SQL^{*23} injection attack. This type of attack uses remote requests to allow an attacker to perform unauthorized manipulation of a database (DB) used as a back end to a Website.

As discussed in “1.3.3 SQL Injection Attacks,” this type of attack occurs continually on the Internet. As a result of this type of attack, customer information stored on a DB can be stolen, Web content can be altered to embed malicious programs or to lead users to malicious sites, etc., causing direct and real damage to users.

■Anatomy of a SQL Injection Attack

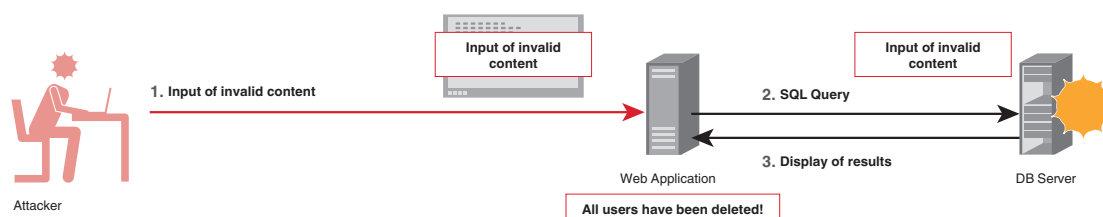
When a Web application makes an inquiry to a DB in response to a user request, a SQL statement is constructed for processing based on the user's input. If no special characters are included in the input value, the DB query is conducted as intended using the unmodified input values in the original SQL statement that serves as a template (Figure 9). However, if an attacker includes quotation marks or other special characters to create a text string that forms part of a SQL syntax, the SQL statement subsequently executed may result in different commands being sent to the DB than intended (Figure 10) if no input format checks or quotation mark escape characters are provided.

In this manner, a SQL statement different than the originally intended statement can be “injected,” serving as a form of attack to cause behavior other than that desired. This is called a SQL injection attack. This attack can be implemented by various means, including via text input forms, or through Cookie HTTP header or GET/POST parameters, depending on Web application or DB implementations. These methods are used to slip through detection.



Under normal access conditions, user input is transmitted to a DB through a Web application, and the relevant content is produced.

Figure 9: Normal Processing



With a SQL injection attack, invalid input is transferred to a DB through the Web application, resulting in unintended behavior.

Figure 10: SQL Injection Attack

*23 SQL is a query language used to give commands to a database for data operations (search, insert, edit, delete, etc.).

In many cases, no error messages or other traces are left behind by the Web application or DB to give evidence of a completed SQL injection attack. This makes detection difficult under normal monitoring conditions. Given the nature of this kind of attack, many incidents are only discovered after a notification is received from a third party. This is particularly true in the case of unauthorized content alteration.

■Impact of SQL Injection Attacks

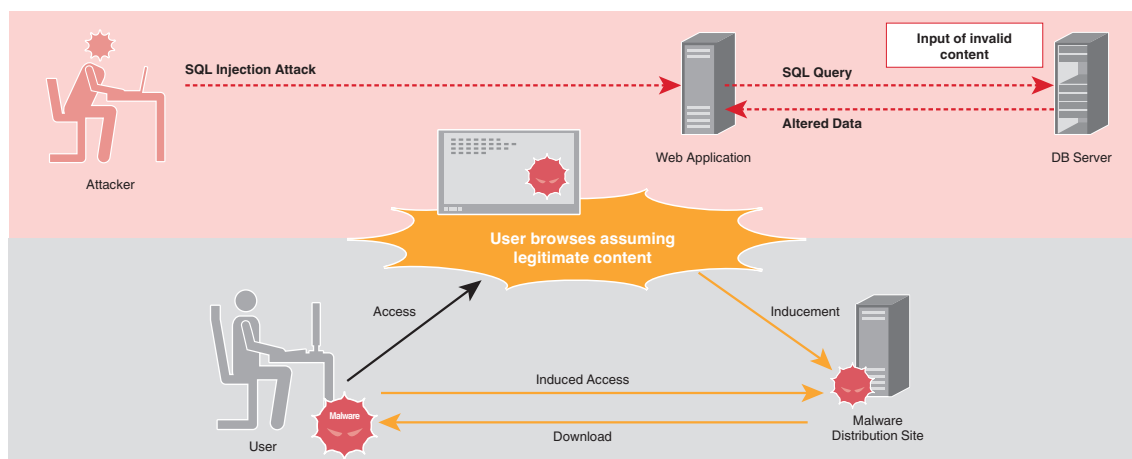
We can conceive of several different types of damages that can occur in the event of a successful SQL injection attack.

First, sensitive corporate or personal information stored in the database is at risk of theft or leakage. Second, attackers may delete or destroy data. Services over the Internet become impossible to provide if Web content and user information are deleted from service databases. Third, database content may be replaced with fraudulent content. If Web content is changed, an attacker can cause unintended information to be broadcast, leading visitors to malicious sites that cause malware infections, or other situations that present direct danger to the visitors. Fourth, a SQL injection attack can be the springboard for deeper systems penetration. An attacker may be able to set up a “back door” to control the system, using it as a stage to attack other systems across the network.

If left undetected or uncared for, successful SQL injection attacks can spread to cause significant damages to third parties and other systems.

■Attacking Users via Web Content Alteration

A recent increasing trend in SQL injection attacks is the alteration of Website content, and the installation of attack scripts that lead Website visitors to malware download sites*²⁴ (Figure 11). Since the Web server itself is legitimate, users are not aware that the content has been changed, resulting in the greater likelihood of malware infection. In these situations, it is difficult for users to recognize the counterfeit information. It is important that Website operators take more active countermeasures against such threats.



Content alteration via SQL injection attack not only involves the alteration of Web server content; unsuspecting Website visitors may also be subject to harm.

Figure 11: Attacking Users via Content Alteration

*24 For example, “Threats to Systems Administrators and Developers No. 1 The Power of Attacks through Legitimate Websites” as detailed in “10 Major Security Threats (<http://www.ipa.go.jp/security/vuln/10threats2009.html>)” published by the Information Technology Promotion Agency, Japan. (in Japanese)

■ Countermeasures

Last, we will offer a summary of measures that can be taken to prevent SQL injection attacks.

■ Care when Creating Web applications

When creating Web applications, Web developers must take care to design resistance to SQL injections. As mentioned previously, SQL injections are possible due to insufficient handling of invalid input text strings. First and foremost, developers can utilize a technique called a bind mechanism^{*25} when queries are made to the DB.

Other countermeasures include checking the format of text string input, limiting information contained in error messages to the smallest amount possible so as not to give away hints of vulnerabilities to attackers, and limiting access permissions to the DB appropriately. In addition, we also recommend that Web application operating tests include the use of software that tests Web application vulnerabilities, as well as the performance of third-party audits. For more detail, see the IPA's "How to Secure Your Web Site"^{*26} and the Open Web Application Security Project (OWASP) "SQL Injection"^{*27}.

■ Operational Countermeasures

Today's websites are built using multiple software components; be sure to pay attention to information regarding implementation vulnerabilities. The same applies, even when providing static content only. To gain an accurate understanding of circumstances surrounding an attack, administrators must create an environment allowing for the recording/storage of communications logs (POST data, cookies, SQL query statements, etc.).

To detect irregularities in their early stages, and to establish an understanding of the circumstances surrounding an attack, systems should be designed and managed so DB query errors are flagged as alerts requiring administrator investigation. Systems owners can adopt IDS or IPS and WAF^{*28}, or even look into hiring a managed security service provider.

1.4.2 Alerts Concerning ID/Password Management

■ Unauthorized ID Usage Occurs All Too Frequently

According to a news story in September 2008, an identity fraud incident involving the theft of IDs and passwords victimized a Japanese auction Website. Many individuals were charged for commissions on items they had no recollection of selling. Reportedly, one of the root causes of this incident was that users were using the same ID and password combinations across a multiple number of Websites. Alerts were sent out warning users against using the same passwords for different Websites. Toward the end of 2008, incidents involving the unauthorized use of IDs to alter Website content began to grow significantly in number. Website visitors were subjected to malicious content causing them to be exposed to malware. IJ confirmed some of the clients were numbered among the victims of such incidents.

*25 A bind mechanism is a function for safely handling text strings in a SQL statement that include special characters. By clearly separating the SQL statement syntax and the input values when creating text strings, Website developers can prevent the injection of invalid SQL statements.

*26 See "How to Secure Your Web Site" from the Information Technology Promotion Agency, Japan (http://www.ipa.go.jp/security/english/vuln/200806_websecurity_en.html).

*27 SQL injection countermeasures by OWASP (http://www.owasp.org/index.php/SQL_Injection).

*28 Web Application Firewall (WAF). A type of firewall that monitors Web communications, validating incoming input and outgoing content to prevent vulnerability exploits and unauthorized intrusions.

■ Password Strength

Strong password selection methods and means for correctly managing passwords have been in existence for many years. An IPA alert^{*29} details basic management methods for requiring periodic password changes and login history confirmation. SANS Password Policy^{*30} and other best practices regarding strong password creation have been publicly available on the Internet, and there also have been open source software^{*31} that automatically creates strong passwords meeting certain requirements. Utilizing these methods introduces basic functions for ensuring a strong password for any given ID.

■ The Difficulty of ID and Password Management

At the same time, Internet users now commonly make use of Internet shopping, SNS, blogs and other Web services. Many users use their email addresses as their IDs across a multiple number of Websites. Users are responsible for creating proper passwords on these Websites and exercising appropriate password management on their own. In other words, it is the Internet user who is in the position of having to create several different “strong passwords” at the various Websites whose services they frequent, periodically changing passwords for effective management. Of course, remembering so many different passwords is a difficult task, and users tend to “reuse” passwords among various Websites.

Given this situation, learning an ID and password at one service presents the potential for gaining fraudulent access to a multiple number of other services—a significant risk when sharing or allowing the disclosure of ID/password combinations.

■ Recommendations for Users

Many users opt to have their Web browsers and email software “remember” their ID and password combinations. Several Web browsers incorporate built-in or add-on functions that let the user know which ID and password they are using at each site. We encourage users who are in the habit of using the same ID and password across a number of Websites to create new, different password for each site. Users should consider the impact of others finding out their ID/password combinations, and be sure to create unique ID and password combinations for different sites, particularly on important online services including online banking and online shops.

Next, consider using a method for remembering multiple passwords. For example, a unified password management tool (Password Safe^{*32}, Password Wallet^{*33}, etc.) can be used to access all of the passwords with a single master password.

■ Recommendations for Administrators

From the standpoint of a corporate network administrator, the minimum for appropriate ID and password usage would be to train all members to not use their IDs or passwords at work for Websites of their personal use. While it isn't an easy matter to detect and identify cases in which an individual uses the same ID and password combinations for both work and personal use, the policy can be clearly stated within the organization, enhancing awareness by stressing the dangers of password “reusing” at work.

We recommend using systems that verify whether passwords being used on a corporate server can be easily guessed. In general, a method incorporating a dictionary^{*34} can be used.

*29 “Be sure to check your password(s) one more time!” from the Information Technology Promotion Agency, Japan (<http://www.ipa.go.jp/security/txt/2008/10outline.html>). (in Japanese)

*30 “SANS Password Policy” from the SANS Institute (http://www.sans.org/resources/policies/Password_Policy.pdf).

*31 For example, pwgen (<http://sourceforge.net/projects/pwgen/>).

*32 Password Safe (<http://www.schneier.com/passsafe.html>, <http://passwordsafe.sourceforge.net/>).

*33 Password Wallet (http://www.apple.com/downloads/macosx/productivity_tools/passwordwalleformacosxandiphone.html).

*34 For example, pam_cracklib, a LINUX PAM (Pluggable Authentication Module) that uses a dictionary to perform password checks (http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/sag-pam_cracklib.html).

■Summary

Herein, we have identified several points about which users and administrators should be aware regarding the management of IDs and passwords. Are you using the same ID and password combinations without much thought for both servers at work and services that deal with private information (SNS, blogs, etc.) or one-time contest entry Websites? We highly recommend that readers perform an ID and password inventory.

1.4.3 Scareware

In this section, we will discuss the recently emerging threat of “scareware”—potentially malicious software pretending to be security software. Scareware is the name for a type of malware (a combination of the words “scare” and “software”).

Scareware is software that acts as one component in a fraudulent scheme. When a user is browsing the Web, they are exposed to a message such as, “Your PC is infected by malware!” The effort is an attempt to fraudulently induce the user to purchase unnecessary software.

The following illustrates a typical scareware scheme, using the example of counterfeit anti-virus software:

1. When a user is browsing the Web, they are suddenly exposed to a pop-up message that reads, “Your computer might be infected with a virus. Do you want to check?” (Figure 12)

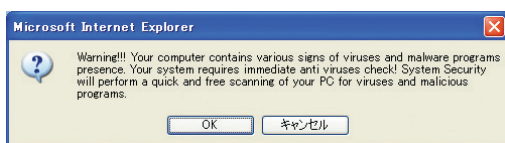


Figure 12: A Typical Scareware Pop-Up Message

2. Clicking the pop-up launches a screen that appears to be performing an online virus scan. (Figure 13)

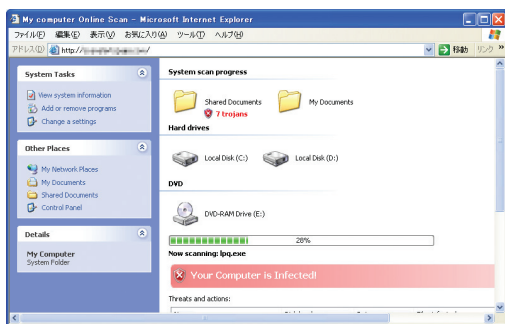


Figure 13: Fake Scan Screen

3. When the fake scan finishes, the user sees a pop-up screen such as the following: “Your computer is at risk. Immediately download and install anti-virus software.” (Figure 14)

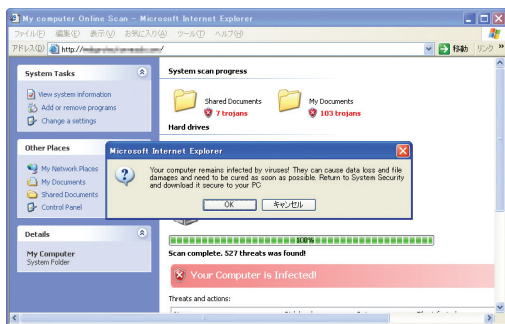


Figure 14: Fake Scan Results

- If the user clicks the pop-up, they are taken to the fake anti-virus software vendor Web page (Figure 15), or at which point a direct download of the scareware begins.

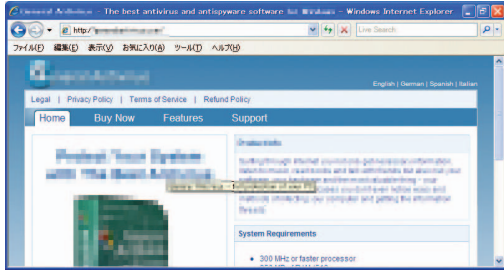


Figure 15: Example of a Fake Anti-Virus Software Vendor Web Page

- The user downloads the fake anti-virus software, and installs it on their computer.
- The fake anti-virus software will start automatically scanning the user's computer, falsely indicating that the PC is infected with numerous malware (Figure 16).

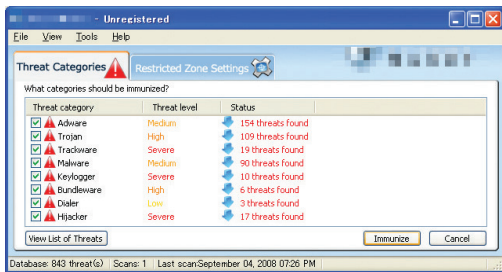


Figure 16: False Scan Results from Fake Anti-Virus Software

- When the user clicks the button to remove the “detected” malware, they are instructed to buy a paid-for version of the software. This scheme convinces users that their computer is infected with malware, and they are then fraudulently induced to purchase worthless software (Figure 17).

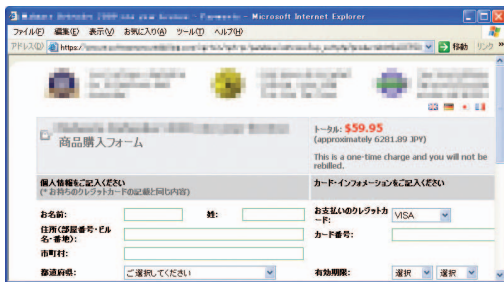


Figure 17: Example of a Fake Anti-Virus Software Purchase Screen

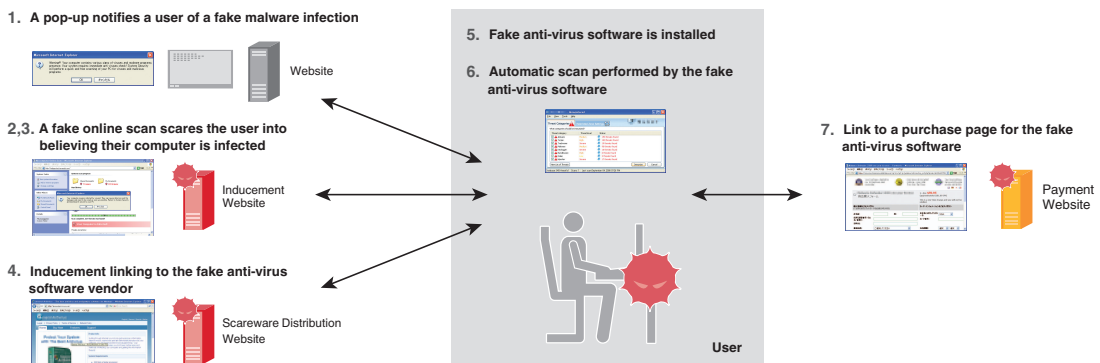


Figure 18: Anatomy of a Scareware Scam

In some cases, installed scareware will actually download and install malware on the pretext of an “update.” The preceding chain of events is shown in Figure 18.

As shown in Figures 15 and 16, fraudulent anti-virus software vendor Web pages and screens have the same look and feel as those of legitimate software vendors, leading us to believe that the damages caused by this type of fraud will continue to spread. We have even come across cases of scareware written in Japanese.

In addition to the fake anti-virus software used above as an example, we have confirmed the existence of fake firewall and anti-spyware schemes as well. To avoid being scammed, it is important that users make a habit of using legitimate security measures software obtained from reputable vendors. For example, users can locate trustworthy anti-virus software by selecting products introduced by trusted information sources^{*35}, or after confirming that a vendor is a member in good standing of an anti-virus industry association^{*36}.

1.5 Conclusion

This whitepaper has provided a summary of security incidents to which IIJ has responded.

In addition to reporting on regular matters, this volume (Vol. 3) has also covered information related to password management and scareware, etc.—issues that are becoming more prevalent, and for which investigations and countermeasure development are ongoing. These incidents have yet to be resolved completely.

By identifying and publicizing incidents and associated responses in whitepapers such as this, IIJ will continue to inform the public about the dangers of Internet usage, providing the necessary countermeasures to allow the safe and confident usage of this important component of the social and corporate infrastructure.

Authors:

Mamoru Saito

General Manager of the Division of Emergency Response and Clearinghouse for Security Information, IIJ Service Business Department

After working in security services development for enterprise customers, Mr. Saito became the representative of the IIJ Group emergency response team, IIJ-SECT in 2001, participating in FIRST, an international group of CSIRTs. Mr. Saito serves as a steering committee member of several industry groups, including Telecom-ISAC Japan, Nippon CSIRT Association, Information Security Operation provider Group Japan, and others.

Shigeki Ohara, Hirohide Tsuchiya (1.4.1 SQL Injection Attacks and their Impact)

Tadaaki Nagao, Yuji Suga (1.4.2 Alerts Concerning ID/Password Management)

Hiroshi Suzuki, Takeshi Umezawa (1.4.3 Scareware)

Division of Emergency Response and Clearinghouse for Security Information, IIJ Service Business Department

Yasunari Momoi (1.4.1 SQL Injection Attacks and their Impact)

Service Promotion Section, Security Service Division, IIJ Network Service Department

Contributors:

Yoshinobu Matsuzaki

Technology Promotion Section, Network Service Division, IIJ Network Service Department

Kiyotaka Doumae

Planning Section, Data Center Business Planning and Operations Division, IIJ Service Business Department

*35 For example, products featured by ISPs, or products from vendors listed on Microsoft's Website (<http://support.microsoft.com/kb/49500>).

*36 Examples of anti-virus industry associations: VIA (Virus Information Alliance) (<http://technet.microsoft.com/en-us/security/cc165596.aspx>); AMTSO (Anti-Malware Testing Standards Organization) (<http://www.amtso.org/members.html>); ASC (Anti-Spyware Coalition) (<http://www.antispywarecoalition.org/about/index.htm>); etc.