

IIR

Internet
Infrastructure
Review

Mar.2017

Vol. 34

Infrastructure Security

Ursnif (Gozi) Anti-Analysis Techniques and Methods for Bypassing Them

Technology Trends

The Current State of Library OSes

I I J

Internet Initiative Japan

Internet Infrastructure Review

March 2017 Vol.34

Executive Summary	3
1. Infrastructure Security	4
1.1 Introduction	4
1.2 Incident Summary	4
1.3 Incident Survey	11
1.3.1 DDoS Attacks.....	11
1.3.2 Malware Activities	13
1.3.3 SQL Injection Attacks.....	17
1.3.4 Website Alterations.....	18
1.4 Focused Research	19
1.4.1 Ursnif (Gozi) Anti-Analysis Techniques and Methods for Bypassing Them	19
1.5 Conclusion	27
2. Technology Trends	28
2.1 Introduction	28
2.1.1 Why Do We Need a New OS Now?	28
2.2 Related Research Projects	29
2.2.1 MirageOS	29
2.2.2 DrawBridge	29
2.2.3 Rump Kernel.....	29
2.3 The Linux Kernel Library	29
2.3.1 Why Was LKL Developed?.....	29
2.3.2 Design and Implementation	30
2.3.3 Performance.....	31
2.4 Conclusion	31

Executive Summary

2017 has begun, and as usual the first few months have flown by. With the Trump administration constantly in the news, and events like something out of a spy movie taking place, all eyes are fixed on overseas matters. Although the political situation in Japan is seemingly calm, recent incidents related to security intelligence are also important issues that concern the Internet, so I cannot help but feel that we will be called upon more and more to respond to events that cannot be resolved through technological theory alone.

This report discusses the results of the various ongoing surveys and analysis activities that IJ, as a service provider, carries out to support the Internet and cloud infrastructure, and enable our customers to continue to use them safely and securely. We also regularly present summaries of technological development as well as important technical information.

In Chapter 1, we provide a summary of Internet security over the past three months, discussing the many DDoS attacks targeting servers that have been observed, and the increase in attacks that saturate the upstream bandwidth of housing equipment. Many of these attacks use a composite approach, and attack techniques are also becoming more elaborate. In our focused research for this period, we examined the anti-analysis techniques of the Ursnif malware and ways to bypass them. We cover the cunning techniques that malware uses to impede its own analysis and connect to C&C servers, giving actual examples to explain methods to avoid this.

In Chapter 2, we discuss Library OSes. OSes have developed along with the evolution of computers, becoming extremely sophisticated. On the other hand, some source code has been made public, enabling everyone from students to researchers to participate in development through collaborative work on a global level. Library OSes are slightly different from OSes like these, as they support the design of OSes that can withstand flexible use, by stripping the kernel down to only the minimum necessary functions, and providing others in the form of libraries. In this chapter, a developer actually participating in development introduces the Linux Kernel Library and its content, so it is well worth a look.

As mentioned in press releases and other media, we launched the “wizSafe” brand that we announced last year under the concept of “making safety a matter of course,” and we are pouring our efforts into bolstering our security business.

Up until now, IJ has integrated security operations into our regular operational structure, utilizing operational technologies in a unified manner across all our technology departments. Going forward, we will overhaul our Security Operations Center, and use it to offer solutions such as the C-SOC Service to customers, enabling security intelligence information based on these operational technologies and IJ’s unique analytics platform to be utilized from the perspective of protecting customer systems.

As part of these initiatives, we will continue striving toward a better Internet, maintaining its stability and making it safer under the “wizSafe” banner, through this IIR and various other outlets for information.



Yoshikazu Yamai

Mr. Yamai is an Executive Managing Officer of IJ and Director of the Service Infrastructure Division. Upon joining IJ in June 1999, he was temporarily transferred to Crosswave Communications, Inc., where he was engaged in WDM and SONET network construction, wide-area LAN service planning, and data center construction, before returning to his post in June 2004. After his return he was in charge of IJ’s Service Operation Division. From April 2016 he joined the Infrastructure Operation Division, and now oversees the overall operation of corporate IT services at IJ. He also heads IJ’s data center operations, and he played a key role in the establishment of the modular “Matsue Data Center Park,” which was the first in Japan to use outside-air cooling.

Ursnif (Gozi) Anti-Analysis Techniques and Methods for Bypassing Them

1.1 Introduction

This report is a summary of incidents that IJ responded to, based on information obtained by IJ for the purpose of operating a stable Internet, information obtained from observed incidents, information obtained through our services, and information obtained from companies and organizations that IJ has cooperative relationships with. This volume covers the period of time from October 1 through December 31, 2016. In this period a number of hacktivism-based attacks were once again carried out by Anonymous and other groups. There were a large number of DDoS attacks and incidents of unauthorized access that led to many information leaks and website defacements. There were also ongoing DDoS attacks by botnets formed using malware that infect IoT devices, and incidents where identity fraud was used to carry out unauthorized login. As shown here, many security-related incidents continue to occur across the Internet.

1.2 Incident Summary

Here, we discuss incidents handled and responded to by IJ, between October 1 and December 31, 2016. Figure 1 shows the distribution of incidents handled during this period*1.

■ Activities of Anonymous and Other Hacktivist Groups

Attack activities by hacktivists such as Anonymous continued during this period. In correspondence with various events and assertions, DDoS attacks and information leaks targeted various companies and government-related sites.

Since 2013, there have been intermittent DDoS attacks thought to be made by Anonymous, as a protest against the drive hunting of dolphins and small whales in Japan. A statement declaring the continuation of this attack campaign was made as the fishing season opened on September 1, and attacks on websites in Japan have continued into October and beyond (OpKillingBay/OpWhales/OpSeaWorld). There have been many incidents of repeated attacks on websites that have been targeted before, as well as attacks on websites not on the list of attack targets. At the time of writing in January, the frequency of attacks has dropped slightly, but the attack campaign is still ongoing, so continued vigilance is required.

During this survey period, there was a series of incidents in which other websites in Japan were attacked, or added to the list of attack targets by Anonymous. In #OpCircus, one of the campaigns carried out in the name of animal rights, DoS attacks were conducted against the website of a circus troupe in Japan.

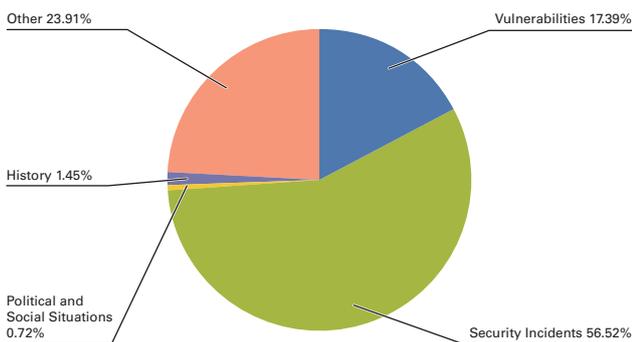


Figure 1: Incident Ratio by Category (October 1 to December 31, 2016)

The OpIcarus Phase 4 OpBlackOct campaign targeting financial institutions around the world began on October 1. Some financial institutions in Japan were included in the list of targets, but no significant damage occurred. Thailand's parliament passed a bill to reinforce Internet usage regulations in the country in December, which led to activists opposing the bill collecting signatures due to concerns that it may hinder freedom of expression. Anonymous also carried out the #OpSingleGateway campaign in protest against these amendments, compromising and launching DoS attacks against sites related to the Thai government. At the end of December, Thai law enforcement authorities had arrested nine

*1 Incidents in this report are split into five categories: vulnerabilities, political and social situations, history, security incidents or other.
Vulnerabilities: Responses to vulnerabilities in network equipment, server equipment or software commonly used across the Internet or in user environments.
Political and Social Situations: Responses to attacks stemming from international conferences attended by VIPs and international conflicts, and other related domestic and foreign circumstances and international events.
History: Warnings/alerts, detection and response to incidents for attacks that occur on the day of a historically significant date that have a close connection to a past event.
Security Incidents: Unexpected incidents and related responses such as wide spreading of network worms and other malware; DDoS attacks against certain websites.
Other: Security-related information, and incidents not directly associated with security problems, including high traffic volume associated with a notable event.

people in relation to these attacks. Financial institutions in Thailand were also listed as attack targets in OpSingleGateway, and because this included a number of Japanese financial institutions with branches in Bangkok, alerts were issued.

■ Vulnerabilities and Responses

During this period many fixes were released for Microsoft's Windows^{*2*}^{*3*}^{*4*}^{*5*}^{*6*}^{*7*}^{*8}, Internet Explorer^{*9*}^{*10*}^{*11}, Edge^{*12*}^{*13*}^{*14}, and Office^{*15*}^{*16}. Updates were also released for Adobe Systems' Flash Player, Acrobat, and Reader. A quarterly update was released for Oracle's Java SE, fixing many vulnerabilities. Several of these vulnerabilities were exploited in the wild before patches were released.

In server applications, a quarterly update was released for a number of Oracle products, such as the Oracle database server, fixing many vulnerabilities. A vulnerability in the BIND9 DNS server that caused the abnormal termination of named when processing responses containing a DNAME record in the answer section was discovered and fixed. The "BlackNurse" attack method that causes a high CPU load on certain firewall models using ICMP packets was also disclosed, and fixes were released by the corresponding vendors. Multiple vulnerabilities in the popular Joomla! CMS that could allow unauthorized user registration or user privilege escalation remotely were discovered and fixed^{*17}, but immediately following release of the patch, attacks exploiting these vulnerabilities were confirmed^{*18}. A vulnerability that could allow arbitrary code execution through an OS command injection was discovered and fixed in email sending libraries such as PHPMailer and SwiftMailer for PHP applications^{*19}.

A vulnerability that could allow arbitrary code execution through malicious TCP packets from an external source was discovered in the agent program for the SKYSEA Client View IT asset management tool. The developer released a fix after attacks exploiting this vulnerability were confirmed^{*20}.

-
- *2 "Microsoft Security Bulletin MS16-120 - Critical: Security Update for Microsoft Graphics Component (3192884)" (<https://technet.microsoft.com/en-us/library/security/MS16-120>).
 - *3 "Microsoft Security Bulletin MS16-122 - Critical: Security Update for Microsoft Video Control (3195360)" (<https://technet.microsoft.com/en-us/library/security/MS16-122>).
 - *4 "Microsoft Security Bulletin MS16-130 - Critical: Security Update for Microsoft Windows (3199172)" (<https://technet.microsoft.com/en-us/library/security/MS16-130>).
 - *5 "Microsoft Security Bulletin MS16-131 - Critical: Security Update for Microsoft Video Control (3199151)" (<https://technet.microsoft.com/en-us/library/security/MS16-131>).
 - *6 "Microsoft Security Bulletin MS16-132 - Critical: Security Update for Microsoft Graphics Component (3199120)" (<https://technet.microsoft.com/en-us/library/security/MS16-132>).
 - *7 "Microsoft Security Bulletin MS16-146 - Critical: Security Update for Microsoft Graphics Component (3204066)" (<https://technet.microsoft.com/en-us/library/security/MS16-146>).
 - *8 "Microsoft Security Bulletin MS16-147 - Critical: Security Update for Microsoft Uniscribe (3204063)" (<https://technet.microsoft.com/en-us/library/security/MS16-147>).
 - *9 "Microsoft Security Bulletin MS16-118 - Critical: Cumulative Security Update for Internet Explorer (3192887)" (<https://technet.microsoft.com/en-us/library/security/MS16-118>).
 - *10 "Microsoft Security Bulletin MS16-142 - Critical: Cumulative Security Update for Internet Explorer (3198467)" (<https://technet.microsoft.com/en-us/library/security/MS16-142>).
 - *11 "Microsoft Security Bulletin MS16-144 - Critical: Cumulative Security Update for Internet Explorer (3204059)" (<https://technet.microsoft.com/en-us/library/security/MS16-144>).
 - *12 "Microsoft Security Bulletin MS16-119 - Critical: Cumulative Security Update for Microsoft Edge (3192890)" (<https://technet.microsoft.com/en-us/library/security/MS16-119>).
 - *13 "Microsoft Security Bulletin MS16-129 - Critical: Cumulative Security Update for Microsoft Edge (3199057)" (<https://technet.microsoft.com/en-us/library/security/MS16-129>).
 - *14 "Microsoft Security Bulletin MS16-145 - Critical: Cumulative Security Update for Microsoft Edge (3204062)" (<https://technet.microsoft.com/en-us/library/security/MS16-145>).
 - *15 "Microsoft Security Bulletin MS16-121 - Critical: Security Update for Microsoft Office (3194063)" (<https://technet.microsoft.com/en-us/library/security/MS16-121>).
 - *16 "Microsoft Security Bulletin MS16-148 - Critical: Security Update for Microsoft Office (3204068)" (<https://technet.microsoft.com/en-us/library/security/MS16-148>).
 - *17 "Joomla! 3.6.4 Released" (<https://www.joomla.org/announcements/release-news/5678-joomla-3-6-4-released.html>).
 - *18 "Joomla Exploits in the Wild Against CVE-2016-8870 and CVE-2016-8869" (<https://blog.sucuri.net/2016/10/joomla-mass-exploits-privilege-vulnerability.html>).
 - *19 "Security notices relating to PHPMailer" (<https://github.com/PHPMailer/PHPMailer/blob/master/SECURITY.md>).
 - *20 "[Important] Alert regarding operation in global IP address environment (CVE-2016-7836) | SKYSEA Client View | Sky Co., Ltd." (<http://www.skyseaclientview.net/news/161221/>) (in Japanese).

October Incidents

1	V 6th: Multiple vulnerabilities in Adobe Acrobat and Reader that could allow unauthorized termination and arbitrary code execution were discovered and fixed. "Security Updates Available for Adobe Acrobat and Reader" (https://helpx.adobe.com/security/products/acrobat/apsb16-33.html).
2	
3	S 6th: The email management server of Yuryo Jutaku Loan, K.K. was accessed from an external source without authorization, emails of executives and staff were forwarded externally and customer information contained in these emails was leaked. "Regarding unauthorized access to our email management server and the potential leak of customers' personal information" (http://www.yuryoloan.jp/wp/wp-content/uploads/2016/10/20161005_YJL_release.pdf) (in Japanese). "Regarding the potential leak of our customers' personal information" (http://www.yuryoloan.jp/wp/wp-content/uploads/2016/10/20161026news_topics.pdf) (in Japanese).
4	
5	S 6th: The U.S. Department of Justice announced that a contract employee for the National Security Agency (NSA) had been arrested in August on the charge of stealing information that constituted a state secret. The suspect had taken document data classified as top secret and stored it in his home and car. "Government Contractor Charged with Removal of Classified Materials and Theft of Government Property" (https://www.justice.gov/usao-md/pr/government-contractor-charged-removal-classified-materials-and-theft-government-property).
6	
7	
8	S 8th: A staff member at Kwansei Gakuin University accessed a phishing site, leaking the personal information of students and graduates. Similarly worded phishing emails were received by a number of other people associated with the university, leading to an alert being issued. "Regarding the leak of personal information due to phishing site access" (http://www.kwansei.ac.jp/notice/2016/notice_20161007_013525.html) (in Japanese).
9	
10	S 10th: It was discovered that a PC used by staff at the Hydrogen Isotope Research Center of the University of Toyama had been infected by malware in November 2015, resulting in the leak of personal information of students at the university and people at other research institutions. Research-related information also leaked, but this was already known to the public, and did not constitute confidential information. "Regarding a targeted cyber attack against the University of Toyama's Hydrogen Isotope Research Center" (https://www.u-toyama.ac.jp/news/2016/1011.html) (in Japanese).
11	
12	V 11th: Multiple vulnerabilities in Adobe Flash Player that could allow unauthorized termination or arbitrary code execution were discovered and fixed. "Security updates available for Adobe Flash Player" (https://helpx.adobe.com/security/products/flash-player/apsb16-32.html).
13	
14	V 12th: Microsoft published their Security Bulletin Summary for October 2016, and released a total of 11 updates, including seven critical updates such as MS16-118, as well as three important updates. "Microsoft Security Bulletin Summary for October 2016" (https://technet.microsoft.com/library/security/ms16-oct).
15	
16	S 13th: A tourist information site for foreigners run by Tottori Prefecture was accessed by an external party without authorization, and approximately 500,000 emails were sent to a large number of unspecified targets. "Tottori Prefecture tourist information site for foreigners accessed by a third party without authorization / News report / Tori Net / Tottori Prefecture Official Website" (http://db.pref.tottori.jp/pressrelease.nsf/5725f7416e09e6da492573cb001f7512/8948346D7BCF82434925804B0018D682?OpenDocument) (in Japanese).
17	
18	V 18th: Oracle released their quarterly scheduled update for multiple products including Java SE and Oracle Database Server, fixing a total of 253 vulnerabilities. "Oracle Critical Patch Update Advisory - October 2016" (http://www.oracle.com/technetwork/security-advisory/cpuoct2016-2881722.html).
19	
20	S 18th: An incident of unauthorized login through impersonation occurred on the "ann-kate" site operated by Marketing Applications Inc., leading to the alteration of email addresses and unauthorized conversion of loyalty points.
21	
22	S 21st: The DNS service of U.S. company Dyn was targeted by a DDoS attack that used Mirai bots, causing failures that affected services for many customers, including Twitter and Spotify. "Dyn, Inc. Status - DDoS Attack Against Dyn Managed DNS" (https://www.dynstatus.com/incidents/nlr4yrr162t8). "Dyn, Inc. Status - Update Regarding DDoS Event Against Dyn Managed DNS on October 21, 2016" (https://www.dynstatus.com/incidents/5r9mppc1kb77).
23	
24	S 21st: It came to light that the information of about 43,430,000 users had leaked from Weeble.
25	V 24th: Apple released iOS 10.1, macOS Sierra 10.12.1, and security updates for OS X, fixing multiple vulnerabilities, including those that could allow a remote attacker to execute arbitrary code. Also, tvOS 10.0.1 and watchOS 3.1 were released. "About the security content of iOS 10.1" (https://support.apple.com/en-us/HT207271). "About the security content of macOS Sierra 10.12.1, Security Update 2016-002 El Capitan, and Security Update 2016-006 Yosemite" (https://support.apple.com/en-us/HT207275). "About the security content of tvOS 10.0.1" (https://support.apple.com/en-us/HT207270). "About the security content of watchOS 3.1" (https://support.apple.com/en-us/HT207269).
26	
27	S 27th: An incident of unauthorized login through impersonation occurred on the website of Matsumotokiyoshi, resulting in the unauthorized use of loyalty points on some accounts. "Report regarding unauthorized login on our website" (http://www.matsukiyo.co.jp/online/html/info/info20161027.html) (in Japanese).
28	
29	
30	S 31st: It was discovered that emails were being sent under the name of ICT-ISAC Japan, informing recipients that they were infected with malware and prompting them to download a removal tool. It was also learned that those following the instructions in these emails would be infected with ransomware. "[Alert] Regarding fraudulent emails misrepresented as being from us ICT-ISAC Japan" (https://www.ict-isac.jp/news/news20161031.html) (in Japanese).
31	

*Dates are in Japan StandardTime

Legend	V Vulnerabilities	S Security Incidents	P Political and Social Situation	H History	O Other
--------	--------------------------	-----------------------------	---	------------------	----------------

■ Attack Activity by IoT Botnets

DDoS attacks by botnets leveraging IoT devices infected with malware such as Mirai that were discussed in the previous volume continued to be observed during this survey period.

On October 21, the DNS server infrastructure of major U.S. DNS service provider Dyn was targeted by Mirai bots^{*21} in a DNS water torture attack^{*22}, and failures occurred on two separate occasions^{*23}. These failures resulted in widespread damage, with many services such as Twitter, Spotify, and Reddit being unreachable for several hours. On September 20, Mirai bots conducted a large-scale DDoS attack of around 623 Gbps against the “Krebs on Security” blog site of Brian Krebs, and the malware creator subsequently published the source code, attracting a lot of interest. Mr. Krebs speculates that the attack on Dyn may have been triggered by the results of an investigation into BGP hijackings by a DDoS protection service, which was conducted together with Dyn researchers and presented at NANOG^{*24*25}. Note that while some articles described the scale of the attack as 1.2 Tbps, which would be one of the largest ever, this figure is unverified and not confirmed by Dyn themselves, so it is unclear whether or not the attack was really this large.

In late November, faults occurred on the routers of around 900,000 users of a Deutsche Telekom communication service, causing issues such as the inability to connect to the Internet^{*26}. This was caused by malware making repeated attempts to infect routers through the management interface, and failing to do so^{*27}. The malware that carried out these infection attempts is a variant of Mirai that uses a management protocol called TR-064: LAN-Side DSL CPE Configuration, in an attempt to infect by exploiting a vulnerability^{*28} in implementations within certain devices. TR-064 is a management protocol for configuring CPE (Customer Premises Equipment) such as routers installed within the premises of a communications line user from a PC. This protocol was originally intended to be used within a LAN on premise, but in this case it was made available for use on the Internet as well, which contributed to the vulnerability being exploited. This vulnerability is also present in routers used by telecommunications companies in other regions, including Ireland and the U.K., and it has been confirmed that it affected services in these areas during the same period as well. In National Police Agency observations, it was reported that access to the 7547/TCP and 5555/TCP ports that this management protocol uses rose sharply during this period^{*29}. In December, activities of a Mirai variant that also scanned ports such as 37777/TCP, 23231/TCP, and 6789/TCP for infection attempts were observed on a global scale^{*30}. See also “1.3.2 Malware Activities” for information on the status of scanning for each port.

On October 5, the U.S. Department of Justice announced that two key members of the hacker groups called Lizard Squad and PoodleCorp had been arrested in the U.S. and the Netherlands in September^{*31}. Lizard Squad runs Shenron, a DDoS-for-hire service (a service that conducts DDoS attacks on behalf of others) also known as booter/stresser. PoodleCorp provided a similar

*21 See “1.4.1 Mirai Botnet Detection and Countermeasures” in the Focused Research section of Vol.33 of this report (<http://www.ij.ad.jp/en/company/development/iir/033.html>) for more information about Mirai bots.

*22 See the following materials by Mr. Morishita of Japan Registry Services for more information about DNS water torture attacks. “DNS Water Torture Attacks” (http://2014.secon.jp/dns/dns_water_torture.pdf) (in Japanese).

*23 “Dyn Analysis Summary Of Friday October 21 Attack | Dyn Blog” (<https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/>).

*24 “BackConnect’s Suspicious BGP Hijacks” (https://www.nanog.org/sites/default/files/20161016_Madory_Backconnect_S_Suspicious_Bgp_v2.pdf).

*25 “DDoS on Dyn Impacts Twitter, Spotify, Reddit – Krebs on Security” (<https://krebsonsecurity.com/2016/10/ddos-on-dyn-impacts-twitter-spotify-reddit/>).

*26 “Deutsche Telekom: Information on current problems” (<https://www.telekom.com/en/media/media-information/archive/information-on-current-problems-444862>).

*27 “WERE 900K DEUTSCHE TELEKOM ROUTERS COMPROMISED BY MIRAI? - Comsecuris Security Research & Consulting Blog” (https://comsecuris.com/blog/posts/were_900k_deutsche_telekom_routers_compromised_by_mirai/).

*28 This vulnerability was first reported by security researchers on November 7, by which point a PoC had also been released. “Eir’s D1000 Modem Is Wide Open To Being Hacked. - Reverse Engineering Blog” (<https://devicereversing.wordpress.com/2016/11/07/eirs-d1000-modem-is-wide-open-to-being-hacked/>).

*29 “Regarding the rapid rise in communications targeting a vulnerability in foreign-made routers” (National Police Agency) (<https://www.npa.go.jp/cyberpolice/detect/pdf/20161221.pdf>) (in Japanese).

*30 “Regarding the rapid rise in communications thought to represent the infection activity of a ‘Mirai’ bot variant (December 2016)” (National Police Agency) (<https://www.npa.go.jp/cyberpolice/detect/pdf/20170120.pdf>) (in Japanese).

*31 “American and Dutch Teenagers Arrested on Criminal Charges for Allegedly Operating International Cyber-Attack-For-Hire Websites | USAO-NDIL | Department of Justice” (<https://www.justice.gov/usao-ndil/pr/american-and-dutch-teenagers-arrested-criminal-charges-allegedly-operating>).

November Incidents

1	V 2nd: A vulnerability in the processing of DNS responses in BIND9 that could allow DoS attacks from an external source was discovered and fixed. "CVE-2016-8864: A problem handling responses containing a DNAME answer can lead to an assertion failure Internet Systems Consortium Knowledge Base" (https://kb.isc.org/article/AA-01434).
2	S 2nd: An incident of unauthorized login through impersonation occurred at the Canon Online Shop run by Canon Marketing Japan Inc., resulting in the unauthorized purchase of products. "Canon: Notice of on transactions made through unauthorized use of member IDs and passwords at the 'Canon Online Shop,' and password change request" (http://cweb.canon.jp/caution/161102.html) (in Japanese).
3	
4	
5	S 7th: A total of 2.5 million pounds was withdrawn from the accounts of about 9,000 customers of U.K. Tesco Bank via unauthorized login. "Tesco Bank announces full service has resumed for customers (8 November, 2016) - News releases - News - Tesco Bank" (http://corporate.tescobank.com/25/news/news-releases/tesco-bank-announces-full-service-has-resumedfor-customers/?newsid=291).
6	
7	V 8th: Multiple vulnerabilities in Adobe Flash Player that could allow unauthorized termination or arbitrary code execution were discovered and fixed. "Security updates available for Adobe Flash Player" (https://helpx.adobe.com/security/products/flash-player/apsb16-37.html).
8	O 8th: In China, the National People's Congress was held, and a new cyber security bill was passed. This requires companies to have users register under their real names, and store the personal information of users within China's borders.
9	
10	V 9th: Microsoft published their Security Bulletin Summary for November 2016, and released a total of 14 updates, including six critical updates such as MS16-129, as well as eight important updates. "Microsoft Security Bulletin Summary for November 2016" (https://technet.microsoft.com/library/security/ms16-nov).
11	
12	S 10th: DDoS attacks were conducted against at least five major Russian banks. It is believed that a DDoS-for-hire service was used. "DDoS attack on the Russian banks: what the traffic data showed - Securelist" (https://securelist.com/blog/incidents/76728/ddos-attack-on-the-russian-banks-what-the-traffic-data-showed/).
13	V 11th: The "BlackNurse" DoS attack method that causes a high CPU load on certain firewall models using ICMP packets was disclosed. "BLACKNURSE it can bring you down" (http://www.blacknurse.dk/).
14	
15	S 14th: It was discovered that account information for approximately 412 million users had leaked from Friend Finder Networks, which runs online dating sites and other services.
16	S 15th: An incident of unauthorized login through impersonation occurred on the "Yorisou e Net" Web service of Tohoku Electric Power, resulting in the unauthorized use of loyalty points on some accounts. The service was suspended for an investigation, and resumed on December 1, after taking measures such as changing the login IDs. "Regarding suspension of service due to unauthorized access to 'Yorisou e Net'" Tohoku Electric Power" (http://www.tohoku-epco.co.jp/news/normal/1193122_1049.html) (in Japanese). "Regarding the 'Yorisou e Net' service restart and changing of login IDs" (https://www3.zf1.tohoku-epco.co.jp/terms/announce.html) (in Japanese).
17	
18	
19	S 18th: An incident of unauthorized login through impersonation occurred on the m3.com site. "Notice of unauthorized login to the m3.com site and password change request" (https://corporate.m3.com/2016/11/18/m3.com-20161118.pdf) (in Japanese).
20	
21	S 21st: An incident of unauthorized login through impersonation occurred on a members-only website for Lawson, resulting in the unauthorized use of loyalty points on some accounts. "Request for customers using the Lawson WEB membership service site to reset their passwords Lawson" (http://www.lawson.co.jp/company/news/detail/1284326_2504.html) (in Japanese).
22	
23	S 22nd: Similarly worded emails threatening to detonate bombs in city offices and other civic facilities were received in many cities around Japan, including Tokushima and Chiba, but no suspicious objects were discovered. "Response to bomb threats against the main building of Tokushima City Office : Official Website of Tokushima City" (https://www.city.tokushima.tokushima.jp/enzen/shoubou_bousai/kikikanrijyoho/enzen_20161121.html) (in Japanese). "Chiba City: No suspicious objects or explosions confirmed regarding bomb threats against civic facilities" (http://www.city.chiba.jp/somu/kikikanri/bakuhayokoku.html) (in Japanese).
24	
25	
26	S 28th: The San Francisco Municipal Transportation Agency (SFMTA) was hit with ransomware that affected around 900 computers. Payment systems at subway stations were temporarily suspended to limit the spread of damages. "Update on SFMTA Ransomware Attack SFMTA" (https://www.sfmta.com/about-sfmta/blog/update-sfmta-ransomware-attack).
27	
28	S 29th: A fault occurred on the routers of around 900,000 users of a Deutsche Telekom communication service, causing issues such as the inability to connect to the Internet. It is believed that this was caused by the infection activity of a Mirai malware variant. "Deutsche Telekom: Information on current problems" (https://www.telekom.com/en/media/media-information/archive/information-on-current-problems-444862).
29	
30	S 29th: An incident of unauthorized login through impersonation occurred on the "Ameba" site run by CyberAgent, and in response the passwords for about 590,000 affected accounts were reset. "Notice regarding unauthorized logins to 'Ameba' and password reset request CyberAgent, Inc." (https://www.cyberagent.co.jp/newsinfo/press/detail/id=12977) (in Japanese).

*Dates are in Japan StandardTime

Legend	V Vulnerabilities	S Security Incidents	P Political and Social Situation	H History	O Other
---------------	--------------------------	-----------------------------	---	------------------	----------------

service called Poodle Stresser. Both of these services are thought to have carried out DDoS attacks using the LizardStresser IoT botnet. The source code for LizardStresser was released in 2015, so it has been used by various attackers to build a large number of botnets and launch DDoS attacks. Both of those arrested were 19-year-old youths, who had used these services to repeatedly conduct DDoS attacks against major game companies, among others.

IoT botnets attempt to infect IoT devices where management ports such as Telnet are open by logging in using default credentials. Some IoT devices have issues, such as not making it possible for users to change the factory set default password for the administrator account, or allowing users to close the Telnet or other ports used for administration. Some product vendors have recognized these issues because of the spread of infections due to Mirai bots, and stopped shipping the products while also releasing firmware that addresses the issue^{*32}. Based on this situation, the IPA^{*33} and JPCERT/CC^{*34} have issued warnings to IoT device users urging them to take appropriate measures, such as changing the default password before use and updating the firmware.

■ Russian Cyber Attacks Related to the U.S. Presidential Election

2016 was the year of the U.S. presidential election held every four years, and the impact of cyber attacks on election campaigns attracted tremendous attention.

In June it was discovered that the Democratic National Convention (DNC) had been compromised by an external party, resulting in the leak of internal information related to election campaigns. There was also a series of information releases that included emails and documents mainly related to the Democratic Party, including some data believed to have leaked from the DNC, by entities such as WikiLeaks, Guccifer 2.0, and DCLeaks. The impact that this unfavorable content to the Democratic Party had on the election results is not clear, but in the end the Republican Party candidate Donald Trump was elected president in the November presidential election.

In May, the Office of the Director of National Intelligence (ODNI) had warned that cyber attacks were being conducted against multiple organizations in the United States related to the presidential election. In response to the results of an internal investigation into this series of events, the Department of Homeland Security (DHS) and the ODNI issued a joint statement regarding the security of the presidential election in October^{*35}. The statement criticized the Russian government by name for being responsible for compromising the DNC and related organizations, the leaking of emails, and other acts. Also, in December after the presidential election, President Obama announced he would amend the content of a previous Executive Order (Executive Order 13694)^{*36}, which would trigger sanctions, including the deportation of 35 Russian diplomats due to the Russian government interfering with the U.S. presidential election. A Joint Analysis Report by the DHS and the FBI, naming this series of attack activities as GRIZZLY STEPPE, was also released at the same time^{*37}. As this demonstrates, the U.S. government is implementing a variety of measures based on the assertion that these cyber attacks were conducted by Russia. However, the information released lacks conclusive evidence, leading to criticism of the content from a number of security experts^{*38}.

*32 For example, these measures have been implemented for IO DATA DEVICE's "WFS-SR01" and Princeton's "PTW-WMS1." "Regarding a security vulnerability on the Wi-Fi storage "WFS-SR01" | IO DATA DEVICE" (<http://www.iodata.jp/support/information/2016/wfs-sr01/>) (in Japanese). "To customers who use the PTW-WMS1 connected to a modem without a router function - notice of firmware update | Support Information | List of Notices | Princeton Ltd." (<http://www.princeton.co.jp/news/2016/12/201612271100.html>) (in Japanese).

*33 "Be sure to change the password for IoT devices such as network cameras and home routers before using them - Safety Consultation Service News: IPA - Information-technology Promotion Agency, Japan" (<https://www.ipa.go.jp/security/anshin/mgdayori20161125.html>) (in Japanese)

*34 "Alert regarding the management of Internet-connected devices" (<https://www.jpCERT.or.jp/at/2016/at160050.html>) (in Japanese).

*35 "Joint Statement from the Department of Homeland Security and Office of the Director of National Intelligence on Election Security" (<https://www.odni.gov/index.php/newsroom/press-releases/215-press-releases-2016/1423-joint-dhs-odni-election-security-statement>).

*36 "Executive Order -- Taking Additional Steps to Address the National Emergency with Respect to Significant Malicious Cyber-Enabled Activities | whitehouse.gov" (<https://obamawhitehouse.archives.gov/the-press-office/2016/12/29/executive-order-taking-additional-steps-address-national-emergency>).

*37 "GRIZZLY STEPPE - Russian Malicious Cyber Activity" (<https://www.us-cert.gov/security-publications/GRIZZLY-STEPPE-Russian-Malicious-Cyber-Activity>).

*38 For example, the following articles were critical of the report. "Critiques of the DHS/FBI's GRIZZLY STEPPE Report - Robert M. Lee" (<http://www.robertmlee.org/critiques-of-the-dhsfbi-grizzly-steppe-report/>). "FBI/DHS Joint Analysis Report: A Fatally Flawed Effort" (<https://medium.com/@jeffreycarr/fbidhs-joint-analysis-report-a-fatally-flawed-effort-b6a98f8e2fa>).

December Incidents

1	S 2nd: A report revealed that a total of two billion rubles (about 31 million dollars) had been stolen from accounts at the Central Bank of Russia and private banks through cyber attacks over the past year.
2	S 2nd: An incident of unauthorized login through impersonation occurred on some pixiv accounts, and the passwords of corresponding users were reset.
3	"[pixiv] Announcements - [Important] Report on login through identity fraud affecting some pixiv accounts and password change request" (http://www.pixiv.net/info.php?id=3897) (in Japanese).
4	S 2nd: A coordinated effort by the European Police Office (Europol), the U.S. Department of Justice, etc. uncovered the Avalanche network that had been used to carry out activities such as malware infections, leading to the arrest of five members and the seizure of servers.
5	"'Avalanche' network dismantled in international cyber operation Europol" (https://www.europol.europa.eu/newsroom/news/%E2%80%98avalanche%E2%80%99-network-dismantled-in-international-cyber-operation).
6	V 5th: Researchers at Newcastle University announced an issue with the VISA payment network. They showed it was possible to break authentication based on card number, expiry date, and CVV in a matter of seconds using a "Distributed Guessing Attack."
7	"Cyber attack - Press Office - Newcastle University" (http://www.ncl.ac.uk/press/news/2016/12/cyberattack/).
8	S 6th: An incident of unauthorized login occurred on the "Cocokara Club" and "Cocokara Official App" run by cocokara fine Inc., resulting in the unauthorized use of loyalty points on some accounts.
9	"Regarding unauthorized access to our 'Cocokara Club' and 'Cocokara Official App' [Cocokara Club] Cocokara Fine Drugstore" (http://www.cocokarafine.co.jp/info/CSfViewNews.jsp?sort=1&no=24) (in Japanese).
10	S 6th: It was discovered that about 85 million pieces of account information had leaked from French video sharing service Dailymotion.
11	"Dailymotion accounts security update - Dailymotion Official Blog" (http://blog.dailymotion.com/en/dailymotion-account-security-update/).
12	S 8th: South Korea's Defense Ministry announced that a PC connected to the national defense intranet of the South Korean military had been infected with malware, leading to the leak of data including military secrets.
13	"국방망 해킹 관련 설명자료" (http://www.mnd.go.kr/user/boardList.action?command=view&page=1&boardId=L_42745&boardSeq=L_4009863&mcCategoryId=&id=mnd_020500000000) (in Korean).
14	O 8th: The Ministry of Economy, Trade and Industry published their "Cybersecurity Management Guidelines Ver 1.1." The IPA also published its "Cybersecurity Management Guidelines Manual," which supplements the content of the guidelines and provides concrete explanations of how to implement it.
15	"Cybersecurity Management Guidelines (METI/Ministry of Economy, Trade and Industry)" (http://www.meti.go.jp/policy/netsecurity/mng_guide.html) (in Japanese). "Cybersecurity Management Guidelines Manual: IPA Information-technology Promotion Agency, Japan" (https://www.ipa.go.jp/security/economics/csmgl-kaisetsusho.html) (in Japanese).
16	O 8th: The National center of Incident readiness and Strategy for Cybersecurity (NISC) held a cross-sectoral exercise for critical infrastructure.
17	"Summary of Cross-sectoral Exercise for Critical Infrastructure [2016 Annual Cross-sectoral Exercise]" (http://www.nisc.go.jp/active/infra/pdf/bunya_enshu2016gaiyou.pdf) (in Japanese).
18	V 12th: Apple released iOS 10.2, macOS Sierra 10.12.2, and security updates for OS X, fixing multiple vulnerabilities, including those that could allow a remote attacker to execute arbitrary code. Also, tvOS 10.1 was released.
19	"About the security content of iOS 10.2" (https://support.apple.com/en-us/HT207422). "About the security content of tvOS 10.1" (https://support.apple.com/en-us/HT207425). "About the security content of macOS Sierra 10.12.2, Security Update 2016-003 El Capitan, and Security Update 2016-007 Yosemite" (https://support.apple.com/en-us/HT207423).
20	V 13th: Multiple vulnerabilities in Adobe Flash Player that could allow unauthorized termination or arbitrary code execution were discovered and fixed.
21	"Security updates available for Adobe Flash Player" (https://helpx.adobe.com/security/products/flash-player/apsb16-39.html).
22	V 14th: Microsoft published their Security Bulletin Summary for December 2016, and released a total of 12 updates, including six critical updates such as MS16-144, as well as six important updates.
23	"Microsoft Security Bulletin Summary for December 2016" (https://technet.microsoft.com/library/security/ms16-dec).
24	S 18th: Ukrainian power company Ukrenergo announced that a blackout had occurred due to a cyber attack.
25	"Щодо аварійної ситуації на підстанції 330 кВ "Північна" (http://www.ukrenergo.energy.gov.ua/Pages/ua/DetailsNew.aspx?nID=3387) (in Ukrainian).
26	O 20th: The Japanese Cabinet issued an order that the amended Act on the Protection of Personal Information was to be enacted on May 30, 2017.
27	P 30th: President Obama of the United States imposed sanctions through an Executive Order (Executive Order 13694), including the deportation of Russian diplomats, due to the Russian government's interference in the presidential election.
28	"Statement by the President on Actions in Response to Russian Malicious Cyber Activity and Harassment" (https://obamawhitehouse.archives.gov/the-press-office/2016/12/29/statement-president-actions-response-russian-malicious-cyber-activity).
29	
30	
31	

*Dates are in Japan StandardTime

Legend	V Vulnerabilities	S Security Incidents	P Political and Social Situation	H History	O Other
---------------	--------------------------	-----------------------------	---	------------------	----------------

■ Government Agency Initiatives

Proposed revisions to the “Basic Act on Cyber Security” were passed by the Diet in April 2016, and Article 13 of the act expanded the targets of cybersecurity initiatives to include not just central ministries, but also independent administrative agencies and special government-affiliated corporations. At the 10th assembly of the Cyber Security Strategic Headquarters held in October 2016, nine organizations were newly designated based on the provisions of Article 13*³⁹. The Japan Pension Service, which was responsible for information leaks that took place in 2015, was included here. Through this new designation, these organizations will be subject to state inspections and the monitoring of information systems for malicious activity, and investigation to determine the cause of any incidents that occur, just like central ministries.

On December 7, the National center of Incident readiness and Strategy for Cybersecurity (NISC) held the 11th Cross-sectoral Exercise for Critical Infrastructure, with around 2,000 operators of critical infrastructure participating. This exercise checked the effectiveness of measures and systems at critical infrastructure operators, including the sharing of information and coordination with stakeholders when IT failures occur.

■ Other

In 2016, it was discovered that leaks of a large amount of password information had occurred on many services such as MySpace and LinkedIn in the past. In December, U.S. company Yahoo! announced that at least a billion pieces of user information had leaked as of 2013*⁴⁰. This was the largest ever information leak known to date. Yahoo! had just announced in September that information of at least 500 million users had leaked as of 2014, but with this, it came to light that even larger-scale leaks had occurred in addition to these.

Due to password information leaks becoming a regular occurrence both in Japan and overseas, there have been many incidents of unauthorized login through impersonation, or so-called list-based attacks, targeting users that use the same password across multiple services. During the current survey period, such incidents were confirmed on multiple membership-based websites in Japan, resulting in the unauthorized use of loyalty points. Users need to take into account the possibility that password information may be leaked and exploited, and take preventative measures such as not using the same password across multiple services.

1.3 Incident Survey

1.3.1 DDoS Attacks

Today, DDoS attacks on corporate servers are almost a daily occurrence, and the methods involved vary widely. However, most of these attacks do not utilize advanced knowledge such as vulnerabilities, but aim to hinder or delay services by causing large volumes of unnecessary traffic to overwhelm network bandwidth or server processes.

■ Direct Observations

Figure 2 shows the state of DDoS attacks handled by the IJ DDoS Protection Service between October 1 and December 31, 2016.

This information shows traffic anomalies judged to be attacks based on IJ DDoS Protection Service criteria. IJ also responds to other DDoS attacks, but these incidents have been excluded here due to the difficulty of accurately understanding and grasping the facts behind such attacks.

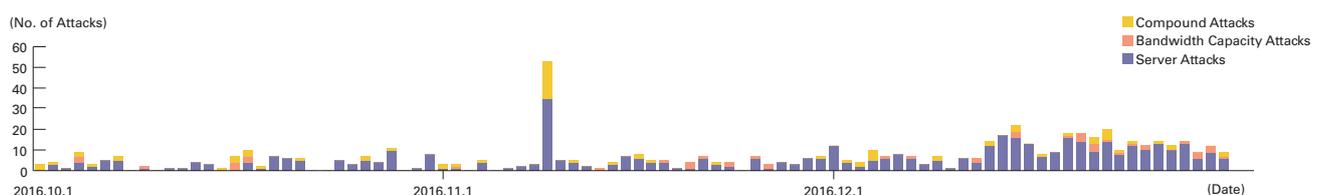


Figure 2: Trends in DDoS Attacks

*39 National center of Incident readiness and Strategy for Cybersecurity (NISC), “Organizational Bodies Designated by the Cyber Security Strategic Headquarters based on the provisions of Article 13 of the Basic Act on Cyber Security” (<http://www.nisc.go.jp/conference/cs/pdf/shiteihojin.pdf>) (in Japanese).

*40 “Important Security Information for Yahoo Users | Yahoo” (<https://yahoo.tumblr.com/post/154479236569/important-security-information-for-yahoo-users>).

There are many methods that can be used to carry out a DDoS attack, and the capacity of the environment attacked (bandwidth and server performance) will largely determine the degree of impact. Figure 2 splits DDoS attacks into three categories: attacks against bandwidth capacity^{*41}, attacks against servers^{*42}, and compound attacks (several types of attacks against a single target conducted at the same time).

During these three months, IJ dealt with 620 DDoS attacks. This averages out to 6.74 attacks per day, which is an increase in comparison to our prior report. Server attacks accounted for 78.39% of DDoS attacks, while compound attacks accounted for 13.23%, and bandwidth capacity attacks 8.39%.

The largest scale attack observed during this period was classified as a compound attack, and resulted in 15.25 Gbps of bandwidth using up to 7,610,000 pps packets. Of all attacks, 90.97% ended within 30 minutes of the start of the attack, 9.03% lasted between 30 minutes and 24 hours, and none lasting over 24 hours were observed. The longest sustained attack for this period was a compound attack that lasted for 16 hours and 53 minutes.

We observed an extremely large number of IP addresses as the attack sources, whether domestic or foreign. We believe this is due to the use of IP spoofing^{*43} and botnets^{*44} to conduct the DDoS attacks.

■ Backscatter Observations

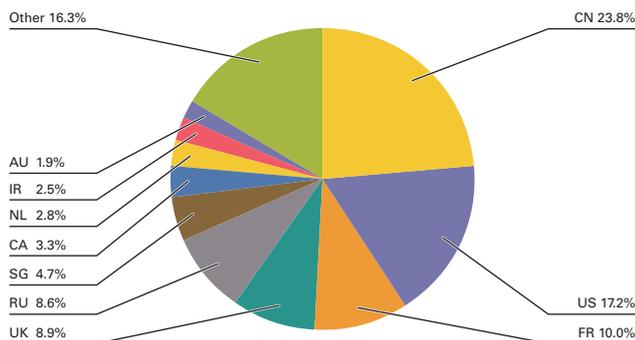


Figure 3: DDoS Attack Targets by Country According to Backscatter Observations

Next we present DDoS attack backscatter observations^{*45} through the honeypots^{*46} of the IJ malware activity observation project, MITF. Through backscatter observations, portions of DDoS attacks against external networks may be detectable as a third-party without intervening.

For the backscatter observed between October 1 and December 31, 2016, Figure 3 shows the source IP addresses classified by country, and Figure 4 shows trends in number of packets by port.

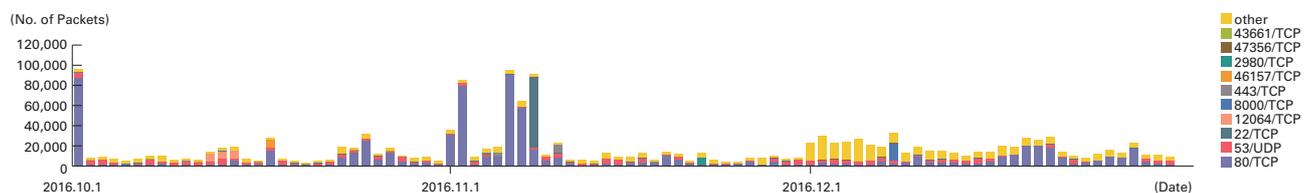


Figure 4: Observations of Backscatter Caused by DDoS Attacks (Observed Packets, Trends by Port)

*41 Attack that overwhelms the network bandwidth capacity of a target by sending massive volumes of larger-than-necessary IP packets and fragments. When UDP packets are used, it is referred to as a UDP flood, while ICMP flood is used to refer to the use of ICMP packets.

*42 TCP SYN flood, TCP connection flood, and HTTP GET flood attacks. In a TCP SYN flood attack, a large number of SYN packets that signal the start of TCP connections are sent, forcing the target to prepare for a large number of incoming connections, resulting in the waste of processing capacity and memory. TCP connection flood attacks establish a large number of actual TCP connections. In a HTTP GET flood a TCP connection with a Web server is established, and then a large number of GET requests in the HTTP protocol are sent, also resulting in a waste of processing capacity and memory.

*43 Impersonation of a source IP address. Creates and sends an attack packet that has been given an address other than the actual IP address used by the attacker to make it appear as if the attack is coming from a different person, or from a large number of individuals.

*44 A "bot" is a type of malware that after the infection, conducts an attack upon receiving a command from an external C&C server. A network made up from a large number of bots is called a botnet.

*45 The mechanism and limitations of this observation method, as well as some of the results of IJ's observations, are presented in Vol.8 of this report (http://www.ij.ad.jp/en/company/development/iir/pdf/iir_vol08_EN.pdf) under "1.4.2 Observations on Backscatter Caused by DDoS Attacks."

*46 Honeypots placed by the MITF, a malware activity observation project operated by IJ. See also "1.3.2 Malware Activities."

The port most commonly targeted by DDoS attacks observed was port 80/TCP used for Web services, and accounted for 47.6% of the total. Attacks were also observed on 53/UDP used for DNS, 22/TCP used for SSH, and 443/TCP used for HTTPS, as well as typically unused ports such as 12064/TCP, 8000/TCP, 46157/TCP, and 2980/TCP.

Looking at the source of backscatter packets by country thought to indicate IP addresses targeted by DDoS attacks in Figure 3, China accounted for the largest percentage at 23.8%. The United States and France followed at 17.2% and 10.0%, respectively.

Now we will take a look at ports targeted in attacks where a large number of backscatter packets were observed. For attacks against Web servers (80/TCP and 443/TCP), there were attacks continuing on from the previous period against the official site of a shopping district for electronics in China on October 1, and attacks against a bookmaker in the United Kingdom from November 1 through November 5. On November 6 there were attacks against a hosting provider in Russia, and from November 6 through November 7 there were attacks against a specific Russian-language site. Attacks were also observed against an online casino in Gibraltar on November 10. Regarding other ports observed to have been affected, there were attacks against 12064/TCP targeting a specific IP address in China from October 11 through October 14, attacks against 46157/TCP targeting a specific IP address in China from October 16 through October 17, and attacks against 22/TCP targeting IP addresses allocated to an ISP in Singapore on November 8. There were also attacks against 2980/TCP targeting the servers of a CDN provider in China on November 22, and attacks against 8000/TCP targeting the servers of a French hosting provider on December 8.

Notable DDoS attacks during the current survey period that were detected by IIJ's backscatter observations included attacks against multiple newspaper company sites in Belgium carried out by a group calling themselves the Syrian Cyber Army on October 24, as well as attacks against the SNS site Tumblr on December 22.

1.3.2 Malware Activities

Here, we will discuss the results of the observations of the MITF^{*47}, the malware activity observation project operated by IIJ. The MITF uses honeypots^{*48} connected to the Internet in a manner similar to general users in order to observe communications that arrive over the Internet. Most appear to be communications by malware selecting a target at random, or scans attempting to search for a target to attack.

■ Status of Random Communications

Figure 5 shows the distribution of source IP addresses by country for incoming communications to the honeypots from October 1 through December 31, 2016. Regarding the total volume (incoming packets), because the most prevalent 23/TCP, third most prevalent 1900/UDP, and fourth most prevalent 2323/TCP were significantly above other ports, they are listed separately in Figure

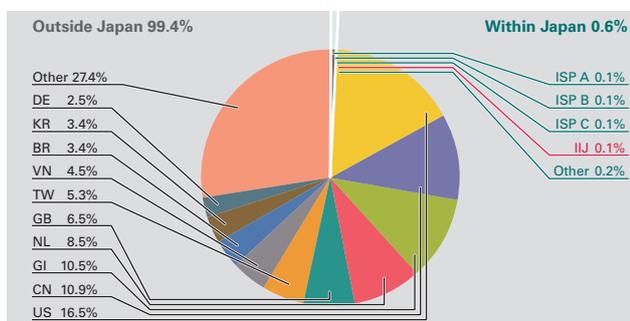


Figure 5: Sender Distribution (by Country, Entire Period under Study)

6 through Figure 8, respectively, with the remaining port trends shown in Figure 9. The MITF has set up numerous honeypots for its observations during the current survey period. Here, we have taken the average number per honeypot, and shown the trends for incoming packet types (top ten). Additionally, in these observations we made an adjustment so that multiple TCP connections to a specific port are counted as one attack, such as attacks against MSRPC.

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

*48 A system designed to record attacker and malware activities and their behavior by emulating vulnerabilities and simulating the damages caused by attacks.

Most of the communications that reached the honeypots during the survey period for this report were on 23/TCP used by Telnet, 1900/UDP used by SSDP, 21/TCP used by FTP, 22/TCP used by SSH, 8080/TCP used by Web proxies, ICMP echo requests, 1433/TCP used by SQL Server utilized on Microsoft OSes, and 3389/TCP used by Microsoft Remote Desktop.

Continuing the trend from the previous report, during the current survey period there was once again a high number of communications targeting 23/TCP used by Telnet, and we saw an increase from late October to mid-November, and another increase from mid-December onward. As reported last time, this is due to the spread of Mirai bots*49 and their variants, Bashlite, Kaiten, and hajime, which target Linux on IoT devices for bot infections. These communications were from a large number of IP

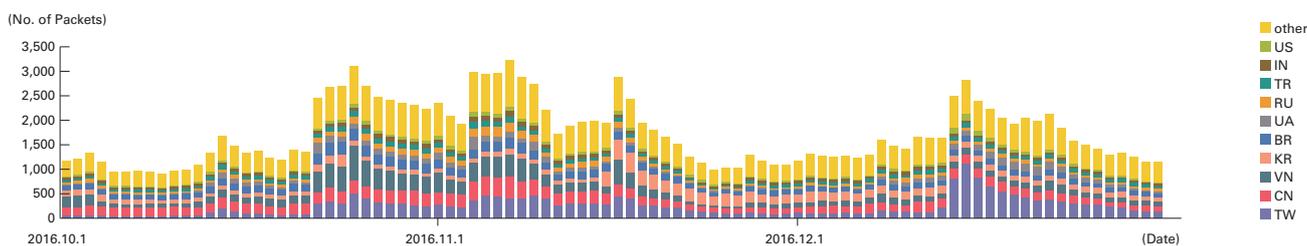


Figure 6: Incoming 23/TCP Communications at Honeypots (by Date, 23/TCP, by country, per Honeypot)

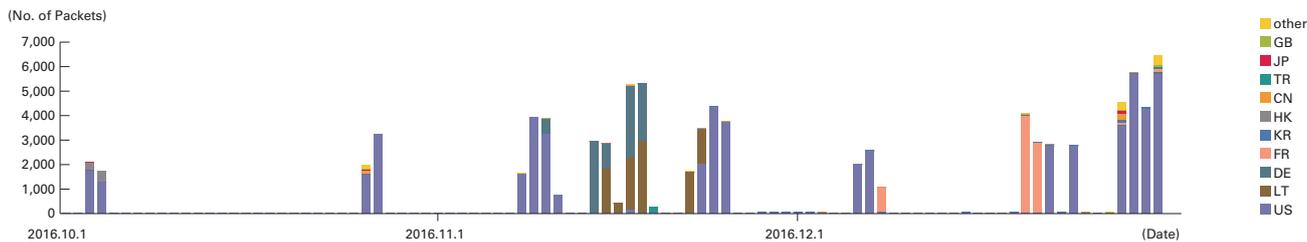


Figure 7: Incoming 1900/UDP Communications at Honeypots (by Date, 1900/UDP, by country, per Honeypot)

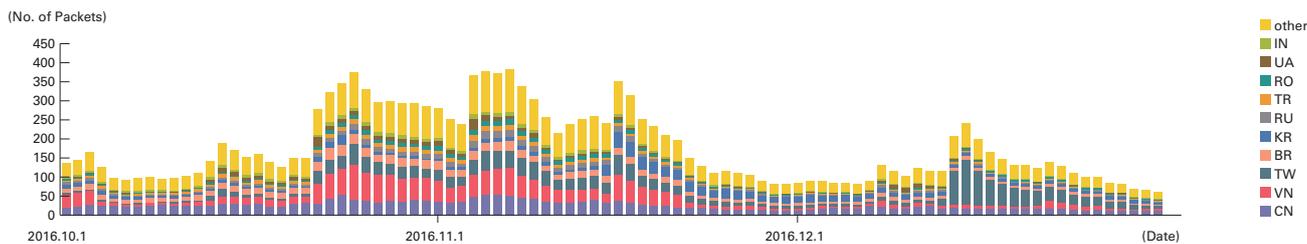


Figure 8: Incoming 2323/TCP Communications at Honeypots (by Date, 2323/TCP, by country, per Honeypot)

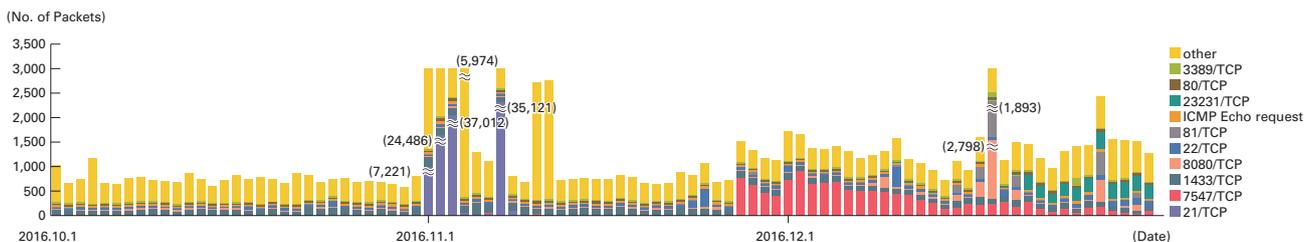


Figure 9: Incoming Communications at Honeypots (by Date, by Target Port, per Honeypot)

*49 See the previous report for more information about the Mirai botnet. "Internet Infrastructure Review (IIR) Vol.33 1.4.1 Mirai Botnet Detection and Countermeasures" (http://www.ijj.ad.jp/en/company/development/iir/pdf/iir_vol33_EN.pdf).

addresses allocated to countries such as Taiwan, China, Vietnam, South Korea, and Brazil. Also due to Mirai bots and their variants, communications to 2323/TCP, 7547/TCP, and 23231/TCP rose dramatically during this survey period.

There were sporadic increases in the 1900/UDP SSDP protocol during the current survey period. SSDP scanning requests were received from IP addresses allocated mainly to countries such as the United States, Lithuania, Germany, and France. These communications are thought to have been scanning for devices that could be used in DDoS attacks that utilize SSDP reflectors. Also, 1900/UDP communications soared to about 10 times higher than normal from the latter half of November, and this is believed to be regular scanning activity carried out by bots, etc.

■ Mirai Bot Communications

Mirai bots scan for IoT devices on the Internet before attempting infections, but we know from analysis results that one characteristic of these packets is that the TCP sequence number and the destination IP address are the same. Figure 10 shows the results of an investigation into the proportion of 23/TCP communications that match this particular characteristic. During the investigation we learned that communications matching this pattern started to appear on August 1, 2016, so we have shown communications after this date. We discovered that about 80% of 23/TCP communications were caused by Mirai bots, or malware with the same algorithm. Figure 11 shows communications matching this algorithm categorized by protocol. 2323/TCP was first observed on September 6, 80/TCP and 8080/TCP on November 2, 7547/TCP*⁵⁰ on November 26, 5555/TCP*⁵¹ on November 28, 23231/TCP*⁵² and 37777/TCP*⁵³ on December 10, 6789/TCP on December 18, 22/TCP*⁵⁴ on December 19, and 2222/TCP on December 20. This shows that development was particularly active after November, perhaps due to the release of the source code.

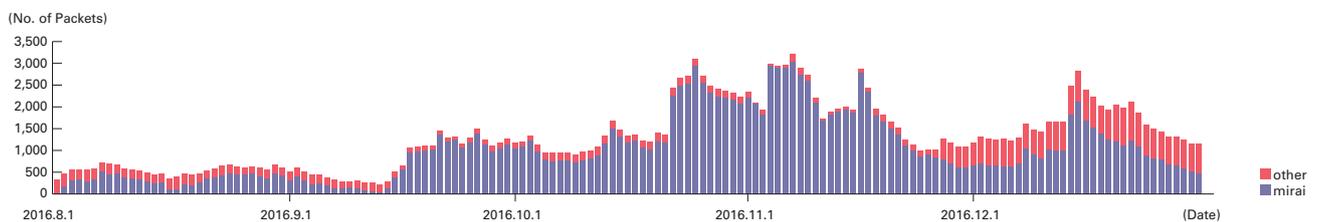


Figure 10: Incoming 23/TCP Communications at Honeybots (by Date, 23/TCP, Mirai bot ratio, per Honeybot)

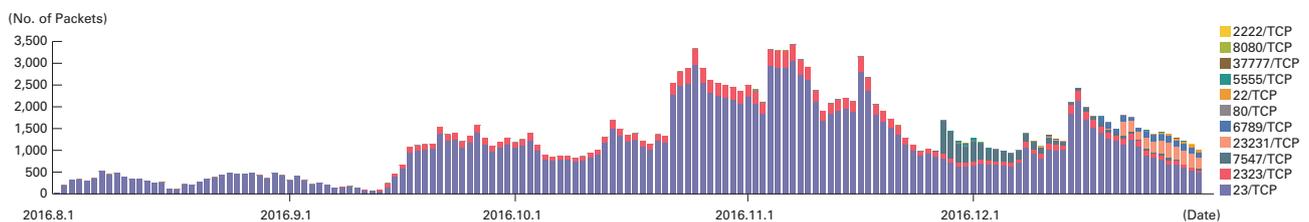


Figure 11: Incoming Communications Thought to be Mirai Bots at Honeybots (by Date, by Target Port, per Honeybot)

*50 "Port 7547 SOAP Remote Code Execution Attack Against DSL Modems" (<https://isc.sans.edu/forums/diary/Port+7547+SOAP+Remote+Code+Execution+Attack+Against+DSL+Modems/21759/>).

*51 5555/TCP scanning by Mirai bot variants is explained at the following URL. "Now Mirai Has DGA Feature Built in" (<http://blog.netlab.360.com/new-mirai-variant-with-dga/>).

*52 Mirai bot variants that scan 23231/TCP and 6789/TCP are mentioned below. "UPDATED x1: Mirai Scanning for Port 6789 Looking for New Victims / Now hitting tcp/23231" (<https://isc.sans.edu/diary/UPDATED%2Bx1%3A%2BMirai%2Bscanning%2Bfor%2Bport%2B6789%2Blooking%2Bfor%2Bnew%2Bvictims%2B%2Bnow%2Bhitting%2Btcp%2B23231/21833>).

*53 "JPCERT/CC Alert 2016-12-21: Alert on managing devices connected to the Internet - Various devices connected to the Internet are under threat -" (<https://www.jpccert.or.jp/english/at/2016/at160050.html>).

*54 The scanning of 22/TCP and 2222/TCP is discussed at the following URL. "Regarding the rapid rise in communications thought to represent the infection activity of a 'Mirai' bot variant" (<https://www.npa.go.jp/cyberpolice/detect/pdf/20170120.pdf>) (in Japanese).

■ Malware Activity in Networks

Figure 12 shows the distribution of the source where malware artifacts were acquired from during the period under study, while Figure 13 shows trends in the total number of malware artifacts acquired. Figure 14 shows trends in the number of unique artifacts. In Figure 13 and Figure 14, the trends in the number of acquired artifacts show the actual number of artifacts acquired per day*⁵⁵, while the number of unique artifacts is the number of artifact variants categorized in accordance with their hash digests*⁵⁶. Artifacts are also identified using anti-virus software, and a color-coded breakdown of the top 10 variants is shown along with the malware names. As with our previous reports, we have detected Conficker using multiple anti-virus software packages, and removed any Conficker results when totaling data.

On average, 280 artifacts were acquired per day during the period under study, while there were 21 unique artifacts per day. In Figure 13, the number of undetected artifacts rose sharply in early December, but this was due to improvements made to the MITF honeypots to enable the acquisition of malware that scans 5555/TCP and 7547/TCP. Upon investigating, we learned that artifacts acquired from these ports were Mirai bot variants.

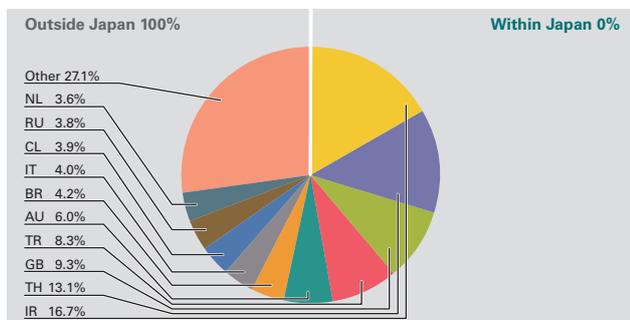


Figure 12: Distribution of Acquired Artifacts by Source (by Country, Entire Period under Study, Excluding Conficker)

After looking at other undetected artifacts more closely, included were multiple SDBOT families (a type of IRC bot) observed from IP addresses allocated to countries such as Vietnam, India, the United States, and China, as well as bitcoin mining tool downloaders.

About 97% of the undetected artifacts were in text format. This ratio is dramatically higher than the previous survey period. This is because most download locations for the Mirai bot variant have been shut down, and errors from Web servers were being output. Another factor is thought to be that the

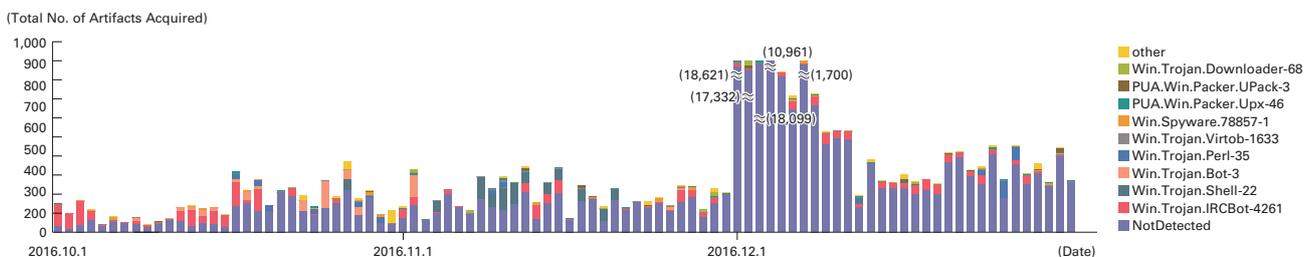


Figure 13: Trends in the Total Number of Malware Artifacts Acquired (Excluding Conficker)

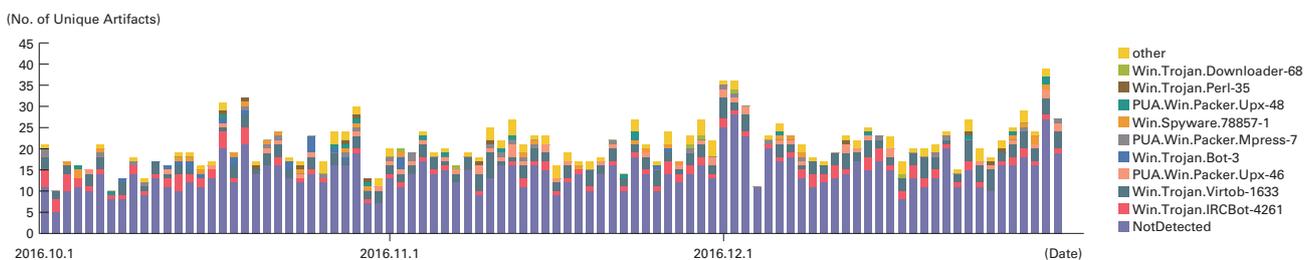


Figure 14: Trends in the Number of Unique Artifacts (Excluding Conficker)

*⁵⁵ This indicates malware acquired by honeypots.

*⁵⁶ This value is calculated by utilizing a one-way function (hash function) that outputs a fixed-length value for each input. Hash functions are designed to produce a different output for practically every different input. We cannot guarantee the uniqueness of artifacts through hash values alone, given that obfuscation and padding may result in artifacts of the same malware having different hash values. The MITF understands this limitation when using this method as a measurement index.

sites newly-infected PCs visit to download malware are already closed, despite the fact that malware such as old worms continue to infect PC's. A MITF independent analysis revealed that during the current period under observation 5.1% of malware artifacts acquired were worms, 78.4% were bots, and 16.5% were downloaders. In addition, the MITF confirmed the presence of 74 botnet C&C servers*⁵⁷ and 87 malware distribution sites.

■ Conficker Activity

Including Conficker, an average of 3,961 artifacts were acquired per day during the period under study for this report, representing 303 unique artifacts. Conficker accounted for 75.8% of the total artifacts acquired, and 93.1% of the unique artifacts. The ratio of total acquired artifacts dropped about 25% from the previous survey period, but this is because it became possible for us to acquire Mirai bot variants. Since Conficker remains the most prevalent malware by far, we have omitted it from the figures in this report. Compared to the previous survey report, the total number of artifacts acquired during this survey period decreased by approximately 5%, and the number of unique artifacts decreased by about 19%, representing a gradual overall decline. According to the observations by the Conficker Working Group*⁵⁸, as of January 2017 a total of just over 450,000 unique IP addresses are infected. This indicates a drop to about 14% of the 3.2 million PCs observed in November 2011, but shows that infections are still widespread.

1.3.3 SQL Injection Attacks

Of the different types of Web server attacks, IJ is conducting ongoing investigations on SQL injection attacks*⁵⁹. SQL injection attacks have been noted a number of times in the past, and continue to remain a major topic in Internet security. SQL injection attacks are known to attempt one of three things: the theft of data, the overloading of database servers, or the rewriting of Web content.

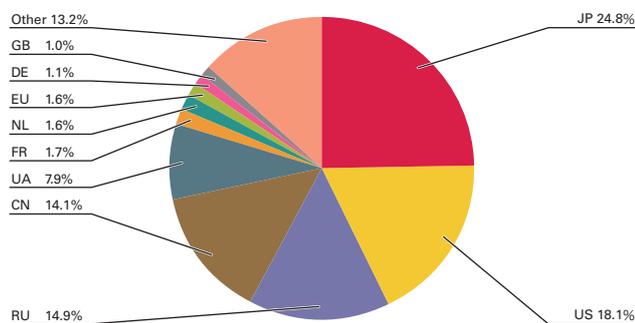


Figure 15: Distribution of SQL Injection Attacks by Source

Figure 15 shows the source distribution of SQL injection attacks against Web servers detected between October 1 and December 31, 2016. Figure 16 shows the trend in the number of attacks. These are a summary of attacks detected through signatures in the IJ Managed IPS/IDS Service. Japan was the source for 24.8% of attacks observed, while the United States and Russia accounted for 18.1% and 14.9%, respectively, with other countries following. The total number of SQL injection attacks against Web servers has increased from the levels seen in the previous report. Japan and Russia have seen particularly dramatic increases.

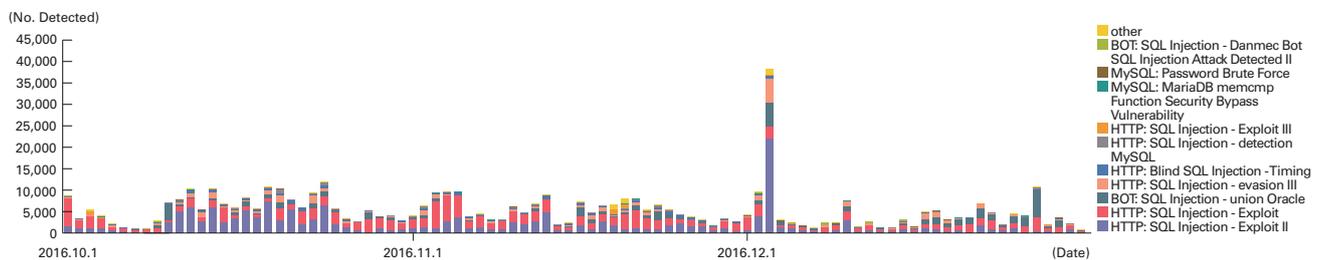


Figure 16: Trends in SQL Injection Attacks (by Day, by Attack Type)

*⁵⁷ An abbreviation of Command & Control server. A server that provides commands to a botnet consisting of a large number of bots.

*⁵⁸ Conficker Working Group Observations (<http://www.confickerworkinggroup.org/wiki/pmwiki.php/ANY/InfectionTracking>). Because no numerical data beyond January 7, 2016 is available within the current survey period, we have visually observed the highest value in the graph from early January, 2017, and used it.

*⁵⁹ Attacks accessing a Web server to send SQL commands, and operating against an underlying database. Attackers access or alter the database content without proper authorization to steal sensitive information or rewrite Web content.

During this period, attacks from a specific source in Japan directed at specific targets took place on December 3. Attacks by multiple sources in Russia against multiple targets occurred on consecutive days, leading to a significant increase in the number of attacks detected. These attacks are thought to have been attempts to discover Web server vulnerabilities.

As shown in this report, attacks of various types have been properly detected and handled within the scope of our services. However, attack attempts continue, requiring ongoing caution.

1.3.4 Website Alterations

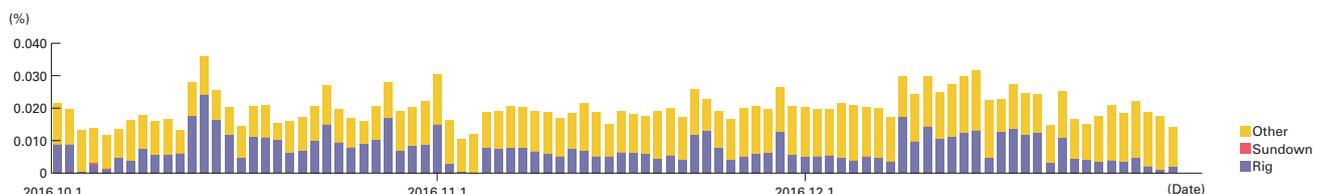
Here we indicate the status of website alterations investigated through the MITF Web crawler (client honeypot)^{*60}.

This Web crawler accesses hundreds of thousands of websites on a daily basis, focusing on well-known and popular sites in Japan. The number of sites that it accesses are added accordingly. In addition to this, we temporarily monitor websites that have seen short-term increases in access numbers. By surveying websites thought to be viewed frequently by typical users in Japan, it becomes easier to speculate on trends for fluctuations in the number of altered sites, as well as the vulnerabilities being exploited and malware being distributed.

For the period between October 1 and December 31, 2016, the Rig Exploit Kit accounted for almost all of the drive-by download attacks detected (Figure 17). This is a trend that has continued since Neutrino stopped being observed in late September, 2016^{*61}. Rig payloads such as Cerber and Ursnif have been confirmed. In early October, Sundown was also observed. It was equipped with functions for exploiting vulnerabilities in Internet Explorer, Flash, and Silverlight in confirmed cases.

We have also confirmed that when accessing websites that redirect users to these exploit kits with a macOS client, you are either not redirected to the landing page, or the landing page does not return a response. During this survey period no drive-by download attacks targeting macOS were observed^{*62}.

There continue to be cases where a fake dialog box attempting to redirect users to fraudulent sites by insinuating a malware infection is displayed in the browser. These attacks are designed to force a user to install a PUA^{*63} or call a fake support center. There have also been multiple instances in which an overseas TDS^{*64} has been used to redirect users to fraudulent sites like



*Covers several hundreds of thousands of sites in Japan.
 *In recent years, drive-by downloads using exploit kits have been configured to change attack methods and even whether or not to attack based on the client system environment or session information, source address attributes, and an attack quota such as the number of attacks. This means that results can vary wildly depending on the test environment and other circumstances.
 *Threats based on passive attacks other than exploit kits, such as direct links to fraudulent sites and executable files, are classified as Other.

Figure 17: Rate of Passive Attack Incidence When Viewing Websites (%) (by Exploit Kit)

*60 Refer to “1.4.3 Website Defacement Surveys Using Web Crawlers” in Vol.22 of this report (http://www.ijj.ad.jp/en/company/development/iir/pdf/iir_vol22_EN.pdf) for a description of Web crawler observation methods.
 *61 A quick report regarding the observation status of the Rig Exploit Kit between late September and mid-October 2016 is given under IJ-SECT’s “Alert regarding the growing number of Rig Exploit Kit observations” (<https://sect.ijj.ad.jp/d/2016/10/178746.html>) (in Japanese).
 *62 The MITF Web crawler system conducts additional surveys using a macOS client environment when a website is observed behaving in a way that indicates the possibility of a passive attack via a Windows client environment.
 *63 An abbreviation of Potentially Unwanted Application. This is a generic term for applications deemed unnecessary for general work tasks, and thought to potentially lead to unwanted results for PC users and system administrators.
 *64 An abbreviation of Traffic Distribution Systems. These are systems that buy and sell website traffic. Normally, the owner of a website redirects traffic to a TDS vendor using links, etc., and thereby receives rewards. The TDS vendor then sells this traffic, redirecting it to the highest bidder. The spread of malware that exploits TDS is discussed in Symantec’s Security Response blog under “Web-Based Malware Distribution Channels: A Look at Traffic Redistribution Systems” (<https://www.symantec.com/connect/blogs/web-based-malware-distribution-channels-look-traffic-redistribution-systems>).

these. For example, we observed that a certain European TDS was used on an ongoing basis to redirect users to multiple types of fraudulent websites from mid-September 2016 to January 2017. For business computers, it may be worth looking into blocking TDS based on the usage of the device. Also, because many TDS and similar systems use well-known cloud services as infrastructure, IP address-based control will not function effectively. A domain-based traffic controlling mechanism is required.

There are also fraudulent sites that impersonate the OS vendor or the ISP that the victim is using in the dialog boxes that are displayed. The IJ name may be displayed in the environment of customers using IJ's Internet access service, but this has nothing to do with IJ, so please beware^{*65}.

Drive-by download attacks using Rig are still continuing. We recommend implementing thorough vulnerability countermeasures such as version management for the OS, applications, and plug-ins, and implementing EMET, in environments where a browser is being used^{*66}. For website operators, it is essential to take measures against vulnerabilities by managing vulnerabilities in Web applications, frameworks, and plug-ins, as well as traffic via TDS, and also managing mashup content provided by external parties, such as advertisements and Web analytics services.

1.4 Focused Research

Incidents occurring over the Internet change in type and scope from one minute to the next. Accordingly, IJ works toward implementing countermeasures by continuing to conduct independent surveys and analyses of prevalent incidents. Here, we present information from the surveys we have conducted during this period regarding Ursnif (Gozi) anti-analysis techniques and methods for bypassing them.

1.4.1 Ursnif (Gozi) Anti-Analysis Techniques and Methods for Bypassing Them

Ursnif (also known as Gozi, Snifula, ISFB, Papras, and Dreambot) is a type of malware categorized as a banking Trojan that steals account information for financial institutions from Web browsers, and uses this information to steal money. In recent years, it has been confirmed that attackers using Ursnif are targeting the account information of financial institutions in Japan^{*67}. Multiple sources of infection have been identified, and there have been reports of Web infection via exploit kit, as well as the additional download of spam emails or URLZone (also known as Bebloh or Shiotob)^{*68}. Consequently, this malware is mentioned frequently in news related to IT security.

*65 This kind of impersonation has been carried out repeatedly, as is also discussed in IJ-SECT's "Alert regarding fraudulent sites that display ISP information and redirect visitors to fake support desks" (<https://sect.ij.ad.jp/d/2015/12/258504.html>) (in Japanese).

*66 Examples include separating administrator privileges and applying application white lists. See Vol.31 of this report (<http://www.ij.ad.jp/en/company/development/iir/031.html>) under "1.4.2 Hardening Windows Clients Against Malware Infections" for more information.

*67 Especially since 2016, there have been many cases of Ursnif infections in Japan, and various vendors and organizations have issued alerts. "Beware of falling victim to the 'Gozi' Internet banking malware" (<https://www.jc3.or.jp/topics/gozi.html>) (in Japanese). "Ursnif (a.k.a. Gozi, etc.) has been more active since March." (<http://www.lac.co.jp/blog/category/security/20160615.html>) (in Japanese). Trend Micro Security Blog, "New spread of 'URSNIF,' which targets Internet banking in Japan" (<http://blog.trendmicro.co.jp/archives/13471>) (in Japanese). Trend Micro Security Blog, "Malware spam attack continues into 2017, spread of a new 'Tuesday morning' case confirmed" (<http://blog.trendmicro.co.jp/archives/14296>) (in Japanese). "2017-01-24 - ONGOING JAPANESE MALSPAM CAMPAIGN SPREADING URSNIF VARIANT" (<http://www.malware-traffic-analysis.net/2017/01/24/index3.html>). Of these, the variant called Dreambot has been updated frequently, and in some cases it accesses a C&C server that exists on a Tor node. "Nightmare on Tor Street: Ursnif variant Dreambot adds Tor functionality" (<https://www.proofpoint.com/us/threat-insight/post/ursnif-variant-dreambot-adds-tor-functionality>). In this report, we examine cases that occurred in Japan in July 2016. JC3 has also reported that an incident of illegal bank transfer caused by Dreambot occurred in late January 2017 (<https://www.jc3.or.jp/info/malware.html#p105>) (in Japanese). The following report also details Dreambot incidents between August 2016 and September 2016 (<https://www.cyber.nj.gov/threat-profiles/trojan-variants/dreambot>).

*68 IJ has confirmed drive-by downloads via the Angler Exploit Kit, Neutrino Exploit Kit, and Rig Exploit Kit. Regarding email, as of January 2017, attachments have included ".svg" and ".js" files, when executed will download and execute Ursnif in most cases. The body text of the email is written in Japanese. In particular, despite the fact that ".svg" is an image file format, it was exploited to bypass mail gateways and sandboxes because it is described using XML, and runs JavaScript. The following report details cases where a ".svg" file was used. "Japanese emails with various subject lines and malware (viruses) attached are spreading" (<http://security-t.blog.so-net.ne.jp/2017-01-23>) (in Japanese). Also, the following reports contain information on infection via URLZone. "'Ursnif' banking malware analysis report" (https://www.nttsecurity.com/-/media/nttsecurity/files/resource-center/what-we-think/ursnif_20161215.pdf) (in Japanese). "Looking back at cyber attacks in 2016 -- 'scattershot-type' and '40 dollar all-purpose malware'" (<https://japan.zdnet.com/article/35093731/>) (in Japanese).

Ursnif uses multiple methods to interfere with or hinder attempts at analysis, and in some cases results may not be obtained correctly in an analysis environment. Therefore, in this report we will discuss the techniques that Ursnif uses for anti-analysis and look at how to bypass them to output communications with C&C servers, and gain information to effectively respond to incidents. Be sure to perform the techniques presented here in an environment that can be restored immediately after analysis is complete, such as a virtual machine, and also be sure to isolate this environment from your network.

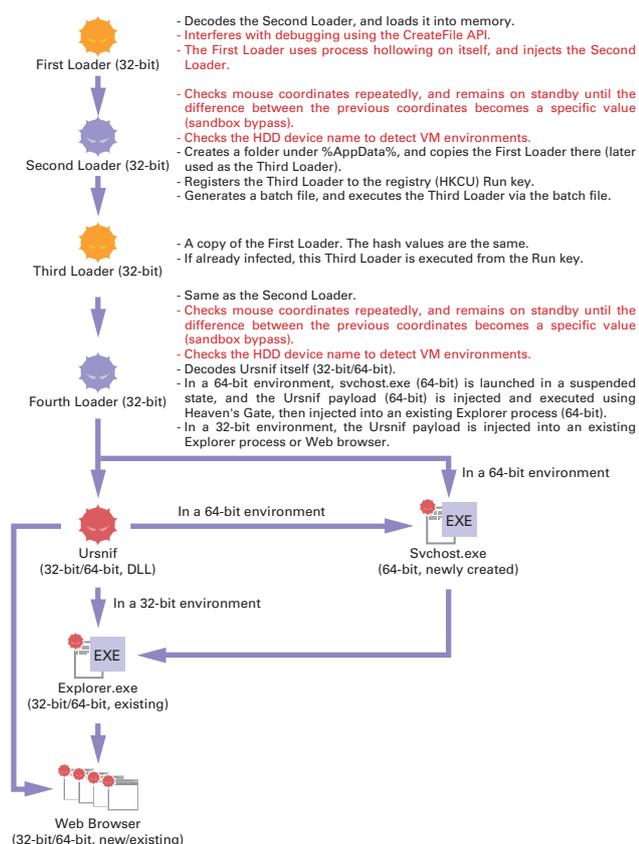


Figure 18: Ursnif Behavior Upon Infection

Overview

Figure 18 shows the behavior of Ursnif when an infection occurs. As shown in the figure, Ursnif performs multiple code injections^{*69}, and during this task, anti-analysis techniques are employed. The First Loader executable file contains the Second Loader, which is only loaded into memory, as well as the encoded Ursnif itself. The red text in Figure 18 indicate the anti-analysis techniques that we will focus on explaining here.

Anti-Debugging^{*70} with the CreateFile API

The Ursnif First Loader uses the "CreateFileA"^{*71} API to open and load itself. At this time, the API call is made with "dwShareMode" set to 0, so it attempts to read itself in exclusive mode. Meanwhile, debuggers also use the CreateFile API to open analysis targets. However, many debuggers do not close the file handle^{*72} generated when a file is opened, so when Ursnif's call to the CreateFile API attempts to read the file in exclusive mode, the file is already opened by the debugger, and the API fails since it attempts to open the same file. The results of an investigation using debuggers that I often use are shown in Table 1.

*69 Code injection is a method for inserting and executing code in other processes.

*70 Debugging generally refers to the act of discovering and eliminating software bugs, and the tool used for this is called a debugger. In malware analysis, malware is run on a debugger not to discover bugs, but to enable analysts to stop the process at the place they want to pinpoint, and inspect the associated code and CPU and memory state.

*71 In this report, we use the term "CreateFile API" in some places, and "CreateFileA" in others. There is a clear reason for this. In Windows, when a parameter or return value contains a pointer to a character string, most APIs have an "A" suffix for handling ANSI format, or a "W" suffix for handling the UNICODE format. For example, there are two CreateFile APIs: CreateFileA (ANSI) and CreateFileW (UNICODE). When the term CreateFile API is used, we are referring to both of these. When writing code, you can just use "CreateFile" and the compiler will automatically select the appropriate API according to the settings at the time of compiling, so programmers don't need to worry about this. However, strictly speaking, there is no "CreateFile" API within Windows, and only CreateFileA and CreateFileW exist, so analysts must distinguish between the two. For example, when setting breakpoints, you cannot set one on "CreateFile". You have to set one to either A or W, or both. On the other hand, the question of which is used by malware depends on the development environment of the malware creator, or simply their preference. Because this is not known when beginning analysis, I start by configuring A to see whether execution is stopped, and if not, then I try configuring W. Thus, when CreateFile API is used in this report, it refers to both CreateFileA and CreateFileW, whereas when "CreateFileA" is used in quotation marks, it refers to the exact API name. Incidentally, Windows has an API called "ZwCreateFile", which is a low-level version of the CreateFile API. CreateFileA and CreateFileW internally call "ZwCreateFile" before the execution is ultimately passed to the kernel, and some malware may use this API directly. This also varies depending on the preference of the malware creator, so analysts must also use different API names when configuring breakpoints accordingly.

*72 A file handle is a kind of permission obtained in advance from the OS when performing file operations, such as reading and writing. In the real world, many people would associate handles with the part of an object designed to be held using the hand to carry or control it, and in a similar way, file handles are designed to control (read/write to, etc.) files.

To bypass this interference when using debuggers listed as “No” in Table 1, close the file handles manually after files are loaded into the debugger. Below we present an example using x32dbg*73 (the 32-bit version of x64dbg).

1. Ursnif is loaded into x32dbg (drag and drop an Ursnif sample after launching x32dbg).
2. Launch Process Hacker*74, right-click the x32dbg.exe process (Figure 19 (1)), and then select “Properties”.
3. Click the “Handles” tab (Figure 19 (2)), select the entry that lists the name of the file being analyzed under “Name” and where the “Type” is “File” (Figure 19 (3)), then right-click it and select “Close” (Figure 19 (4)).

Figure 20 shows the state immediately after executing the CreateFile API via Ursnif when this measure is not applied (Figure 20 top), and when it is implemented (Figure 20 bottom). When the measure is implemented (Figure 20 bottom), the file handle is created correctly with an EAX value of 0x74, whereas when the measure is not implemented (Figure 20 top), EAX becomes 0xFFFFFFFF (INVALID_FILE_HANDLE). In other words, this means that the file failed to open. Thus, any subsequent processing is not carried out properly, and Ursnif attempts to read a nonexistent memory area (the area where the Second Loader would normally be stored), which leads to a crash.

Table 1: Whether or Not the File Handle of the Analysis Target is Closed

Debugger	Closed
OllyDbg 1.10 / 2.01	Yes
Immunity Debugger 1.85	Yes
x64dbg / x32dbg (Jan 27 2017)	No
WinDbg 6.2 / 6.3	No
IDA Pro 6.95 (Local Win32 Debugger)	No

■ Process Hollowing

Process hollowing is a type of code injection technique, where a new process to be injected is launched, then the code and data related to the corresponding process is deleted from memory, and the malicious code and data is injected into that memory location and executed. In recent years, malware frequently uses this or a similar injection technique.

This technique is used to slip past personal firewalls and host-based IDS solutions. For example, when injecting into a web browser such as IE, the web browser is allowed to communicate, so it is possible to access C&C servers. Also, it

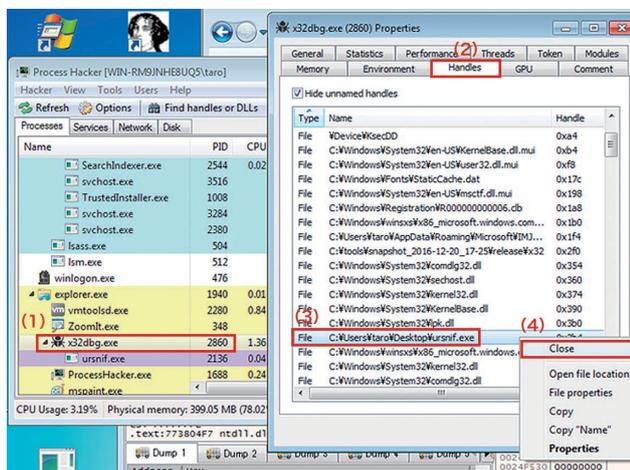


Figure 19: Closing the File Handle of the Analysis Target Within a Debugger

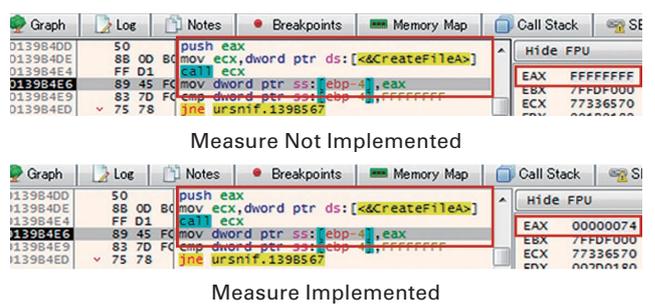


Figure 20: CreateFileA Results

*73 x32dbg is a debugger for 32-bit executables included within x64dbg. It feels very similar to OllyDbg, which was the de facto standard among analysts for a long time. Compared to OllyDbg, there are the advantages that it is possible to debug 64-bit executables (PE32+ format), and memory breakpoints on execution can be configured. Another appealing aspect is the fact that improvements are made to it every few days. It can be obtained from the following URL (<http://x64dbg.com/>).

*74 Process Hacker could be thought of as an advanced version of Task Manager, with features similar to Sysinternals Process Explorer. Compared to Process Explorer, it has a function for rewriting arbitrary memory areas, as mentioned in this report. It can be obtained from the following URL (<http://processhacker.sourceforge.net/>).

*75 EAX is one of the general-purpose registers (roughly speaking, those are like variables dedicated to the CPU) in the x86 architecture, and in Windows implementations, return values are stored there after executing a function. For CreateFile, the file handle is included in the return value, so by checking this you can identify whether or not acquisition of the file handle was successful.

looks just like a legitimate running process when examining the process list in Task Manager, preventing incident responders from recognizing the presence of malware at first glance.

A method for executing process hollowing is as follows:

1. Create a new process (in most cases, a regular Windows process such as svchost.exe or iexplore.exe^{*76}) in a suspended state (hereinafter Process B). CREATE_SUSPENDED is used as the parameter for CreateProcess API (Figure 21).
2. Delete the area related to the Process B executable using ZwUnmapViewOfSection API (Figure 21).
3. Copy the malware code and data to the Process B area (Figure 22). The ZwCreateSection and ZwMapViewOfSection APIs or VirtualAllocEx and WriteProcessMemory APIs are often used for this.
4. The Process B entry point^{*77} is overwritten with the malware. The GetThreadContext and SetThreadContext APIs are used for this.
5. The malicious code is executed at the copy destination by using the ResumeThread API to resume the execution of Process B that was originally suspended.

Some people may believe that after the CreateProcess API is executed, they can just attach^{*78} the created process using a debugger straight away. However, due to Windows specifications, it isn't actually possible for analysts to attach the corresponding process using a debugger until the execution of Process B is resumed in step 5. This is because code execution is suspended far earlier than the entry point of this executable when a process is started using CREATE_SUSPENDED. For this reason, you cannot attach the process until the OS has executed various preliminary tasks. Thus, before reaching step 5 analysts must use some method to advance the execution of the corresponding process until a debugger can attach, and then stop code execution before malicious code is executed.

In cases such as this, I use the following technique as a workaround:

1. Proceed with execution up to the SetThreadContext API (step 4 above), and confirm the address of the entry point for malicious code.
2. Rewrite the entry point for malicious code into an infinite loop.
3. Execute the ResumeThread API to begin execution of Process B.
4. Attach the infinitely looping Process B using a debugger, and restore the infinite loop to the original code to perform analysis.

Using this technique, because the process is executed up to the entry point for malicious code (code for the preliminary processes carried out by the OS, and not malicious code), and then sent into an infinite loop, the malicious code has not yet been executed. Once the entry point is reached, you can attach the process using a debugger, which meets the aforementioned requirements.

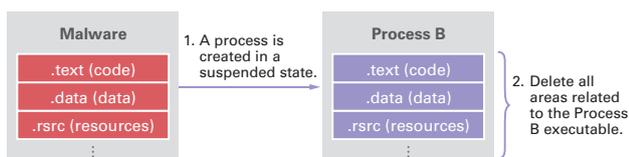


Figure 21: Process Hollowing (Starting in a Suspended State)

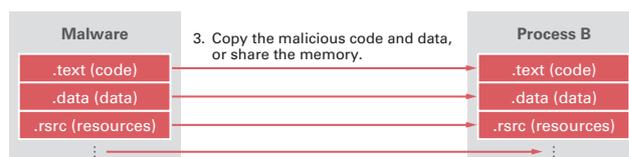


Figure 22: Process Hollowing - Continued (Copying of Malicious Code and Data)

*76 However, the Ursnif First Loader creates the First Loader itself as a child process, and conducts process hollowing on this child process. For this reason, I believe this is used as a kind of anti-debugging, rather than a way to slip past personal firewalls.

*77 The entry point is the address for the code that is executed first in an executable.

*78 To "attach" means to debug an existing process using a debugger.

As an example, let us examine how to handle process hollowing using x32dbg and Process Hacker together. For Ursnif, we will continue this discussion as if a workaround has been implemented for interference performed using the aforementioned CreateFile API.

1. Click the disassembly area at the top-left of the x32dbg screen, and then press CTRL + G.
2. Enter "CreateProcessA" into the pop-up window that appears (Figure 23 (1)), and then click "OK" (Figure 23 (2)).
3. Once you have confirmed that you have moved to CreateProcessA (Figure 23 (3)), press the F2 function key to configure the breakpoint*79. Once configured, the address part will turn red (Figure 23 (4)).
4. Use F9 to execute the code, and due to the breakpoint, code execution is stopped at the first line of code in "CreateProcessA," and the EIP*80 will point to "CreateProcessA." When stopped at the breakpoint, the address part will show as red text over a black background (Figure 24 (1))*81. Looking at the stack area (Figure 24 bottom right), "4" is contained in the part corresponding to "dwCreationFlags," the sixth parameter of the CreateProcess API (Figure 24 (2)), and this represents "CREATE_SUSPENDED." This could be considered a sign of process hollowing.

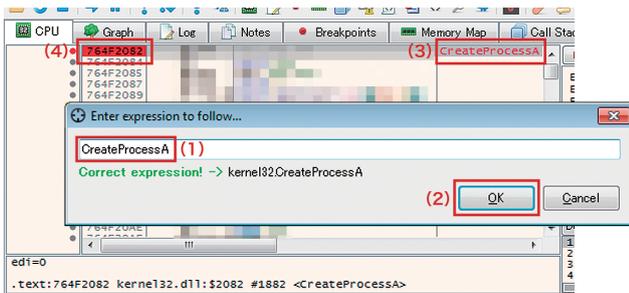


Figure 23: Setting a Breakpoint on CreateProcessA

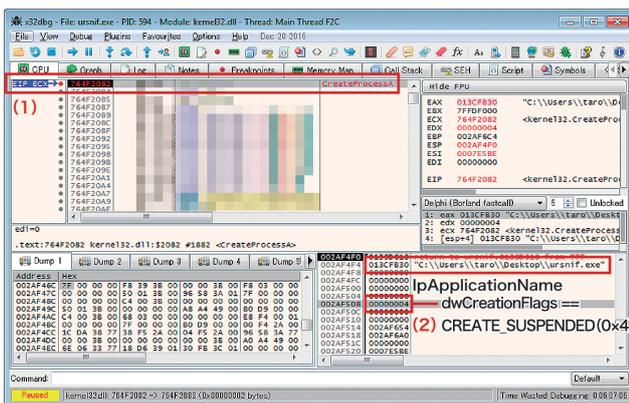


Figure 24: Confirming Ursnif was Created Using CREATE_SUSPENDED

5. Set a breakpoint in "SetThreadContext" in a similar way, and then execute it. After confirming the process paused at "SetThreadContext" (Figure 25 (1)), click on the second parameter, "lpContext" (Figure 25 (2)), and then select "Follow DWORD in Dump" from the right-click menu.

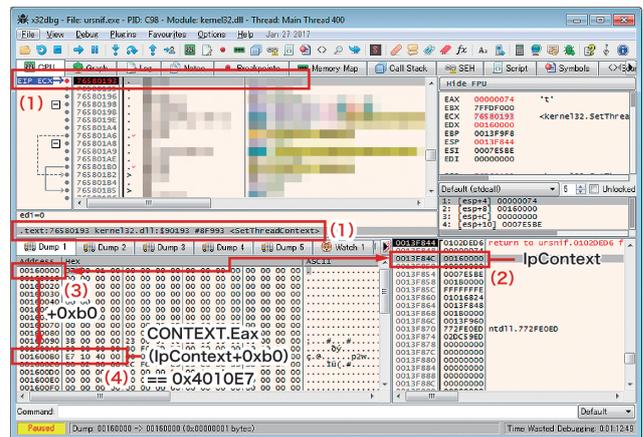


Figure 25: Execution Stopped at SetThreadContext to Confirm Status

*79 A breakpoint is a mechanism for stopping code execution for the process subject to debugging, and returning the processing back to the debugger. There are software breakpoints that change the first byte of the code you want to stop at to the int 3 (0xcc) instruction inside the debugger to trigger an exception that causes the execution to stop temporarily, as well as hardware breakpoints that use the CPU's debug register to stop a process, and memory breakpoints that will be described later. In this case, a software breakpoint is set. Each of these types of breakpoints has its advantages and disadvantages, so they are used accordingly.

*80 EIP is a dedicated register for x86 architecture that always points to the address of the code to be executed next.

*81 Looking at the MSDN CreateProcess API page ([https://msdn.microsoft.com/en-us/library/windows/desktop/ms682425\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms682425(v=vs.85).aspx)), you can see that the sixth parameter is dwCreationFlags. Meanwhile, in Figure 24 the execution is stopped at the initial address of "CreateProcessA". At this time, the top of the stack (the value pointed to by the ESP register) is the return address (which stores the address of the code for returning to the caller of CreateProcessA), and after these parameters are stored in 4-byte units (as we are performing analysis on 32-bit Windows in this case, each parameter is 4 bytes (=32-bit)). For example, in Figure 24, ESP is 0x2af4f0, the first parameter (lpApplicationName) is 0x2af4f4, and the second parameter (lpCommandLine) is 0x2af4f8. In other words, the sixth parameter can be calculated as ESP + 6*4 = 0x2af4f0 + 24 (0x18) = 0x2af508.

- Check that the address part of the "Dump 1" tab matches the address corresponding to "IpContext" in the stack (Figure 25 (3)). This address points to the beginning of the CONTEXT structure. The 4 bytes in the "+0xb0" position from here is the value set to EAX after this API is executed. When started using CREATE_SUSPENDED, the address stored in EAX will ultimately be treated as the entry point*⁸². In this artifact, we can see that this is the value "0x4010e7" (Figure 25 (4))*⁸³. Make a note of this value. By changing the instruction at this address to an infinite loop, you can stop the program before the malicious Ursnif code is executed.
- Launch Process Hacker, and double-click the Ursnif child process (Figure 26 (1)). The "Properties" window will open, so switch to the "Memory" tab (Figure 26 (2)), and find the area where the address you made note of earlier belongs to. Because the address is "0x4010e7" in this case, which is within the 396 KB space from "0x400000," double-click "0x400000" to open it (Figure 26 (3)). The "0x400000" dump window will open, click "Go to..." (Figure 26 (4)), enter the remaining "0x10e7," and then click "OK." You will jump to 0x10e7, write down those 2 bytes. Next, change them to "eb fe," click "Write," and then click "Close." This rewrites the entry point into an infinite loop ("eb fe" means an infinite loop in x86 machine language).
- Go back to x32dbg, and press F9 to execute the program through to the end. This terminates the parent process, but since the child process has an infinite loop at the entry point, you can confirm by using Process Hacker if a high ratio of CPU resources are still being consumed (Figure 27 (1)). After noting down the PID of this process, press "ALT + A" in x32dbg, and attach a process with a matching PID. When doing so, keep in mind that Process Hacker displays the PID using decimal numbers, while x32dbg displays it using hexadecimal numbers.

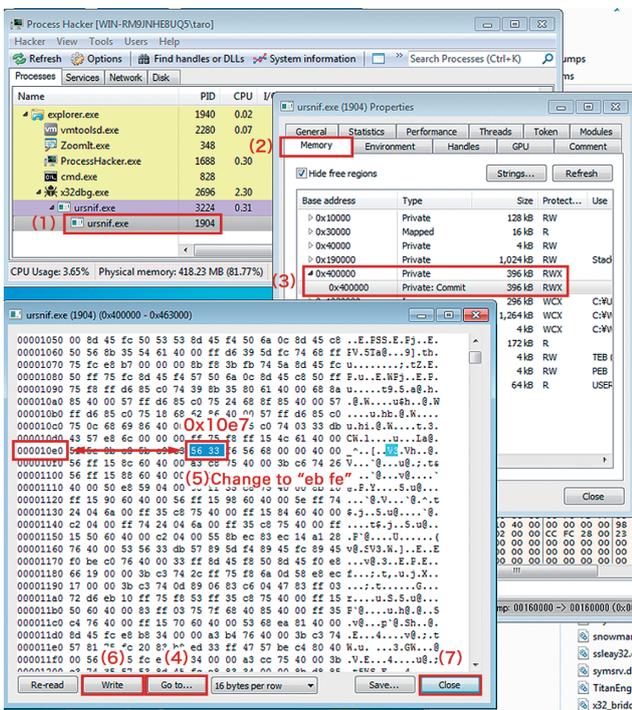


Figure 26: Changing the Entry Point to an Infinite Loop

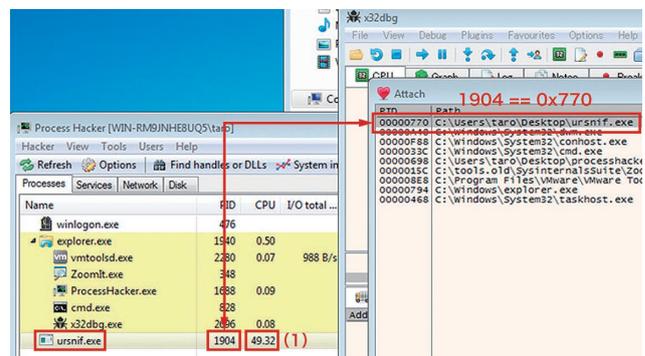


Figure 27: Attaching an Ursnif Process During the Infinite Loop

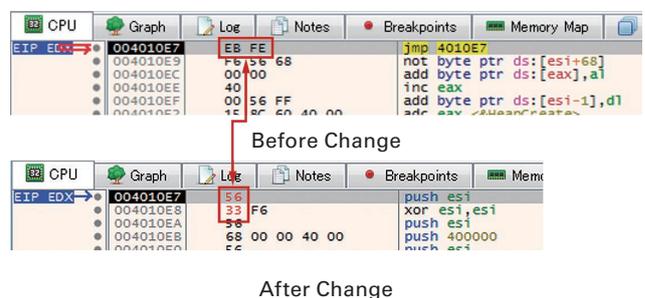


Figure 28: Restoring the Original Byte String

*⁸² Although in principle it is possible to rewrite the EIP part within the CONTEXT structure, when injecting code into a process started using CREATE_SUSPENDED, in many cases it does not run correctly even when passing the address for the malicious code to EIP directly, because the preparations for executing the corresponding process have not yet been completed. For this reason, EAX is usually used. However, because this is not always the case for injection into already running processes, it is necessary to check EIP as well. The SetThreadContext API is also often used as a set with the GetThreadContext API. When it is unclear which part the malware has rewritten, you can always check the difference to find out.

*⁸³ Since x86 architecture uses little-endian byte order, when handling this as an int format integer value, it is necessary to read it back-to-front in byte increments.

strings likely to be detected, such as product names or “virtual” with arbitrary characters (Figure 30 (4)). This enables operation of the malware without being detected by the malware itself. You can also check the character strings detected by malware by pressing F7 once to perform step execution, and return from the API to the malicious code (Figure 31).

■ Communications with C&C Servers

Once this VM detection task has completed, the Second Loader creates a folder under “%AppData%” and copies the First Loader there (hereinafter called the Third Loader) (Figure 32). The Third Loader is also registered to the registry (HKCU Run key), and configured to start automatically the next time the machine is launched. Then a batch file is created, and the Third Loader is executed. The Third Loader is the same as the First Loader, and the Fourth Loader is the same as the Second Loader. Because their behavior is identical up to a certain point, to enable communication with the C&C server, it is necessary to counter the anti-analysis techniques used by the Third Loader and Fourth Loader. So, after executing the Second Loader until it completes, load the created Third Loader using a debugger, and bypass the anti-analysis techniques once more. When this is successful, the Fourth Loader process will end. But after a few minutes communications will be sent from the malicious code injected into Explorer.exe to the C&C server, as shown in Figure 33^{*85}. Because there are clearly distinctive character strings such as “/images/” in the path, a path with a long character string, and strings ending with an image format extension such as “.gif,” “.bmp,” or “.jpeg,” one can see that detection using pattern matching should be relatively simple.

Also, in 32-bit environments, injections are carried out from the Fourth Loader to Explorer or IE, Firefox, Chrome, and Opera. The ZwCreateSection API and ZwMapViewOfSection API are used for this. By advancing the code up to this point, attaching when a new section is created in the Explorer process, and setting memory breakpoints^{*86} in that area, it is possible to further trace the behavior of malicious code injected into Explorer, including communications with C&C servers. In a 64-bit environment, the Fourth Loader (32-bit code) creates a svchost.exe (64-bit) process using CREATE_SUSPENDED, injects Urnsif itself into the 64-bit process through extensive use of the Heaven’s Gate^{*87} technique, and performs another injection into Explorer.

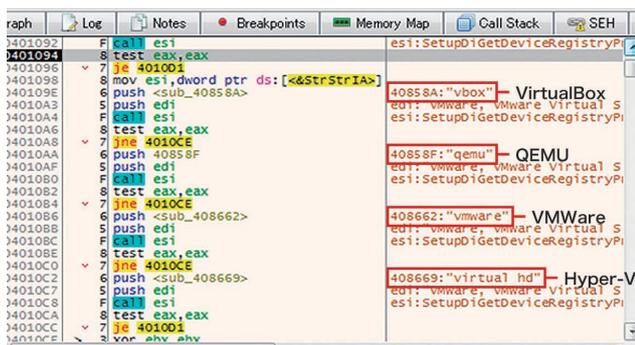


Figure 31: Detection Keywords in HDD Device Names

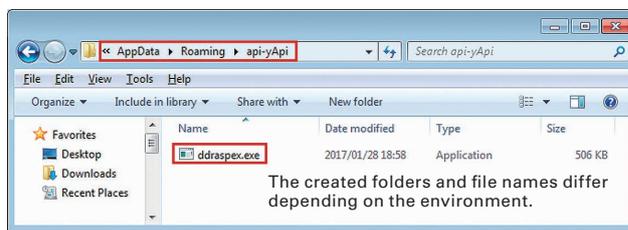


Figure 32: Urnsif Copied into a Folder Within %AppData%

*85 Figure 33 uses an Internet emulator called Fakenet-NG. You can transfer malware communications to this software and observe it even in a closed environment. It can be obtained from the following URL (<https://github.com/fireeye/flare-fakenet-ng>).

*86 A memory breakpoint is a type of breakpoint. Unlike software breakpoints that are set to the initial address of individual arbitrary instructions, and hardware breakpoints that are set in arbitrary 1 byte, 2 byte, or 4 byte units, memory breakpoints can be set to an entire arbitrary area of memory. Additionally, while software and hardware breakpoints are implemented at the architecture level, these breakpoints are implemented individually by each debugger. This type of breakpoint is implemented in the x32dbg debugger we are using this time, as well as OllyDbg.

*87 Heaven’s Gate is a technique for executing 64-bit code from 32-bit code. In normal usage situations, Windows runs in a WOW64 (Windows on Windows) environment when executing a 32-bit executable file in a 64-bit environment, with WOW64 bridging between 32-bit and 64-bit files, so users can execute them without being aware of it. However, because Urnsif injects 64-bit code from 32-bit code directly into a 64-bit process, it solves this issue by handling processes equivalent to WOW64 by itself, and this is called Heaven’s Gate. “Heaven’s Gate: 64-bit code in 32-bit file” (<http://vxheaven.org/lib/vrg02.html>). “Knockin’ on Heaven’s Gate - Dynamic Processor Mode Switching” (<http://rce.co/knockin-on-heavens-gate-dynamic-processor-mode-switching/>).

Here we have used Ursnif as an example, examining the anti-analysis techniques that this malware uses, and presented a method for bypassing them. As you can see, most anti-analysis techniques are performed through the Windows API. By investigating such techniques used by malware in advance, it should be possible to find workarounds for them, and thus be able to identify most behavior of an artifact. Also, because there are only a limited number of anti-analysis techniques, similar methods are used across a large number of malware. For example, the detection method based on checking the HDD device name that we introduced here has been confirmed in URLZone, as well as a number of adware. Gathering information in advance will enable you to cope with most situations.

We have confirmed that the methods presented here are successful when using an artifact with the following SHA-256 hash value.

```
5feeee23ecd310ed552b56c1992d5e7f6dbf4e656224a9f3073b83770768e994
```

1.5 Conclusion

This report has provided a summary of security incidents that IIJ has responded to. This time we discussed Ursnif (Gozi) anti-analysis techniques and methods for bypassing them. IIJ makes every effort to inform the public about the dangers of Internet usage by identifying and disclosing information on incidents and associated responses through reports such as this.

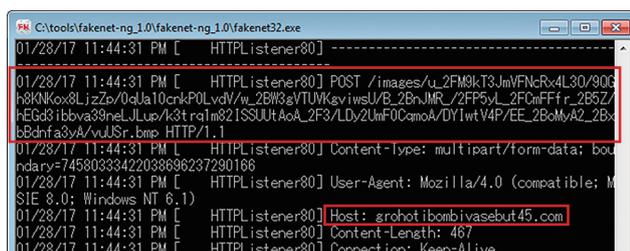


Figure 33: Communication from Ursnif to a C&C server



Authors:

Mamoru Saito

Director of the Advanced Security Division, and Manager of the Office of Emergency Response and Clearinghouse for Security Information, IIJ. After working in security services development for enterprise customers, in 2001 Mr. Saito became the representative of the IIJ Group emergency response team IIJ-SECT, which is a member team of FIRST, an international group of CSIRTs. Mr. Saito serves as a steering committee member for several industry groups, including ICT-ISAC Japan, Information Security Operation providers Group Japan, and others.

Masafumi Negishi (1.2 Incident Summary)

Tadashi Kobayashi, Tadaaki Nagao, Hiroshi Suzuki, Minoru Kobayashi, Hisao Nashiwa (1.3 Incident Survey)

Hiroshi Suzuki (1.4.1 Ursnif (Gozi) Anti-Analysis Techniques and Methods for Bypassing Them)

Office of Emergency Response and Clearinghouse for Security Information, Advanced Security Division, IIJ

Contributors:

Yuji Suga, Yasunari Momoi, Hiroyuki Hiramatsu, Office of Emergency Response and Clearinghouse for Security Information, Advanced Security Division, IIJ

The Current State of Library OSes

2.1 Introduction

Over the past few years, software called library OSes (operating systems) has been attracting a lot of attention. In this report, we examine the significance of this type of software, and discuss the software under development at the IJ Research Laboratory.

2.1.1 Why Do We Need a New OS Now?

OSes were invented as computers evolved, and their performance and convenience have been improved since then. There are now various types of OS for each usage scenario, and they serve as the heart of all software in computers. So why would we need a new or different OS when the technology has already matured to this level?

The unit of ownership for computer resources has changed from multiple people to individuals. And with the pseudo-computers made possible by virtualization technology today, each service or session can own a computer and OS instead of each user. These virtual computers require an OS with functions that are as slimmed down as possible, rather than a generic OS that has variety of functions. Some anticipated forms of use may include situations where the boot time is required to be as fast as several milliseconds, or where the OS only runs for a few seconds, and is shut down once a specific task is completed.

Additionally, for the programs used in data centers and the high-performance computing (HPC) used in scientific computation, the functions and programs run are limited in scope in the first place, so to achieve specific performance improvements they require an OS dedicated to boosting the efficacy of certain computational processes, instead of a generic OS.

To implement functions like these, the library OSes that were researched and developed in the 1990s are being considered again. Library OSes are based on mechanisms called exokernels, which are distinct from the monolithic kernels adopted by many current OSes.

As shown in Figure 1, by having the kernel that serves as the core part of the OS handle only the minimum necessary functions, and incorporating the majority of functions in the library OS portion. These functions can be added easily, and are reusable over different kernels. By keeping the kernel size to a minimum, exokernels support the design of OSes able to withstand flexible applications.

In the next section, we introduce the features of library OSes that have been proposed around the world as actual examples of software with these kinds of characteristics.

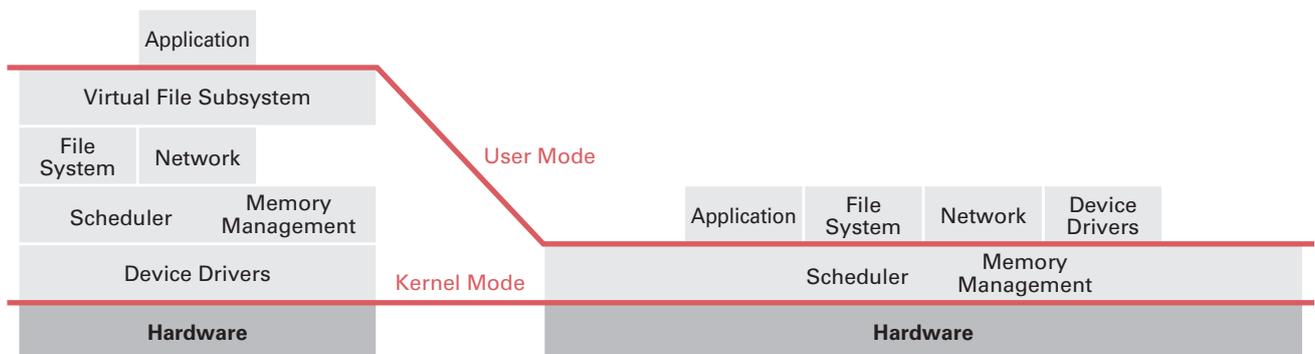


Figure 1: Comparing the Concepts of Monolithic Kernels and Exokernels*1

*1 Source: <https://commons.wikimedia.org/wiki/File:OS-structure.svg>.

2.2 Related Research Projects

2.2.1 MirageOS

MirageOS, which was presented at USENIX HotCloud in 2010, originally began research and development as an OCaml runtime that runs on a Xen hypervisor without the intervention of a regular OS. This research and development was carried out mainly by researchers at the University of Cambridge in the United Kingdom.

Their greatest contribution was introducing the concept (name) of unikernels for this scaled down software that does not depend on a specific technology such as OCaml or Xen*².

2.2.2 DrawBridge

Around the same time, a technology called DrawBridge developed by researchers at Microsoft Research was presented at ACM ASPLOS*³. DrawBridge implemented an application-specific kernel by porting over the Windows kernel source code as a library that can be directly used in user space. Their contributions are now used in the Windows virtualization function*⁴.

2.2.3 Rump Kernel

The rump kernel was proposed by Antti Kantee and others as an enhancement to the NetBSD kernel. It was designed based on a concept called anykernel. The main characteristic of anykernel is that it enables source code to be reused as a user-space library or a unikernel instead of being just a monolithic kernel.

Research and development for various other library OSes is being carried out, and technologies such as library OSes and unikernels are highly anticipated. In the next section we take a look at a library OS based on the Linux kernel, also known as the Linux Kernel Library (LKL), the development of which I am also involved in.

2.3 The Linux Kernel Library

The Linux Kernel Library (LKL) is an extension of the Linux kernel that began development from around 2007. It implements an abstraction layer atop the original Linux kernel source code that hides low-level resources such as the OS or hypervisor of the operating environment, allowing Linux-based user-space programs to link to and use it, or enabling Windows-based programs to use it as well.

2.3.1 Why Was LKL Developed?

LKL was initially developed to access files in file systems on a disk image created using another OS. Linux uses ext4 as a file system, and people can develop software that understands this file system if we need to access those files on a different OS. Although implementing this software is all that is needed to access files, this is actually quite tricky to accomplish. Because file systems are fairly sophisticated components that involve an enormous amount of source code, the cost of porting is not small. And even if it is ported, the tracking of software enhancements on a daily basis becomes even more challenging.

It is possible to run an OS on another OS through virtualization technology, but this is not efficient because resource utilization for the virtual environment is generally too large. For example, it only takes a few seconds to boot a large number of OSes if the OS size is minimal, while it can take from 10 to 20 seconds per instance when launching a general OS (such as Linux), so it is difficult to start OSes quickly*⁵.

Libraries like LKL work well in such cases. Since LKL can use the kernel source code “as-is,” there is no need to worry about the cost of porting. Quick instantiation of OSes is also possible because you can build another OS environment using a single and minimized program, without the unused code path that is required if we use a general purpose OS over virtual machines.

*2 Madhavapeddy et al., “Unikernels: Library operating systems for the cloud.” In Proceedings of ACM ASPLOS 2013, ACM, pp. 461-472.

*3 Porter et al., Rethinking the Library OS from the Top Down. In Proceedings of ACM ASPLOS 2011, ACM, pp. 291-304.

*4 Yuma Kurogome, The Front Line of Microsoft and Library OSes, IJ Lab Seminar, July 2015 (in Japanese)

*5 Kanatsu et al., Start-Up Time Comparison and Classification for Cloud Environment-Oriented Library OSes, Internet Conference 2016 (in Japanese)

This characteristic is largely due to the benefits of anykernel, which is also implemented in the aforementioned rump kernel. This new form of use, in which only some of the OS functions are utilized, enables the intended functions to be achieved without a wasteful re-implementation process. Generally, improving reusability by increasing the level of abstraction incurs some kind of penalty, leading to issues such as decreased performance or increased complexity, but with LKL these shortcomings have been ignored to give priority to abstraction. In the next section, we examine the kinds of abstraction that have been incorporated into the design.

2.3.2 Design and Implementation

LKL is implemented as a hardware architecture within the Linux kernel source code tree. This architecture was originally introduced to abstract the differences among different processors (CPUs), but LKL, although it is not a type that absorbs CPU differences, uses this mechanism to define the roles of intermediaries so that operating environments can be controlled by external programs. This enables a Linux kernel implemented to operate on certain CPU architecture to be used in user-space programs, as well as programs for other OSes.

Meanwhile, abstraction layers added in this way generally introduce overhead. One example is the packet scheduler, a function that runs on an underlying OS to control the timing of packet transmission. Because this also exists within LKL, it is possible that individual packets will be controlled multiple times. For virtualization technology like LKL, an appropriate design that takes into account such unnecessary duplicated functions is required.

As a basic requirement of this intermediary layer, it is necessary to have the three resources, clock management, program scheduling, and memory management depend on external programs. This enables environment-independent operation using a single piece of software.

The program also performs the relay of input and output (I/O) mechanisms for exchanging data with external resources on this layer, such as accessing data on a disk drive or via a network device. In Linux, a protocol called virtio provides a mechanism for using virtual devices such as these, and LKL also performs I/O operations using (or reusing) this protocol. Reuse not only eliminates the time and effort for an implementation, it also contributes to benefitting various useful functions such as acceleration and availability that utilize virtio “as-is.”

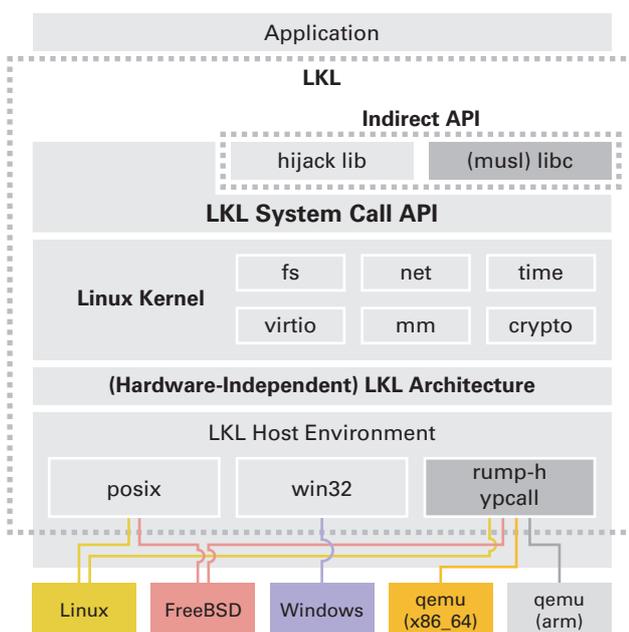


Figure 2: Linux Kernel Library Outlook

LKL is eventually built as libraries that user-space programs can link to and call. It is difficult to use the libraries as they are, so a couple of interfaces are provided as an application programming interface (API). Figure 2 illustrates an overview of the components of LKL. Abstraction layers are added at the top and bottom to enable the Linux kernel source code at the core (and the functions it implements) to be reused across a range of environments and applications. There are currently two available APIs: 1) the system call interface provided by the standard Linux kernel, known as the LKL system call, and 2) the API that uses this indirectly (Figure 2). The indirect APIs include the hijack library that uses LKL system calls by overriding functions when a program is executed, and the standard library implementation, a port of musl libc, that is linked to accordingly when compiling programs. With the latter LKL standard library, it is becoming possible to build unikernel-like applications.

2.3.3 Performance

We attempted to understand the penalties associated with abstraction and virtualization of LKL by measuring packet transmission performance with the traffic generator program called netperf. Although, strictly speaking, this is not a perfect measurement method, it can serve as an index to gain a certain insight into the performance of software.

The measurement was carried out in an environment consisting of two PCs with Intel Core i7-6700 CPUs (8-core, 3.40 GHz), connected via an Intel X540-T2 10 Gbps Ethernet card. For comparison, we used a netperf program that does not use LKL, and runs on the standard Linux kernel (version 4.9.0).

Figure 3 and Figure 4 plot the measurement results for TCP_STREAM and UDP_STREAM results, respectively. The names TCP_STREAM and UDP_STREAM indicate whether netperf uses TCP or UDP. Regarding TCP_STREAM performance, as reported in netdev 1.2*⁶, the TCP Segmentation Offload (TSO) and checksum calculation processing offload are also implemented in the virtio-net drivers. This means when the size of data to be sent by the application is large (65,535 bytes), and those offload features can be used, then packet transmission processing is carried out efficiently, limiting the drop in throughput to about 10%, and contributing to improved performance. On the other hand, in the current implementation of LKL, this offload processing is not carried out when the packet size is small, so packet transmission processing is performed via software for each packet, which is a cause of slowdown.

Meanwhile, for UDP packets where offloading does not work even in the Linux kernel, LKL still demonstrated up to 47% lower performance than the standard Linux kernel (with a packet size of 1,024 bytes). This result indicates that as additional overhead via virtualization is included when individual packets are sent, there is room for improvement. However, we observed that the transmission performance of LKL and Linux is almost identical when the application specified a large packet size. As with TCP, we believe this is because overhead was reduced by performing bulk packet transmission processing, resulting in improved throughput.

2.4 Conclusion

Although the OS is a mature technology, it is one type of software that will keep evolving, as the number of new uses keeps increasing in response to new requirements. In this report, we discussed the background behind the appearance of these new OS uses, and explained the significance of such software by giving examples of the software under research and development at the IJ Innovation Institute. We hope this helps people to rethink the role of software and operating systems with an eye toward the future development of the Internet.

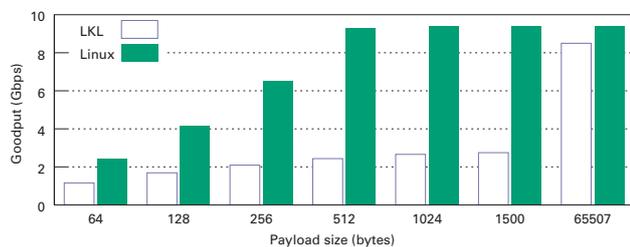


Figure 3: TCP Packet Transfer Performance When Changing Packet Size (Goodput)

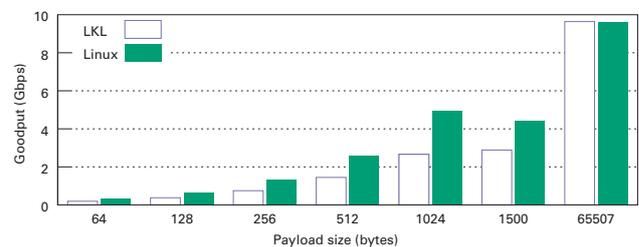


Figure 4: UDP Packet Transfer Performance When Changing Packet Size (Goodput)



Author:
Hajime Tazaki
 Senior Researcher, Research Laboratory, IJ Innovation Institute Inc.

*6 Chu et al., User Space TCP - Getting LKL Ready for the Prime Time. Linux Netdev 1.2 (Oct. 2016)



Internet Initiative Japan

About Internet Initiative Japan Inc. (IIJ)

IIJ was established in 1992, mainly by a group of engineers who had been involved in research and development activities related to the Internet, under the concept of promoting the widespread use of the Internet in Japan.

IIJ currently operates one of the largest Internet backbones in Japan, manages Internet infrastructures, and provides comprehensive high-quality system environments (including Internet access, systems integration, and outsourcing services, etc.) to high-end business users including the government and other public offices and financial institutions.

In addition, IIJ actively shares knowledge accumulated through service development and Internet backbone operation, and is making efforts to expand the Internet used as a social infrastructure.

The copyright of this document remains in Internet Initiative Japan Inc. ("IIJ") and the document is protected under the Copyright Law of Japan and treaty provisions. You are prohibited to reproduce, modify, or make the public transmission of or otherwise whole or a part of this document without IIJ's prior written permission. Although the content of this document is paid careful attention to, IIJ does not warrant the accuracy and usefulness of the information in this document.

©Internet Initiative Japan Inc. All rights reserved.
IIJ-MKTG020-0032

Internet Initiative Japan Inc.

Address: Iidabashi Grand Bloom, 2-10-2 Fujimi, Chiyoda-ku,
Tokyo 102-0071, Japan
Email: info@ij.ad.jp URL: <http://www.ij.ad.jp/en/>