Electronic Preprint for Journal of Information Processing Vol.21 No.2

Regular Paper

A Malicious Bot Capturing System using a Beneficial Bot and Wiki

Takashi Yamanoue[1,a)]   Kentaro Oda[1]   Koichi Shimozono[1]

Received: June 29, 2012, Accepted: December 7, 2012

Abstract: Locating malicious bots in a large network is problematic because the internal firewalls and network address translation (NAT) routers of the network unintentionally contribute to hiding the bots' host address and malicious packets. However, eliminating firewalls and NAT routers merely for locating bots is generally not acceptable. In the present paper, we propose an easy to deploy, easy to manage network security control system for locating a malicious host behind internal secure gateways. The proposed network security control system consists of a remote security device and a command server. The remote security device is installed as a transparent link (implemented as an L2 switch), between the subnet and its gateway in order to detect a host that has been compromised by a malicious bot in a target subnet, while minimizing the impact of deployment. The security device is controlled remotely by 'polling' the command server in order to eliminate the NAT traversal problem and to be firewall friendly. Since the remote security device exists in transparent, remotely controlled, robust security gateways, we regard this device as a beneficial bot. We adopt a web server with wiki software as the command server in order to take advantage of its power of customization, ease of use, and ease of deployment of the server.

Keywords: network security, security monitor, security control, bot, wiki, Java, API

1. Introduction

A bot is a software application that runs automated tasks over the Internet. Bots are usually malicious applications that are controlled by a malicious master herder. A number of recent viruses are used for recruiting hosts into a botnet, which is a collection of malicious bots. Once malicious bots have intruded into a campus LAN, for example, important information, such as private student data and research secrets, may be leaked. Furthermore, the bots may spam other people and attack other web sites via distributed denial of service (DDoS). A campus with malicious bots may be considered to be engaging in criminal activity. The manager of the campus LAN must be careful about malicious bots and remove bots quickly when found.

Firewalls and network address translation (NAT) are powerful tools for enhancing the network security of a LAN. These tools may defend the LAN against the intrusion of a malicious bot. A LAN protected by these tools is like a house protected by a door with a key. Only permitted IP packets may pass through the firewall or the NAT, similar to how only people who have a key may pass through the door of the house. However, when a host in the LAN is compromised by a malicious bot, it is difficult to identify the compromised host from outside the LAN. This is similar to the difficulty involved in finding a thief who is hiding in a house or building. The dynamic host configuration protocol (DHCP) and IPv6 with privacy address extension (RFC 3041) also make it difficult to identify a compromised host because the IP address of a suspicious host that is using them is changed dynamically.

A campus's LAN usually consists of a central network infrastructure and sub-LANs. Some sub-LANs may be protected by a firewall or a NAT. Network managers must sometimes find bots that are hidden in such protected sub-LANs. One way to achieve this is to prohibit the use of a firewall or NAT in sub-LANs. Defining the rule is easy but unrealistic because broadband routers with firewalls or NATs are extremely common.

When malicious communication between a bot in a protected sub-LAN and another host outside the sub-LAN is discovered by the manager of the central network infrastructure (or the central manager), the central manager usually directs the manager of the sub-LAN (or the sub-manager) to disconnect the sub-LAN from the central network infrastructure immediately. The sub-manager inspects all of the PCs in the sub-LAN using anti-virus software. However, this process cannot always find the bot because recent malwares use obfuscation and encryption of their codes, anti-virus cannot detect such malwares, which is a computer threat that tries to exploit computer application vulnerabilities that are *unknown* to others or the software developer, and the central manager cannot observe the malicious communication.

Sometimes, the central manager would like to monitor sub-LANs which are protected by users' NATs in order to find a compromised host, which should be found as quickly as possible. The central manager can monitor the sub-LAN by reconfiguring the LAN (for example, by connecting the sub-LAN directly to the central network infrastructure). However, done carelessly, such reconfiguration may make a loop in the core switch and it may put the whole network down. In order to prevent such a problem, it usually needs validation by several persons. The manager should have an easy and fast method by which to monitor and

[1] Kagoshima University, Kagoshima 890–0065, Japan
[a)] yamanoue@cc.kagoshima-u.ac.jp

control sub-LANs.

We are developing a network security control system that uses a remote security device and a web site with wiki software. The remote security device can be deployed quickly and easily because it is portable. The central manager can easily monitor and control the sub-LAN behind a firewall or a NAT from a web site using common wiki software and the remote security device. The remote security device is a type of bot that is controlled by the central manager and can perform the following tasks:

- Monitoring traffic between hosts in the sub-LAN and outside hosts.
- Filtering out malicious packets from the traffic.
- Intercepting DNS query packets from the suspicious host and returning the IP address of the fake host, which is pretending to be the herder's host.
- Pretending to be the herder's host, for example by returning a fake syn-ack packet in response to a syn packet from the suspicious host.
- Notifying the user of the suspicious host about the infection by forwarding http packets from the host to the "notifying web server."

The remote security device is connected to a sub-LAN that may have a malicious bot between the switch of the sub-LAN and its NAT. The device is controlled remotely, and communication in the sub-LAN can be monitored by the central manager using wiki software.

The remote security device is controlled by command lines on a wiki page and execution results of the command lines are written on the same wiki page. The wiki site of the wiki page can be used as a knowledge database of machine generated incident logs and similar malicious traffic patterns can be linked together. The commands history on the web site can also be seen as a knowledge database of how the central manager copes with malicious hosts. Wiki equipped with functions to share such knowledge database with people.

This system assumes that the portable security device can communicate with the wiki site. Traffic of all hosts in the sub-LAN, except the NAT, which communicate with out-side hosts, pass through the portable security device. There must be no router or no NAT between the portable security device and monitored hosts in the sub-LAN.

The present paper shows an improved version of the proposed system, which is discussed in a previous paper [14]. This paper discusses the implementation and usage of the security controlling system as well as related research. The remainder of the present paper is organized as follows. Section 2 describes related research. Section 3 presents a summary of the monitoring system. Section 4 presents a usage example. Finally, Section 5 presents a summary and describes areas for future research.

2. Related Research

This section describes the differences between the proposed system and other systems or methods.

2.1 Prohibiting the Use of a NAT

Network managers usually identify a bot-infected host by the MAC address of its network interface card. However, there are many users who cannot obtain their PC's MAC addresses. Moreover, some sub-managers cannot obtain their PCs' MAC addresses of the sub-LAN. In order to cope with such situations, Hiroshima University provides an easy way to register the MAC address of a user's host to the PC authentication gateway and the monitoring system of the central network infrastructure of the university and prohibits the use a NAT [4]. In this way, when a host is infected by a bot, the central manager notifies the user of the host directory of the infection and disconnects the host logically at the manager's office. As mentioned in Section 1, we believe that, at present, prohibiting the use of a NAT in a university would be difficult. The proposed method provides another solution by which to directly notify the user of the bot-infected host of the infection, as mentioned in Section 5.5.

2.2 Security Monitoring System

We have developed a security monitoring system using a remote sensor device and wiki software previously [13]. The previous monitoring system can also monitor the traffic of the sub-LAN behind a NAT but cannot control the traffic. The proposed security controlling system is developed by extending the security monitoring system.

2.3 Traffic Anomaly Detection Using Software Defined Networking

Mehdi and et al. showed a way to detect traffic anomaly using software defined networking [6]. Their way is similar to ours in a sense that both of their way and our way using the proposed system detect traffic anomaly at end users' side of a network. Their paper did not show the way to report the anomaly to the central network manager. On the other hand, the way to report the anomaly to the central manager using the wiki site is shown in this paper.

2.4 Snort

Snort [17] is a common open source IDS, and has a function for automatically updating signatures. The manager can view the results of Snort on a web site using ACID and has the option to receive remote Snort alerts by e-mail. It is not possible to change the settings of Snort at the NAT-protected sub-LAN from the outside. Unlike Snort, the proposed system can control the traffic.

2.5 Observing MAC Addresses at the WAN Side

Yamai et al. demonstrated a technique by which to observe MAC addresses of hosts in a NAT-protected sub-LAN from the outside [8]. This technique is effective when replacing the NAT with a new, less expensive device. In addition, with their system, there is no need to add new monitoring infrastructure, such as a web server. Their system can effectively make use of the existing monitoring infrastructure. In contrast, the proposed system requires a new web server for controlling the sensor device and monitoring the sub-LAN. However, the proposed system does not require replacement of the NAT and can observe the MAC address, even if another NAT is placed between the sub-LAN and the monitoring location.

2.6 Unix Device with Two NICs

Ishida et al. presented a method by which to manage devices (target devices) with a networking function without SNMP [3]. Their technique uses a Unix device with two NICs. A target device is connected to one of the NICs of the Unix device, and another NIC of the Unix device is used to manage the target device. The Unix device replies with management messages delegating the target device. Other messages for the target device pass through the Unix device to the target device. This technique can also use the existing management infrastructure effectively. Both their devices and the proposed sensor device use a Unix machine with two NICs. Their technique is used for management, and the proposed system is used for monitoring, even though both share a similar mechanism. A combination of these devices would be a more effective tool.

2.7 KASEYA and UNIFAS

The PC management system of KASEYA [16] and the wi-fi access point management system of Furuno Systems (UNIFAS) [18] consist of agent programs at the devices, such as PCs or wi-fi access points, and a web site to manage them, as in the proposed security monitoring system. Their devices can also communicate with the web site over a NAT. However, their devices use a specialized web server, whereas the proposed monitoring system uses a web site with common wiki software. Using common wiki, instead of using specialized web software for specific devices such as KASEYA and UNIFAS, is a better way for collecting and sharing knowledge of security expertise.

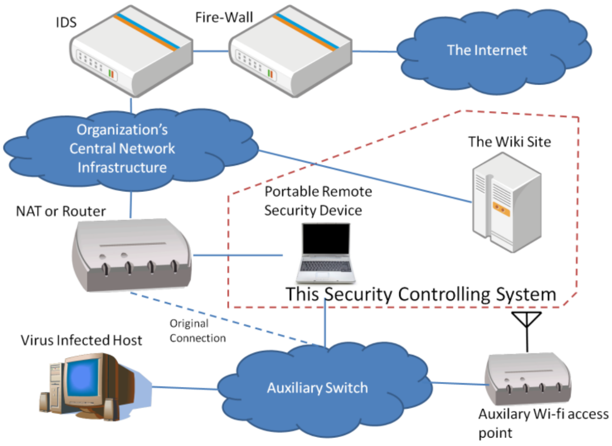

IDS
Fire-Wall
The Internet
Organization's Central Network Infrastructure
The Wiki Site
Portable Remote Security Device
NAT or Router
This Security Controlling System
Original Connection
Virus Infected Host
Auxiliary Switch
Auxilary Wi-fi access point

Fig. 1 Outline of the system.

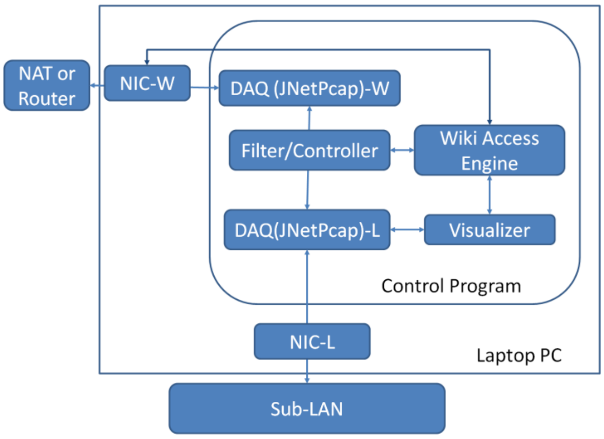


Fig. 2 Structure of the remote security device.

3. Implementation

The security control system consists of a portable remote security device and a web site with Wiki (PukiWiki) software. The security device consists of a laptop computer and an auxiliary network interface. The sensor device is controlled by commands that are written on the wiki page of the site. The results of command execution are written on the same page. The security device is connected between a sub-LAN-side port of a NAT (or router) and the switch of the sub-LAN (**Fig. 1**).

An auxiliary switch is used if PCs are connected to LAN ports of the NAT (or router) directly. An auxiliary wi-fi access point is also connected to the auxiliary switch if it is required. The web site with wiki software is connected to the network such that both the sensor device and the web site are accessible.

Figure 2 shows the structure of the remote security device. The hardware of the device consists of a laptop PC with two network interface cards (NICs), which is realized by adding an auxiliary NIC to the PC. One NIC is the WAN-side NIC (NIC-W), which is connected to the NAT or the router. The other NIC is the LAN-side NIC (NIC-L), which is connected to the sub-LAN. The control program has two data acquisition libraries (DAQs) and a filter/controller. The two DAQs are connected by the filter/controller, which monitors and controls traffic between them. Each of the DAQs is connected to one of the NICs. All communication between hosts (except the NAT) in the sub-LAN and hosts outside the sub-LAN passes through the filter/controller. The communication can be observed and controlled by the filter/controller. The filter/controller is controlled by commands from the wiki access engine and performs the following operations for each communication packet:

- If the packet matches a “select pattern,” it is passed from one DAQ to the other, and the packet frame information is sent to the wiki access engine together with the status.
- If the packet matches a “drop pattern,” it is not passed, and the packet frame information is sent to the wiki access engine together with the status.
- If the packet matches a “forward pattern,” the destination IP address and port are replaced with those of a mimicking server on a different host and the replaced packet is passed to the other DAQ. We show the details of the mimicking server in the section five. The original packet frame information is sent to the wiki access engine together with the status.
- The filter/controller also sends a packet to one of the DAQs. The sending packet is one of the following:
  - ▷ A mimicking syn-ack packet in response to a syn packet of dropped packets.
  - ▷ A mimicking DNS answer packet in response to a DNS query packet.

We show the details of the mimicking syn-ack packet and mimicking DNS answer packet in the section five.

The wiki access engine sends frame information, which includes the packet that is selected, dropped, or forwarded to the wiki page of the web site. This means that the central manager

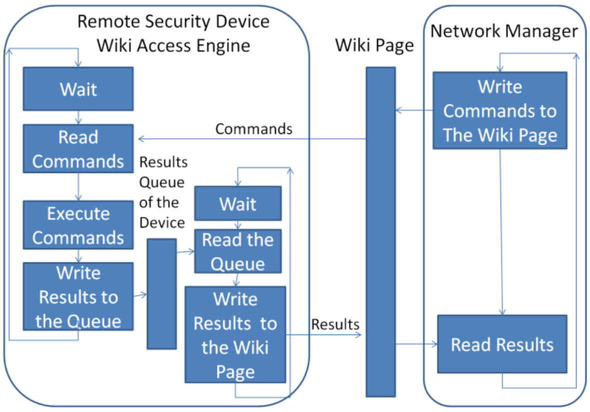


Fig. 3 Control flow outline of the proposed system.

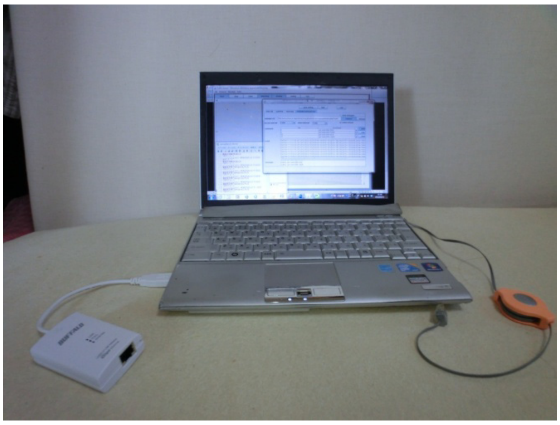
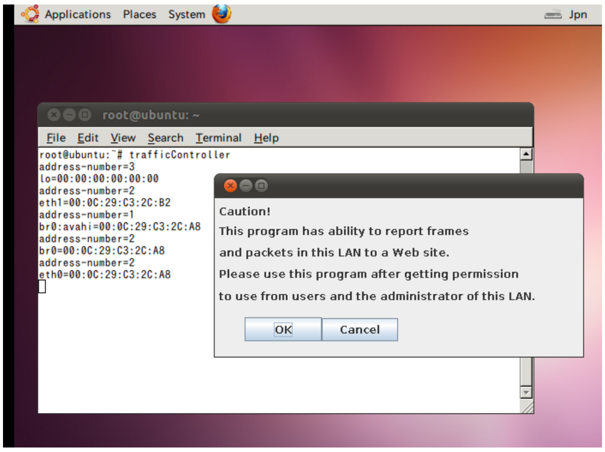


Fig. 4 Photograph of the remote security device.



```
root@ubuntu:~# trafficController
address-number=3
lo=00:00:00:00:00:00
address-number=2
eth1=00:0C:29:C3:2C:B2
address-number=1
br0:avahi=00:0C:29:C3:2C:A8
address-number=2
br0=00:0C:29:C3:2C:A8
address-number=2
eth0=00:0C:29:C3:2C:A8
```

Caution!
This program has ability to report frames
and packets in this LAN to a Web site.
Please use this program after getting permission
to use from users and the administrator of this LAN.

Fig. 5 Notification window after booting the software.

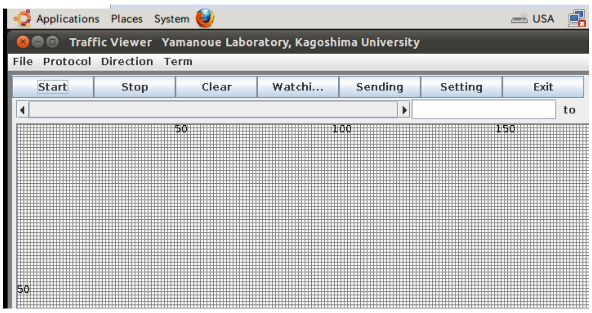
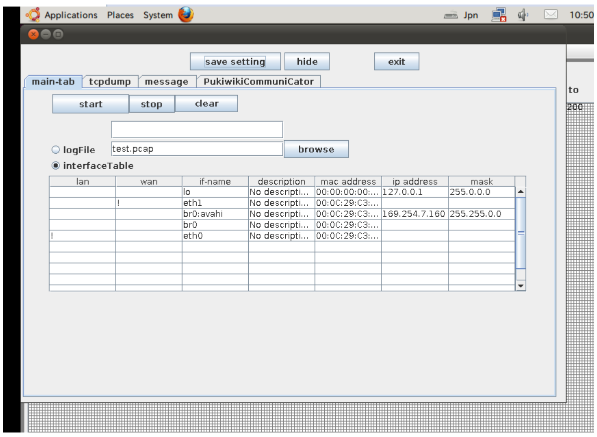


Fig. 6 Traffic viewer.



○ logFile test.pcap

● interfaceTable

lan	wan	if-name	description	mac address	ip address	mask
		lo	No descripti...	00:00:00:00:...	127.0.0.1	255.0.0.0
	!	eth1	No descripti...	00:0C:29:C3:...		
		br0:avahi	No descripti...	00:0C:29:C3:...	169.254.7.160	255.255.0.0
		br0	No descripti...	00:0C:29:C3:...		
!		eth0	No descripti...	00:0C:29:C3:...		

Fig. 7 Network interface settings page.

can obtain the MAC address of the suspicious host and identify the host from outside the sub-LAN. The central manager can also control the traffic of the suspicious host directory from the wiki.

All observed communication at the DAQ of the sub-LAN-side is visualized by the "visualizer" [7], [12].
jNetPcap [15] is used as the DAQ and is capable of not only capturing frames but also generating and sending frames. Commands for the filter/controller are set by the wiki access engine by executing commands that are written on the wiki page. The wiki access engine is implemented by converting the "Pukiwiki-Java Connector" [9], [10], [11].

Figure 3 shows the control flows of the wiki access, and **Fig. 4** shows a photograph of the sensor device.

4. Usage Example

This section presents a usage example.

4.1 Booting and Setting

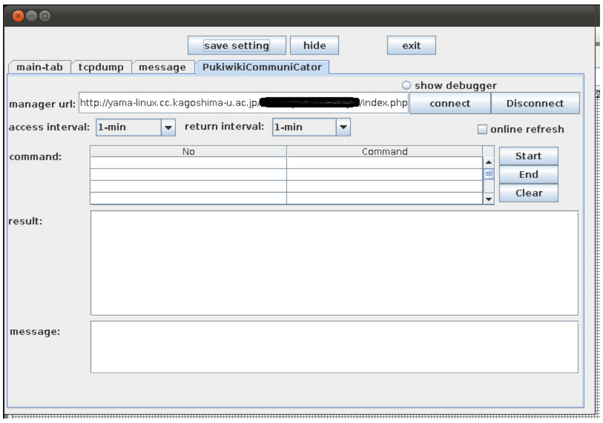
After the security device has been connected to the sub-LAN, the sensor device program is booted by executing the "trafficController" command in the Linux virtual machine of the sensor device. The window shown in **Fig. 5** then appears.

After clicking the "OK" button in Fig. 5, the "Traffic Viewer" window (**Fig. 6**) is shown. The settings window (**Fig. 7**) is shown when the "Settings" button of the window in Fig. 6 is clicked. The user of this system chooses the network interface for the DAQ in the "main-tab" page of the window. **Figure 8** shows the settings page of the "PukiwikiCommunicator" tab of the settings window.

The url of the wiki page of the web site is written in the text field at the right-hand side of the "manager url:" label. The access interval and the return interval are also set on this page. The settings are saved when the "Save settings" button is clicked. The settings window is hidden when the "hide" button is clicked.

4.2 Monitoring and Control

Communication frames are acquired from the DAQ after the "Start" button in Fig. 6 is clicked. Accessing the wiki page will begin after the "Sending" button in Fig. 6 is clicked. If authentication is required to access the page, the authentication dialog in **Fig. 9** is shown. Reading and executing commands, and writing results on the wiki page are started when the "Watching" button in Fig. 6 is clicked.



manager url: http://yama-linux.cc.kagoshima-u.ac.jp/[redacted]/index.php

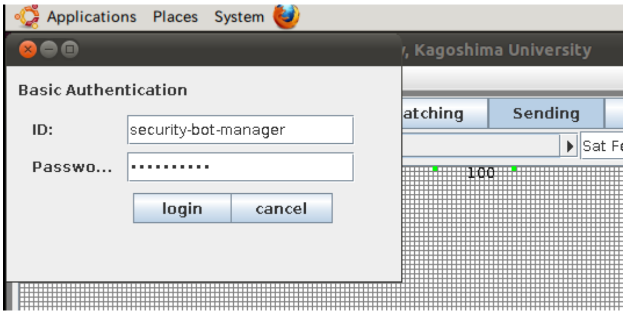
access interval: 1-min return interval: 1-min

command:

result:

message:

Fig. 8 Wiki page settings.



Basic Authentication

ID: security-bot-manager

Passwo... ••••••••••

Fig. 9 Authentification dialog.

When commands are read normally and information on filtered frames is acquired normally, the commands and results shown in **Fig. 10** are displayed at the sensor device.

4.3 Commands and Results on the Wiki Page

We show commands for controlling the remote security device in this sub section. Non-terminal symbols in a command are defined as follows.

<Address> ::= ip=<IP address>
|mac=<MAC address>
<Destination-Address> ::= dip=<IP address>
|dmac=<MAC address>
<Source-Address>::= sip=<IP address>
|smac=<MAC address>

<IP address> is substituted by an IP address and <MAC address> is substituted by an MAC address. “ip=<IP address>” means the IP address of the source or the destination address. “mac=<MAC address>” means the MAC address of the source or the destination address. As this way, “dip=<IP address>” and “dmac=<MAC address>” means the destination IP address and the destination MAC address, “sip=<IP address>” and “smac=<MAC address>” menas the source IP address and source MAC address. In order to discriminate non-terminal symbols, sub-scrip is used.

The followings are the commands.

get <Address>

This command selects an IP packet, which has the <Address> of the source address or the destination address, in the packets that are captured. One of the two DAQs passes the packet to the other and writes the packet frame information to the wiki page.

get startsWith <String constant>

This command selects an IP packet, the payload of which starts with the <String constant>, in the packets that are captured. One of the two DAQs passes the packet to the other and writes the packet frame information to the wiki page. For example, if “PING,” “PONG,” “NIC,” and “USER” are replaced with the <String constant>, the communication (which may be IRC) can be detected. The <String constant> is compared with the heading of the payload of a TCP packet only. It is not perfect but it is enough to detect some commands of IRC, HTTP, POP3 and others, and it does not consume so much time.

lan2wan drop <Address>

This command drops a packet from the LAN side if the source or the destination address matches and writes the packet frame information to the wiki page. The packet is not forwarded to the WAN side.

wan2lan drop <Address>

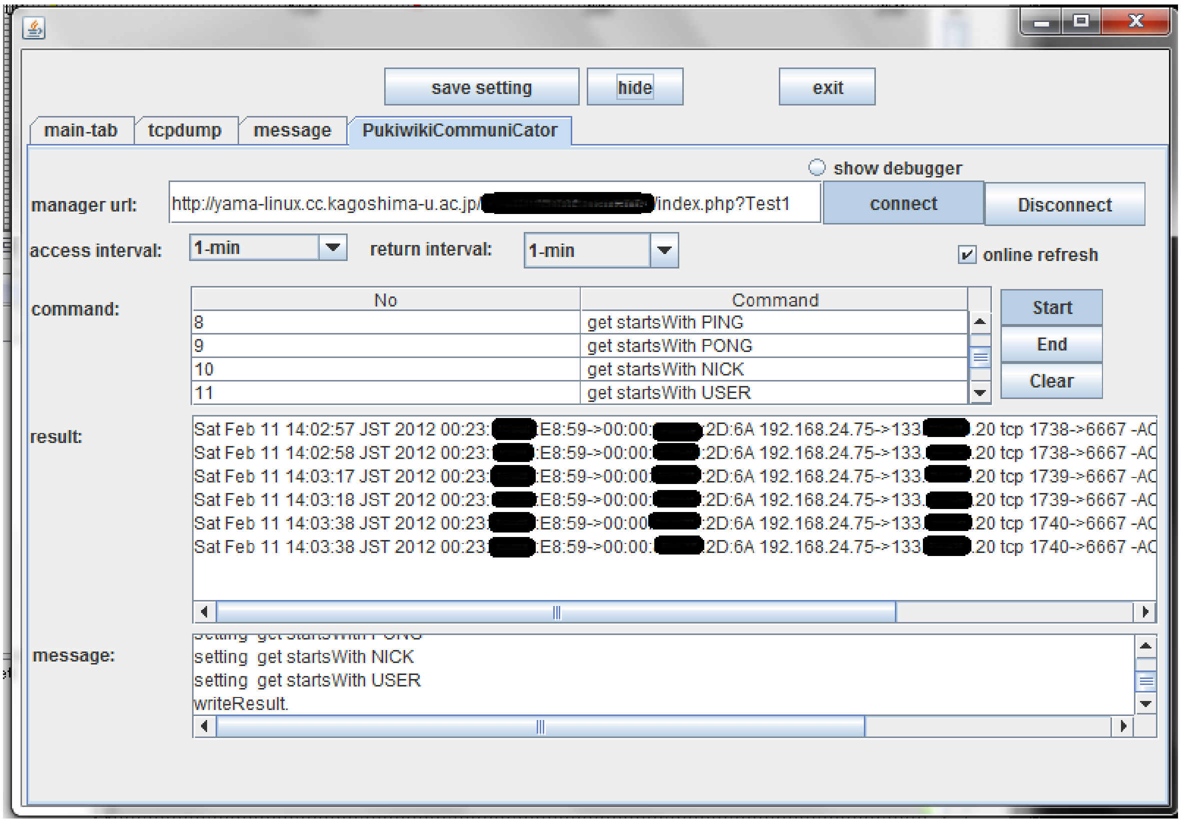
This command drops a packet from the WAN side if the source or the destination address matches the <Address> and writes the packet frame information to the wiki page. The packet is not forwarded to the LAN side.

lan2wan return-syn-ack <Address>

This command drops a packet from the LAN side if the destination address matches the <Address> and writes the packet frame information to the wiki page. This command also returns a syn-ack packet in response to a syn packet with the destination <Address> from the LAN side. This command can be used to capture packets from a malicious bot without the suspicious host to know that the host is under surveillance until the three-way-handshake has been finished.

lan2wan forward <Destination-Address$_1$> to
<Destination-Address$_2$>:<Port>

This command forwards a packet with a destination address of <Destination-Address$_1$> from the LAN side to the application of the <Port> at the host of <Destination-Address$_2$> and saves the original destination IP address, the original destination port, the original source IP address, the original source port, the replaced destination IP address and the replaced port in the security device. When a packet for which the source IP address is the replaced address and the source port is the replaced port comes from the WAN-side NIC, the source address of the packet is replaced by the original destination IP address and the source port is replaced by the original port.
This command is used to mimic the herder site behavior by the central manager using the application such as a telnet server.



○ show debugger

manager url: http://yama-linux.cc.kagoshima-u.ac.jp/[redacted]/index.php?Test1

access interval: 1-min return interval: 1-min ☑ online refresh

command:

No	Command
8	get startsWith PING
9	get startsWith PONG
10	get startsWith NICK
11	get startsWith USER

result:

```
Sat Feb 11 14:02:57 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1738->6667 -AC
Sat Feb 11 14:02:58 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1738->6667 -AC
Sat Feb 11 14:03:17 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1739->6667 -AC
Sat Feb 11 14:03:18 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1739->6667 -AC
Sat Feb 11 14:03:38 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1740->6667 -AC
Sat Feb 11 14:03:38 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1740->6667 -AC
```

message:

```
setting  get startsWith NICK
setting  get startsWith USER
writeResult.
```

Fig. 10 Commands and results of the sensor device.

lan2wan forward <Source-Address> to
<Destination-Address>:<Port>

This command forwards a packet with a source address of <Source-Address$_1$> from the LAN side to the application of the <Port> at the host of <Destination-Address> and saves the original destination IP address, original destination port, original source IP address, original source port, replaced destination IP address and the replaced destination port in the security device. When a packet for which the source IP address is the replaced IP address and the source port is the replaced port comes from the WAN-side NIC, the source IP address of the packet is replaced by the original destination IP address and the source port is replaced by the original destination port. This command is used to transfer packets from the bot-infected host to the notifying web server, which displays a notification of the bot infection to the user of the bot-infected host using the host web browser when the user use the web browser.

lan2wan dns-intercept <IP address$_1$> to <IP address$_2$>

This command intercepts a DNS query for which the answer is <IP address$_1$> and returns the answer as <IP address$_2$>. This command is also used to mimic herder site behavior by the central manager using the application as a telnet server.

Commands and results are written in the pre-formatted area of the wiki page. **Figure 11** shows an example of the commands and results in the wiki page of the monitoring system. A command is written after the label "command:" in a line of the pre-formatted area. Results are written after the line with "result:" label, which should be followed by the last command.

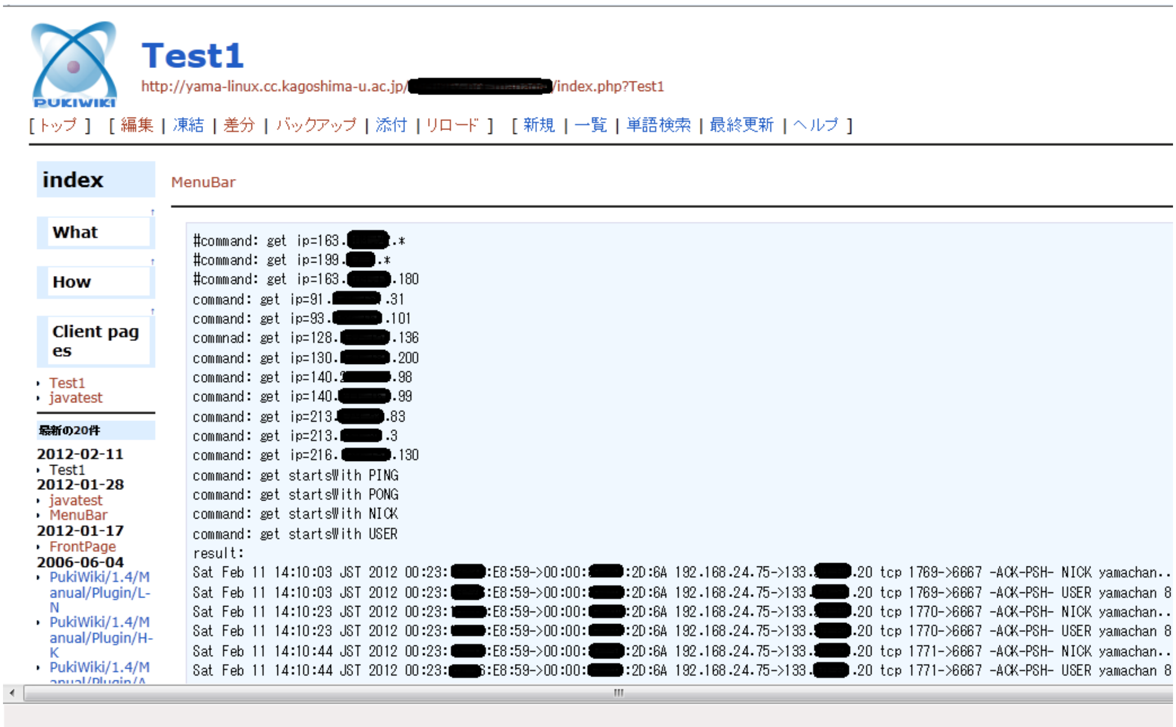
In this figure, lines that start with "#" are comment lines. In this example, commands in this page direct the sensor device to capture IP packets with IP addresses of commands in the page and packets for IRC communication with their frames. Each line of the results shows the time, source MAC address, destination MAC address, source IP address, destination IP address, protocol of the IP packet, source port, destination port, flags of TCP (if the IP packet is TCP), and the pay-load from the left side.

5. Responding to Infection

The following basic procedure is an example of a response to a malicious bot infection after the central manager notices suspicious communication.

5.1 Basic Procedure

1. The central manager identifies the suspicious sub-LAN by using an IDS or a firewall. We assume that the destination IP address from the suspicious host is x.y.z.w.
   Some of recent bots are members of a *P2P botnet* rather than clients of a herder's server [1]. They have list of FQDN of the botnet. Also, records of FQDN are often updated. By referring the list and DNS, bots maintain connectivity to the herder. Such facts make difficult to identify IP addresses of the herder's site. However, we can use commercial network security monitor services recently [5]. We can know the destination IP addresses of packets from the bot in a sub-LAN to the botnet, from a service contracted from them.
2. The central manager asks the sub-manager of the sub-LAN


Test1
http://yama-linux.cc.kagoshima-u.ac.jp/[redacted]/index.php?Test1
[トップ] [編集 | 凍結 | 差分 | バックアップ | 添付 | リロード] [新規 | 一覧 | 単語検索 | 最終更新 | ヘルプ]
index
MenuBar
What
How
Client pages
Test1
javatest
最新の20件
2012-02-11
Test1
2012-01-28
javatest
MenuBar
2012-01-17
FrontPage
2006-06-04
PukiWiki/1.4/Manual/Plugin/L-N
PukiWiki/1.4/Manual/Plugin/H-K
PukiWiki/1.4/M

```
#command: get ip=163.[redacted].*
#command: get ip=199.[redacted].*
#command: get ip=163.[redacted].180
command: get ip=91.[redacted].31
command: get ip=93.[redacted].101
commnad: get ip=128.[redacted].136
command: get ip=130.[redacted].200
command: get ip=140.[redacted].98
command: get ip=140.[redacted].99
command: get ip=213.[redacted].83
command: get ip=213.[redacted].3
command: get ip=216.[redacted].130
command: get startsWith PING
command: get startsWith PONG
command: get startsWith NICK
command: get startsWith USER
result:
Sat Feb 11 14:10:03 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1769->6667 -ACK-PSH- NICK yamachan..
Sat Feb 11 14:10:03 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1769->6667 -ACK-PSH- USER yamachan 8
Sat Feb 11 14:10:23 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1770->6667 -ACK-PSH- NICK yamachan..
Sat Feb 11 14:10:23 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1770->6667 -ACK-PSH- USER yamachan 8
Sat Feb 11 14:10:44 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1771->6667 -ACK-PSH- NICK yamachan..
Sat Feb 11 14:10:44 JST 2012 00:23:[redacted]:E8:59->00:00:[redacted]:2D:6A 192.168.24.75->133.[redacted].20 tcp 1771->6667 -ACK-PSH- USER yamachan 8
```

Fig. 11 Commands and results on the wiki page.

to disconnect the NAT or router of the sub-LAN from the central network infrastructure. The central manager writes commands on the wiki page to capture and filter out the suspicious packets. The command line is as follows:

```
command:  get ip=x.y.z.w
```

The manager configures the remote security device to connect the device to the wiki page.

The central manager sends the portable sensor device to the sub-manager after the sub-manager agrees with the need to identify the suspicious host. The sub-manager connects the remote security device to the sub-LAN and starts the remote security device.

3. The remote security device reads the commands on the wiki page periodically. When the device detects suspicious packets, the device writes the information of the packets with the MAC address of the suspicious host in the sub-LAN on the wiki page. The result line of the information will be as follows:

```
Sun Jun 17 23:28:38 JST 2012
E8:XX:XX:XX:XX:7F->00:XX:XX:XX:XX:6A
192.168.24.73->x.y.z.w tcp 49406->80 -SYN-
```

In the above line, E8:XX:XX:XX:XX:7F is the MAC address of the suspicious host in the sub-LAN, and 192.168.24.73 is the IP address of the suspicious host in the sub-LAN.

4. The central manager confirms the information on the suspicious packets on the wiki page, and if the manager judges the packets to be malicious, the central manager writes commands on the wiki page to capture and filter out the suspicious packets. The command line is as follows:

```
command:  drop smac=E8:XX:XX:XX:XX:7F
```

After the remote security device reads the commands on the wiki page, all packets from the suspicious host will be dropped.

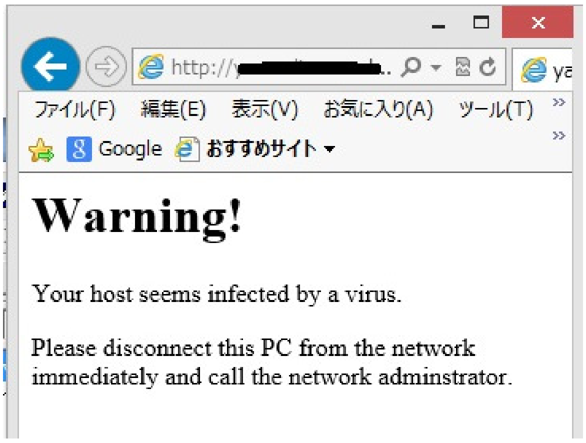
5. The central manager asks the sub-manager to disconnect the host from that sub-LAN.
6. The sub-manager disconnects the suspicious host, which has the source MAC address of the suspicious host, from the sub-LAN and removes the viruses from the host.

A bot may try to investigate whether it is under surveillance or not by accessing not only the herder's site but also well-known web sites, e.g., Google, Microsoft. If their response differs from the herder's site, the bot believes that it is under surveillance, i.e., sandbox. In some case, the bot stops its activity. The above procedure prevents the bot to notice that it is under surveillance because if the bot cann't communicate with the herder's site, the bot cann't communicate with the well-known sites either.

5.2 Survey Activity of the Bot

If the central manager feels that a more deep traffic analysis is required, the manager can prepare a telnet server and write commands for forwarding the packets from the suspicious host to the telnet server on the wiki page. When a suspicious packet is forwarded to the telnet server, the central manager can see the contents of the packet and can respond to the packet on the telnet server. The central manager writes commands on the wiki page to capture and filter out the suspicious packets. The command line is as follows:

```
command:  lan2wan forward dip=x.y.z.w
          to ip=o.p.q.r:23
```



Warning!

Your host seems infected by a virus.

Please disconnect this PC from the network immediately and call the network adminstrator.

Fig. 12 Example page of the notification web server.

In the above line, "o.p.q.r" is the IP address of the telnet server. The central manager may know the kind of the bot of the suspicious host from the commercial network security monitor service and the manager may know commands of the bot from signatures of IDS such as SNORT, other web sites and papers such like Ref. [1]. If the suspicious host seems infected by an IRC based bot, interaction between the bot and the manager will be such as follows:

```
NICK Bot-nick                          // from the bot
:irc.xxx.com 020 * : Please wait while we
      process your connection.         // from the manager
PING                                   // from the manager
PONG                                   // from the bot
:irc.xxx.com 020 #ch
         :Bot-nick .remove             // from the manager
```

By the above interaction, the manager can confirm the assumption that the suspicious host is infected by an IRC based bot. The manager can also stop the activity of the bot by the remove command if it is correct.

The telnet server should be corresponding to only one suspicious host in this case. If traffic of several suspicious hosts should be monitored, several servers, which corresponding to each suspicious host, are required.

5.3 End User Notification

The sub-manager is not always the person who can understand technical terms and technical operations. When the sub-manager cannot identify the suspicious host, the central manager writes a command that transfers packets from the host to the notification web server on the wiki page. The command line is as follows:

```
command:  lan2wan forward sip=192.168.24.73
          to ip=192.168.24.81:80
```

In the above line, 192.168.24.81 is the IP address of the notification web server.

The notification web server notifies the user of the suspicious host that the host is suspicious and asks the user of the host to call the sub-manager. **Figure 12** shows an example of the page of the notification web server.

6. Concluding Remarks

A network security control system for capturing malicious bots is presented. This system provides an easy method for capturing malicious bots using a portable security device and a web site with common wiki software. This system can be used for easy and fast identification of a virus-infected host behind a NAT from the central network infrastructure of an organization.

We already have implemented central functions of the system and we confirmed the ability of these functions. We are implementing rest parts of the system and improving the stability of the system by using this system in our laboratory's LAN

The proposed system can be regard as a bot system. If the proposed system is used for malicious purposes, the system can be classified as malware because the system can obtain and control internal information from the outside. We intend to improve the proposed system in order to eliminate this possibility.

Recent bots use encrypted communication such as https. They also use UDP rather than TCP. The central manager can locate the bot infected host at sub-LAN using this system even if the communication is encrypted or UDP is used, provided the IP address of a herder's host is acquired by the manager. However it is impossible to investigate further more now using the way such as the step 5 of the section five. We are researching to cope with such cases by referring papers such like Ref. [1].

References

[1] Binsalleeh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbai, M. and Wang, L.: On the analysis of the Zeus botnet crimeware tool kit, *2010 8th Annual International Conference on Privacy Security and Trust* (*PST*), pp.31–38 (2010).

[2] Hoshizawa, Y., Okada, K. and Tachikawa, T.: Extracting BOT commands automatically by dynamic analysis, *IPSJ Symposium 2009* (21), pp.265–270 (2009) (in Japanese).

[3] Ishida, M., Nakano, N. and Masuda, H.: Implementation and Evaluation of Transparent Proxy System of Addable Network Functions, *IOTS2011-02*, pp.1–7 (2011) (In Japanese).

[4] Tashima, K., Kondo, T., Kishiba, S., Ohigashi, T., Iwata, N., Nishimura, K. and Aibara, R.: A Management Method of MAC Address Authentication in Large-Scale Campus Networks, IPSJ SIG Technical Reports 2009 (21), pp.265–270 (2009) (in Japanese).

[5] Masuya, M., Yamanoue, T. and Kubota, S.: An Experience of Monitoring University Network Security Using Commercial service and DIY Monitoring, *Proc. 34th Annual ACM SIGUCCS Conference on User Services*, pp.225–230, Edmonton, Alberta, Canada (2006).

[6] Mehdi, S.A., Khalid, J. and Khayam, S.A.: Revisiting Traffic Anomaly Detection using Software Defined Network, *RAID'11*, *Proc. 14th International Conference on Recent Advances in Intrusion Detection*, pp.161–180, Menlo Park, CA, USA (2011).

[7] Shinkawa, T. and Yamanoue, T.: A Visualization of Network Traffic by a 2D Plane of IP address and Port, IPSJ Technical Report, 2006-DSM-043, pp.31–36 (2006) (in Japanese).

[8] Yamai, N., Murakami, R., Okayama, K. and Nakamura, M.: A MAC-address Relaying NAT Router for Host Identificaton from Outside of Internal Networ, *IPSJ Journal*, Vol.52, No.3, pp.1348–1356 (2011) (in Japanese).

[9] Yamanoue, T.: A Draw Plug-in for a Wiki Software, saint, *10th IEEE/IPSJ International Symposium on Applications and the Internet*, pp.229–232 (2010).

[10] Yamanoue, T., Oda, K. and Shimozono, K.: PukiWiki-Java Connector, a Simple API for Saving Data of Java Programs on a Wiki, ACM *WikiSym '11, Proc. 2011 International Symposium on Wikis*, Mountain View, CA, USA (2011).

[11] Yamanoue, T., Oda, K. and Shimozono, K.: A Simple Application Program Interface for Saving Java Program Data on a Wiki, *Advances in Software Engineering*, Vol.2012, Article ID 981783, Hindawi Publishing Corporation (2012).

[12] Yamanoue, T., Oda, K. and Shimozono, K.: A LAN Traffic Visual-

ization System which can Show Changes of Traffic between Past and Now, IPSJ Technical Report, 2012-IOT-16 (2012) (in Japanese).

[13] Yamanoue, T., Oda, K. and Shimozono, K.: A Casual Network Security Using a Portable Sensor Device and Wiki Software, *12th IEEE/IPSJ International Symposium on Applications and the Internet*, pp.387–392 (2012).

[14] Yamanoue, T., Oda, K. and Shimozono, K.: Capturing Malicious Bots using a Beneficial Bot and Wiki, *Proc. 40th Annual ACM SIGUCCS Conference on User Services*, pp.91–96, Memphis, Tennessee, US (2012).

[15] jNetPcap, available from ⟨http://jnetpcap.com/⟩.

[16] KASEYA, available from ⟨http://www.kaseya.com/⟩.

[17] SNORT, available from ⟨http://www.snort.org/⟩.

[18] UNIFAS, available from ⟨http://www.furunosystems.co.jp/product/unifas.html⟩ (in Japanese).

Takashi Yamanoue received his B.S. M.S. and Ph.D. in computer science from Kyushu Institute of Technology, Kitakyushu, Japan, in 1982, 1984 and 1993, respectively. He was a Ph.D. candidate of the Interdisciplinary Graduate School of Engineering Sciences, Kyushu University. He is a professor of the Computing and Communications Center, Kagoshima University. His research interests include P2P, distributed computing, compiler-compilers, web mining and computer assisted teaching systems. He is a member of IEEE, ACM, IPSJ, IEICE, Japan Software Science Society (JSSST), the Robotics Society of Japan (JRSJ).

Kentaro Oda received his M.S. and Ph.D. from the Department of Artificial Intelligence, Kyushu Institute of Technology, Japan, in 1999, 2008 respectively. He is currently an assistant professor of Computing and Communications Center at Kagoshima University since 2009. His current research interests include adaptive middleware architecture, multi-agent systems (robotics soccer RoboCup), and distributed systems. He is a member of ACM, IEEE (IEEE Computer Society).

Koichi Shimozono received his B.E. and M.E. degrees from Kyushu University, Japan, in 1991 and 1993, respectively. He is currently an associate professor in the Computing and Communications Center at Kagoshima University. His research interests include Japanese text processing, distributed systems, educational technology and internetworking. He is a member of IPSJ, IEICE.