How to Design Practical Client Honeypots Based on Virtual Environment

Jin-hak Park*, Jang-won Choi*, and Jung-suk Song*
*Science and Technology Information Security Center
Korea Institute of Science and Technology Information, Daejeon, Korea
Email: {painstars, jwchoi, song}@kisti.re.kr

***Abstract*—Honeypot is known as the most famous and widely deployed tool for collecting malwares on the Internet. Conventional honeypots lure attackers into them by simulating vulnerable applications, programs and services, and are able to collect malwares by monitoring malicious activities of attackers. While client honeypots visit websites linked to URLs which are previously provided by users and collect malwares by analyzing the websites. Since attackers mainly use websites for spreading their well-crafted malwares or compromising their target systems, client honeypots have a remarkable attention for the purpose of collecting malwares effectively. However, most existing approaches focus on only collecting malwares by using open source client honeypots such as Capture-HPC, HoneyClient, HoneyMonkey, etc and analyzing them. In this paper, we present how to design practical client honeypots based on virtual environment. The proposed client honeypots are able to help users who want to develop their own client honeypots and deploy them. The experimental results show that the proposed client honeypots visited 2,276,733 URLs, identified 28,831 malicious URLs and succeeded in collecting 2,115 malwares.**

***Keywords*-Client honeypot, Malicious URL, Malware, Virtual Environment.**

I. Introduction

With the rapid growth and development of the Internet, a lot of cyber threats on it are emerging and evolving. Internet users can access to websites through link URLs of the emails, web surfing, etc. This kind of link URLs can be exposed to various threats. The spread of malwares using zero day vulnerabilities has brought an enormous threat to network security. Therefore, we need to collect malwares and analyze them effectively. Recently, various precautionary methods have been studied on malware collection. Among many approach, honeypot is known as the most famous and widely deployed tool for collecting malwares on the Internet [1], [2], [3], [4], [5].

Conventional honeypots lure attackers into them by simulating vulnerable applications, programs and services, and are able to collect malwares by monitoring malicious activities of attackers. In general, honeypots are divided into two types based on the level of interaction with attacker, namely high-interaction and low-interaction honeypot. Low-interaction honeypot has a limited level of interaction because it only emulates a particular service on a system. On the other hand, high-interaction honeypot has a high level of interaction because it uses the actual systems and services to be accessed by crackers. This means that high-interaction honeypots have higher risk compared with low-interaction one.

Honeypots also can be classified into two types from view point of their operation mode : passive and active honeypots. The passive honeypot indicates that it passively waits for attacks in order to detect them, while the active honeypot, also called a client honeypot, interacts with web pages to identify and determine its potential effect on the browser or the operating system.

Client honeypots visit websites linked to URLs which are previously provided by users and collect malwares by analyzing the websites. Since attackers mainly use websites for spreading their well-crafted malwares or compromising their target systems, client honeypots have a remarkable attention for the purpose of collecting malwares effectively. Therefore, many researcher have been studying a client honeypot to find what kind of malwares.

In this paper, we propose how to design practical client honeypots based on virtual environment. The main contribution of the proposed client honeypots is to provide a technical know-how and real experiences for users who want to develop their own client honeypots and deploy them. Specifically, the proposed client honeypots consist of four main modules, i.e., Hypervisor, URL Crawler, Honeypot Agent and Main Service, which are aiming at visiting web sites in real time, at conducting static and dynamic analysis for them and at finding out malwares effectively. The experimental results show that the proposed client honeypots visited 2,276,733 URLs, identified 28,831 malicious URLs and succeeded in collecting 2,115 malwares.

The rest of the paper is organized as follows. In Section 2, we give a brief description of the existing approaches related to client hoypots. In Section 3, we present the proposed architecture and the experimental results are given in Section 4. Finally, we explain conclusions for given advantages in Section 5.

978-1-5090-2285-4/16 $31.00 © 2016 IEEE
DOI 10.1109/AsiaJCIS.2016.19

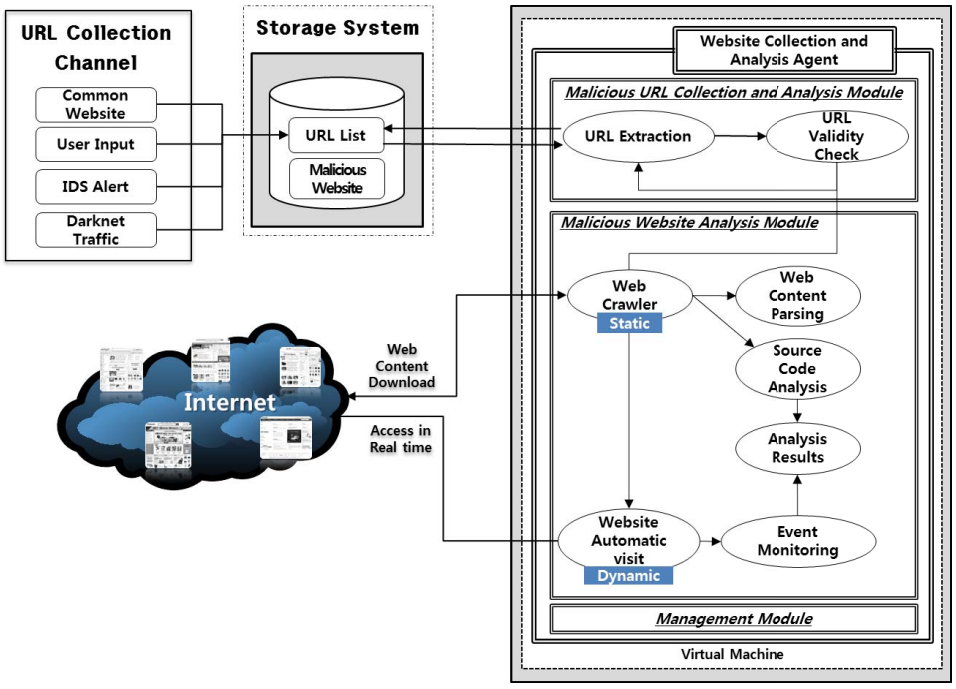


Figure 1. Concept of proposed Client Honeypot

II. Related Work

Recently, many researcher have been studying about client honeypot. Some of many study, firstly, [6] presents the idea of using web-based technology and integrating it with a client honeypot by building a low interaction client honeypot tool called Honeyware. Secondly, [7] analyzes the main approaches used by client honeypots to detect client-side attacks. It also addresses how attackers can evade and hide from client honeypots. Moreover, this paper discusses and analyzes various issues related client honeypots : detection problems, invisibility of honeypots, and integrity issues. Thirdly, [8] proposes multiplication approaches to improve performance efficiency and to enable in-depth analysis on high interaction systems. Finally, [9] focuses on deploying high interaction honeypot system coupled with intrusion detection system on different operating system flavors which work as clients. Clients collect URLs by specifically crafted web links crawler. These URLs are then visited by application needed to visit these URLs. Finally, if these URLs are malicious and exploit the application software, an alert is triggered by signature based intrusion detection system deployed on the machine.

III. Proposed Architecture

A. Concept of Client Honeypot

Figure 1 shows the concept of the proposed client honeypot. The concept consists of five main modules : URL Collection Channel, Storage System, Website Collection and Analysis Agent, Malicious Website Analysis, Management. A URL list of concept is created through the URL collection channels. The collection channels are a common website, user input, IDS alert, darknet traffic, etc. The storage system saves this information. Also, Website Collection and Analysis Agent module confirms validity whether the input URLs are effective form or not. Malicious website analysis is operated static and dynamic analysis. The static analysis is the condition check through the web content parsing and the source code analysis. The dynamic analysis is the event monitoring through the website automatic access in real time. Finally, the management module control overall modules. The main concept is the use of virtual machine.

B. Overall Structure

For an implementation of client honeypot, a concept through various function must has explained in more details. Therefore, Figure 2 shows the overall structure that proposed architecture of a using client honeypot based on virtual machines. This architecture is proposed about URLs analysis of malicious code on website. The main focus of the proposed implementation is a virtual machine, management system, analysis system, and storage system. Also, it can be performed to the static and dynamic analysis that can be connected to offer an URL by a collection channels in real time. The architecture is composed of Honeypot Agent, Hypervisor, URL Crawler, and Main Service. A management

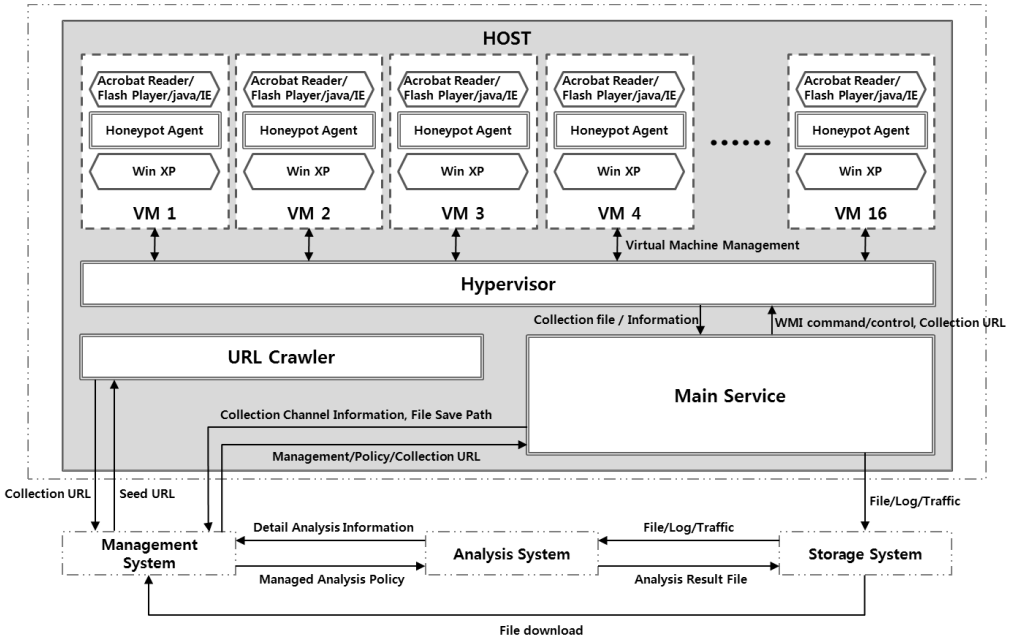


Figure 2. Overall structure of the proposed client honeypot.

system sends seed URLs to a URL Crawler and it finds collection URLs through the seed URLs. The Hypervisor receives the collection URLs from the Main service and feeds them to the virtual machines.

- Honeypot Agent : A Honeypot Agent can be offered to the system monitoring information, raw file collection, and management function. The system monitoring information of file, process, registry, and network through a kernel driver can be collected. Also, the API information through a user level API hook can be collected. A management function is offered to the overall agent setup and the management.
- Hypervisor : A function of host resources control, guest OS control, and transmission control can be controlled.
- URL Crawler : The collection function of link URL on seed and collection URLs, filtering function of collected link URLs information, and insertion of filtered link URLs information to database are operated.
- Main Service : The collected file transmission to the storage system, insertion of information of collected file to database, and after testing collected file, virtual machine restoration to the original state are operated.

We describe the following subsection for detailed.

C. Honeypot Agent

Honeypot Agent is operating on virtual machines. Figure 3 shows the structure of Honeypot Agent. It sets up various URLs of an inputted source and a visited URL through web browser. One of function makes the network collection

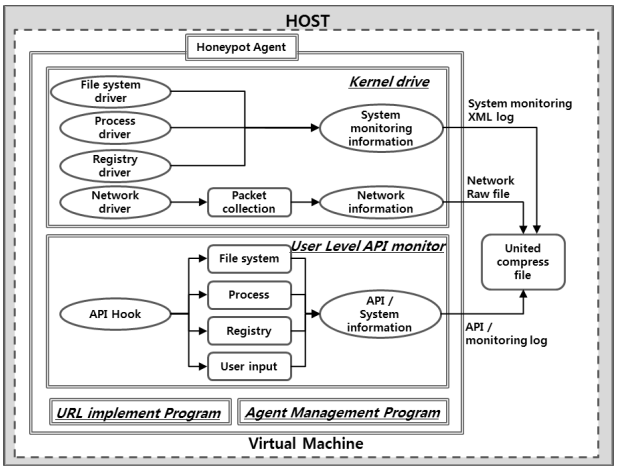


Figure 3. Structure of Honeypot Agent.

information and the system monitoring information such as the file system, process, and registry through a kernel drive. Also, The API hooks information such as the file system, process, registry, and inputted action of user. Through a agent management program, it can conceal the implementing process information from attackers.

To prevent an attackers from recognizing the communication activity, the function is disguised on the transmission protocol and control of packet transmission rate. It is ability to use virtual machine technology to hide the fact

from attackers. A URL implement program is automatically collecting the information by the entered collection URLs. Finally, an agent compress to the united various information. Therefore, we can know the information what change a virtual machine condition. Using virtual environment has several advantages such as easy control for virtual machines, guest OS control, and easy restoration to the original state. However, attackers will know that victim systems are operated by the virtual machines. To overcome this shortcoming, the concealment skill of virtual machines applies to the systems.

D. Hypervisor

The proposed structure has sixty unit Honeypot Agent. To the control for overall Honeypot Agents needs management system. A system consists of three functions : host resources control, guest OS control, and transmission control. The host resources control checks each virtual machines condition. The guest OS control function can control the user access. Finally, transmission control function can control between virtual machine and Main Service.

E. URL Crawler

URL Crawler consists of three functions : the seed URL management, collection and analysis of URL Crawler, and filtering of URL Crawler. Figure 4 shows the structure of the URL Crawler. A seed URL management is received to a seed of URL for crawling from a collection channels that is the entered data or connected another equipment.

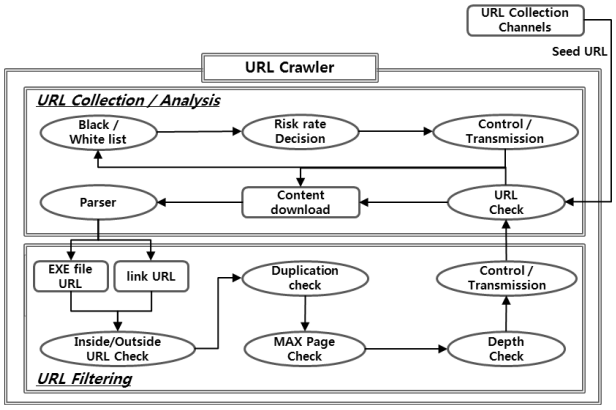


Figure 4. Structure of URL Crawler.

The collection and analysis of URL Crawler is checking the information of URL for an analysis. Firstly, it checks a black and white list. Case of a black list, after a suspicious information is sent to the database then it is downloading a content. Case of a white list, after a information of formal is sent to the database then it is terminated. Besides it is conducting the download and parsing of a content of seed URLs. The filtering of URL Crawler is checking various methods about parsing of a execution file URL and a link URL. It can remove a useless URL by the checking patterns. Also, It can check connecting the possibility of a effective URL. Depth of a link URL can be analyzed to the recursive by the function setup. Therefore, the filtered information data transmits to the database. Finally, this system extracts for collection URLs through seed URLs.

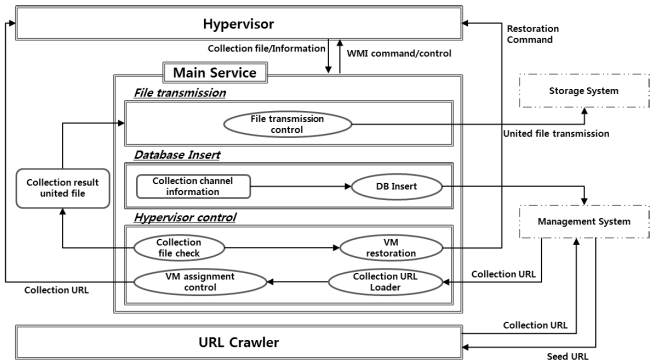
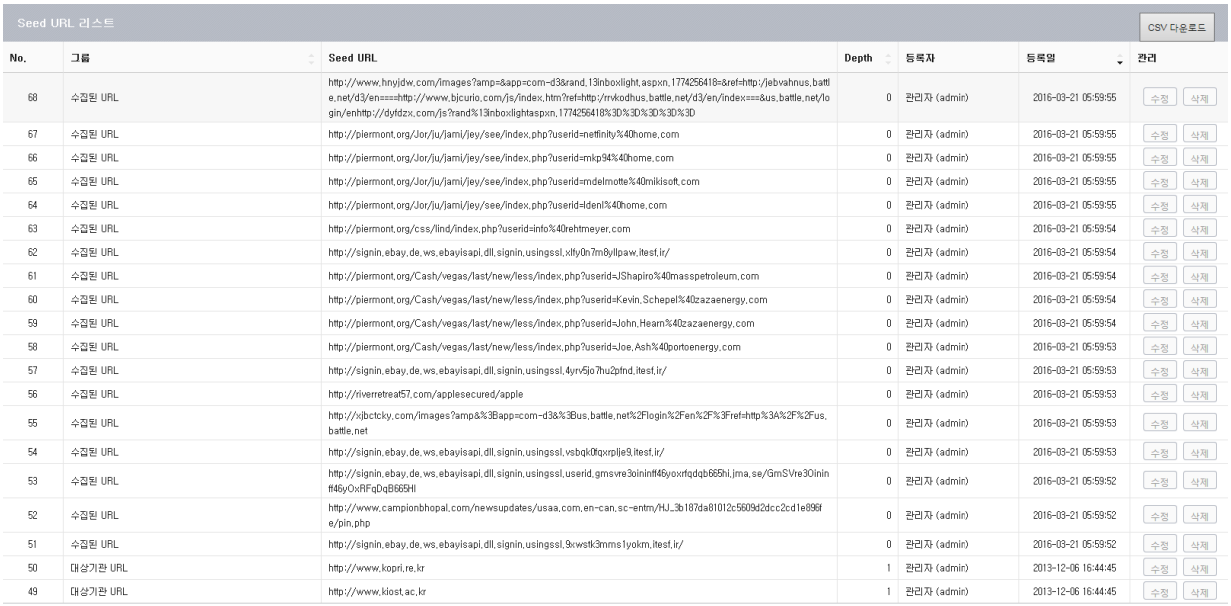


Figure 5. Structure of Main Service.

F. Main Service

In this function can manage the system control. Figure 5 shows the structure of the Main Service. Mainly, it can be operated to the file transmission, insertion to database, and control of Hypervisor. In the case of the file transmission, a united file of collected information by Honeypot Agent transmits to the storage system. The insertion to database transmits the information such as the collection channel and the collection file route to management system. Control of a Hypervisor can be operated that a virtual machine is restoration to the original state after checking of a creation file. Between virtual machines and various functions are the connection way. Through this system can be controlled the various functions.

Focus on the analysis, it divides the static and the dynamic analysis. The static analysis focuses on the source code analysis about a web content as a iframe and a insertion of flash. Through this information, we can get a composition of malicious URL. Therefore, the static analysis makes a comparison the setting signature and the URL content information. In this function is very important what to do set signatures. Therefore, we need to know malicious signatures by a malicious code. The dynamic analysis focuses on the modified system information such as a file, registry, and network process. It functions a connection URL in real time through real browser of a virtual machine. The important system information can be monitored about the actions such as the modify, delete, and creation. Therefore, we can get the infection information by malicious URLs. Through getting the information can be analyzed to the change source and the

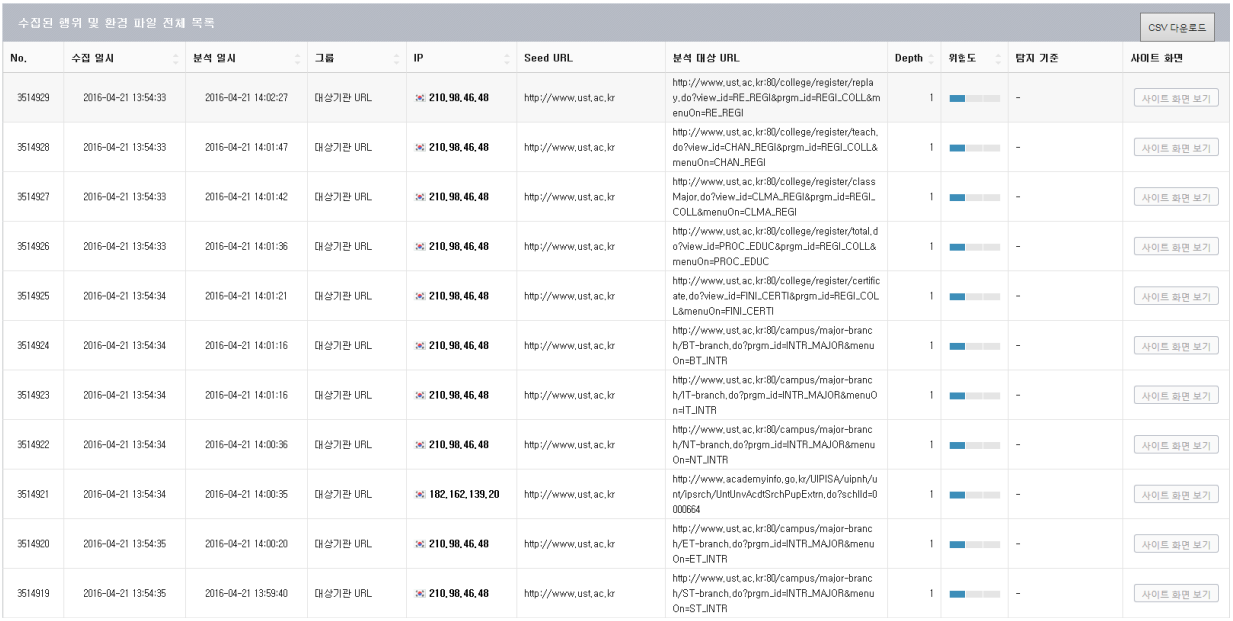


Seed URL 리스트

CSV 다운로드

No.	그룹	Seed URL	Depth	등록자	등록일	관리
68	수집된 URL	http://www.hnyjdw.com/images?amp=&app=com-d3&rand.13inboxlight.aspxn.1774256418=&ref=http://jebvahnus.battle.net/d3/en====http://www.bjcurio.com/js/index.htm?ref=http://rrvkodhus.battle.net/d3/en/index===&us.battle.net/login/enhttp://dyfdzx.com/js?rand%13inboxlightaspxn.1774256418%3D%3D%3D%3D%3D	0	관리자 (admin)	2016-03-21 05:59:55	수정 삭제
67	수집된 URL	http://piermont.org/Jor/ju/jami/jey/see/index.php?userid=netfinity%40home.com	0	관리자 (admin)	2016-03-21 05:59:55	수정 삭제
66	수집된 URL	http://piermont.org/Jor/ju/jami/jey/see/index.php?userid=mkp94%40home.com	0	관리자 (admin)	2016-03-21 05:59:55	수정 삭제
65	수집된 URL	http://piermont.org/Jor/ju/jami/jey/see/index.php?userid=mdelmotte%40mikisoft.com	0	관리자 (admin)	2016-03-21 05:59:55	수정 삭제
64	수집된 URL	http://piermont.org/Jor/ju/jami/jey/see/index.php?userid=ideni%40home.com	0	관리자 (admin)	2016-03-21 05:59:55	수정 삭제
63	수집된 URL	http://piermont.org/css/lind/index.php?userid=info%40rehtmeyer.com	0	관리자 (admin)	2016-03-21 05:59:54	수정 삭제
62	수집된 URL	http://signin.ebay.de.ws.ebayisapi.dll.signin.usingssl.xlfy0n7m8yllpaw.itest.ir/	0	관리자 (admin)	2016-03-21 05:59:54	수정 삭제
61	수집된 URL	http://piermont.org/Cash/vegas/last/new/less/index.php?userid=JShapiro%40masspetroleum.com	0	관리자 (admin)	2016-03-21 05:59:54	수정 삭제
60	수집된 URL	http://piermont.org/Cash/vegas/last/new/less/index.php?userid=Kevin.Schepel%40zazaenergy.com	0	관리자 (admin)	2016-03-21 05:59:54	수정 삭제
59	수집된 URL	http://piermont.org/Cash/vegas/last/new/less/index.php?userid=John.Hearn%40zazaenergy.com	0	관리자 (admin)	2016-03-21 05:59:54	수정 삭제
58	수집된 URL	http://piermont.org/Cash/vegas/last/new/less/index.php?userid=Joe.Ash%40portoenergy.com	0	관리자 (admin)	2016-03-21 05:59:53	수정 삭제
57	수집된 URL	http://signin.ebay.de.ws.ebayisapi.dll.signin.usingssl.4yrv5jo7hu2pfnd.itest.ir/	0	관리자 (admin)	2016-03-21 05:59:53	수정 삭제
56	수집된 URL	http://riveretreat57.com/applesecured/apple	0	관리자 (admin)	2016-03-21 05:59:53	수정 삭제
55	수집된 URL	http://xjbctcky.com/images?amp&%3Bapp=com-d3&%3Bus.battle.net%2Flogin%2Fen%2F%3Fref=http%3A%2F%2Fus.battle.net	0	관리자 (admin)	2016-03-21 05:59:53	수정 삭제
54	수집된 URL	http://signin.ebay.de.ws.ebayisapi.dll.signin.usingssl.vsbqk0fqxrplje9.itest.ir/	0	관리자 (admin)	2016-03-21 05:59:53	수정 삭제
53	수집된 URL	http://signin.ebay.de.ws.ebayisapi.dll.signin.usingssl.userid.gmsvre3oininff46yoxrfqdqb665hi.jma.se/GmSVre3Oininff46yOxRFqDqB665HI	0	관리자 (admin)	2016-03-21 05:59:52	수정 삭제
52	수집된 URL	http://www.campionbhopal.com/newsupdates/usaa.com.en-can.sc-entm/HJ_3b187da81012c5609d2dcc2cd1e896fe/pin.php	0	관리자 (admin)	2016-03-21 05:59:52	수정 삭제
51	수집된 URL	http://signin.ebay.de.ws.ebayisapi.dll.signin.usingssl.9rwstk3mms1yokm.itest.ir/	0	관리자 (admin)	2016-03-21 05:59:52	수정 삭제
50	대상기관 URL	http://www.kopri.re.kr	1	관리자 (admin)	2013-12-06 16:44:45	수정 삭제
49	대상기관 URL	http://www.kiost.ac.kr	1	관리자 (admin)	2013-12-06 16:44:45	수정 삭제

Figure 6. Seed URL list.



수집된 행위 및 환경 파일 전체 목록

CSV 다운로드

No.	수집 일시	분석 일시	그룹	IP	Seed URL	분석 대상 URL	Depth	위험도	탐지 기준	사이트 화면
3514929	2016-04-21 13:54:33	2016-04-21 14:02:27	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/college/register/replay.do?view_id=RE_REGI&prgm_id=REGI_COLL&menuOn=RE_REGI	1		-	사이트 화면 보기
3514928	2016-04-21 13:54:33	2016-04-21 14:01:47	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/college/register/teach.do?view_id=CHAN_REGI&prgm_id=REGI_COLL&menuOn=CHAN_REGI	1		-	사이트 화면 보기
3514927	2016-04-21 13:54:33	2016-04-21 14:01:42	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/college/register/classMajor.do?view_id=CLMA_REGI&prgm_id=REGI_COLL&menuOn=CLMA_REGI	1		-	사이트 화면 보기
3514926	2016-04-21 13:54:33	2016-04-21 14:01:36	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/college/register/total.do?view_id=PROC_EDUC&prgm_id=REGI_COLL&menuOn=PROC_EDUC	1		-	사이트 화면 보기
3514925	2016-04-21 13:54:34	2016-04-21 14:01:21	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/college/register/certificate.do?view_id=FINI_CERTI&prgm_id=REGI_COLL&menuOn=FINI_CERTI	1		-	사이트 화면 보기
3514924	2016-04-21 13:54:34	2016-04-21 14:01:16	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/campus/major-branch/BT-branch.do?prgm_id=INTR_MAJOR&menuOn=BT_INTR	1		-	사이트 화면 보기
3514923	2016-04-21 13:54:34	2016-04-21 14:01:16	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/campus/major-branch/IT-branch.do?prgm_id=INTR_MAJOR&menuOn=IT_INTR	1		-	사이트 화면 보기
3514922	2016-04-21 13:54:34	2016-04-21 14:00:36	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/campus/major-branch/NT-branch.do?prgm_id=INTR_MAJOR&menuOn=NT_INTR	1		-	사이트 화면 보기
3514921	2016-04-21 13:54:34	2016-04-21 14:00:35	대상기관 URL	182.162.139.20	http://www.ust.ac.kr	http://www.academyinfo.go.kr/UIPISA/uipnh/unt/ipsrch/UntUnvAcdtSrchPupExtrn.do?schIId=0000664	1		-	사이트 화면 보기
3514920	2016-04-21 13:54:35	2016-04-21 14:00:20	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/campus/major-branch/ET-branch.do?prgm_id=INTR_MAJOR&menuOn=ET_INTR	1		-	사이트 화면 보기
3514919	2016-04-21 13:54:35	2016-04-21 13:59:40	대상기관 URL	210.98.46.48	http://www.ust.ac.kr	http://www.ust.ac.kr:80/campus/major-branch/ST-branch.do?prgm_id=INTR_MAJOR&menuOn=ST_INTR	1		-	사이트 화면 보기

Figure 7. Collected URL list.

infection path. This system can be checked to the malicious action of a principal target organ URL and a entered URL.

IV. Experimental Results

We show real experimental results of the proposed architecture. The architecture is using a nine virtual machines unit for a principal target organ URLs, and a seven virtual machines unit for an entered seed URLs.

The proposed architecture can check each of virtual machine conditions. Through this information can be known for the current collection condition of virtual machine. It shows the use rate such as CPU, Memory, and HDD. Also,

Table I
LIST OF SEED URLS.

Principal target organ URLs	Entered seed URLs
http://www.science.go.kr	http://www.gfile.co.kr
http://www.sciencecenter.go.kr	http://www.clubnex.co.kr
http://www.ssm.go.kr	http://www.mfile.co.kr
http://www.dnsm.go.kr	http://www.joyfile.co.kr
http://www.ibs.re.kr	http://www.filedok.com
...	...

Table II
LIST OF ANALYSIS RESULTS OF URLS.

2015	Normal	Malicious URLs	malware
Jan	176,574	2,562	133
Feb	219,612	2,222	121
Mar	174,092	2,262	118
Apr	234,886	3,140	291
May	327,879	5,027	440
Jun	400,963	3,804	402
Jul	94,333	3,484	78
Aug	188,725	3,127	337
Sep	102,600	1,576	126
Oct	105,967	539	61
Nov	110,679	630	4
Dec	109,477	458	4
Total	2,245,787	28,831	2,115

we can check the condition either the stand-by phase or the analysis phase. The screen shot can be seen on the current operation monitor.

Table III
LIST OF THE ANALYZED COLLECTION URLS.

Malicious URL	Malware
http://bluechemitopia.krict.re.kr	http://mall.epost.go.kr
http://tv.mpss.go.kr/web/main /main.do	http://ekp.kims.re.kr
http://www.venture.or.kr/kova /index.jsp	http://ebid.kisti.re.kr
http://youtu.be/xTV2VV0Aem0	http://www.filetour.com/
http://news.kbs.co.kr/news /view.do?ncd=3194837&ref=A	http://www.g4b.go.kr/svc/cis/tos/top /TestInfoList.do?relatePrdlstTy=M
...	...

Figure 6 shows the entered seed URL list. So, URL Crawler is operated to collect the link URLs by seed URLs. This action have been periodically operated. A seed URL is weekly entered. Various sources through such as IDS alert, darknet traffic, common website, etc are obtained. Therefore, this information have a suspicious or malicious URL. Also, a principal target organ URL is set. Table I shows seed URLs. It divides to the principal target organ URLs and the entered seed URLs.

Figure 7 shows the information about the collected URLs. Actually, the following analysis result shows the degree of risk whether the normal, malicious URL, or malware. One of results list shows between the malicious URL or malware. A result shows what match a detection standard. In analysis result clicks relevant article. We can get more information such as an image of URL, a packet capture information download, and an extracted file download. Table II shows monthly analysis result in 2015. During the 2015, overall analysis URLs are 2,276,733. The result show normal of 98.6%, malicious URL of 1.3%, and malware of 0.1% among overall analysis URLs. That can be changed to depending on the implemented environment. Table III shows list of a malicious URL and a malware. This list has similar result in monthly because recursively analyze source URLs.

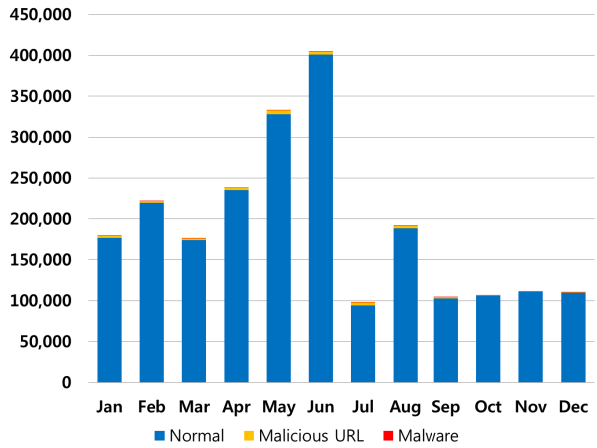


Figure 8. Analysis results of URLs in the proposed client honeypot

The proposed system can show result of a daily, monthly, annual, and specific day. Figure 8 shows most of normal result. Because of result show the influenced by the inputted source URL, the monthly result was detected a few malicious URL and malware.

V. CONCLUSIONS

We proposed practical client honeypots based on virtual environment, which consist of four main modules, i.e., Hypervisor, URL Crawler, Honeypot Agent and Main Service, which are aiming at visiting web sites in real time, at conducting static and dynamic analysis for them and at finding out malwares effectively. The proposed client honeypots have two main advantages. Firstly, the architecture of the Honeypot Agent has a benefit for inspecting whether any malicious activities on a virtual machine occurred or not. Secondly, the architecture of the URL Crawler is able to extract linked URLs from seed URLs, visit them in real time and analyze them statistically and dynamically. By using the proposed client honeypots with these advantages, users are able to develop their own client honeypots and deploy them. In fact, the experimental results show that the proposed client

honeypots visited 2,276,733 URLs, identified 28,831 malicious URLs and succeeded in collecting 2,115 malwares.

References

[1] L. Spitzner, "Honeypots : Tracking Hackers.", Boston : Addison-Wesley Professional, pp.1-480, 2002.

[2] The Honeynet Project, "The Honeynet Project", http://www.honeynet.org/.

[3] C. Leita, K. Mermoud and M. Dacier, "Scriptgen : an automated script generation tool for honeyd.", In Proceedings of the 21st Annual Computer Security Applications Conference, pp.203-214, 2005.

[4] Jungsuk Song, Hiroki Takakura and Yasuo Okabe, "Cooperation of intelligent honeypots to detect unknown malicious codes.", WISTDCS 2008, IEEE CS Press, pp.31-39, 2008.

[5] HoneyTrap, http://honeytrap.mwcollect.org/.

[6] Y. Alosefer and O. Rana, "Honeyware : A Web-Vased Low Interaction Client Honeypot.", Software Testing, Verification, and Validation Workshops(ICSTW), pp.410-417, 2010.

[7] M. Qassrawi and H. Zhang , "Client honeypots : Approaches and challenges.", New Trends in Information Science and Service Science(NISS), pp.19-25, 2010.

[8] M. Akiyama, Y. Kawakoya and T. Hariu , "Scalable and Performance-Efficient Client Honeypot on High Interaction System.", Applications and the Internet(SAINT), pp.40-50, 2012.

[9] R. Shukla and M. Singh , "PythonHoneyMonkey : Detecting malicious web URLs on client side honeypot systems.", Reliability, Infocom Technologies and Optimization(ICRITO), pp.1-5, 2014.