MALWARE AND COUNTRY SUPERVISED CLUSTERING FROM BOTNET/MALWARE DOWNLOAD BEHAVIORS

KHAMPHAO SISAAT

A THESIS SUBMITTED IN FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2016

KMITL-2016-EN-D-018-176

MALWARE AND COUNTRY SUPERVISED CLUSTERING FROM BOTNET/MALWARE DOWNLOAD BEHAVIORS

KHAMPHAO SISAAT

A THESIS SUBMITTED IN FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2016

KMITL-2016-EN-D-018-176

THESIS CERTIFICATION

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

Thesis Title Malware and Country Supervised Clustering from Botnet/Malware Download Behaviors

Student Mr. Khamphao Sisaat

Student Id. 54601002

Degree Doctor of Engineering

Program Electrical Engineering

Thesis Advisor Asst. Prof. Dr. Surin Kittitornkun

Thesis Reference Number KMITL-2016-EN-D-018-176

EXAMINERS		SIGNATURES
Assoc. Prof. Dr. Boontee	Kruatrachue	
Asst. Prof. Dr. Chutimet	Srinilta	
Assoc. Prof Dr. Surapong	Auwatanamongkol	
Assoc. Prof. Dr .Kietikul	Jearanaitanakij	
Asst. Prof. Dr. Surin	Kittitornkun	

Date 7th October 2016 **Time** 10:00-12:00 PM

Place Building A , Conference no.3

(Assoc. Prof. Dr. Komsan Maleesee)

Dean, Faculty of Engineering

7th October 2016

หัวข้อวิทยานิพนธ์	การคลัสเตอร์มัลแวร์และประเทศแบบมีผู้สอนจากพฤติกรรมดาวน์โหลดของบ็อทเน็ท/มัลแวร์
นักศึกษา	นายคำเพ้า สีสะอาท
รหัสนักศึกษา	54601002
ปริญญา	วิศวกรรมศาตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2559
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร. สุรินทร์ กิตติธรกุล

บทคัดย่อ

มัลแวร์จำนวนมากได้ถูกดาวน์โหลดผ่านเครือข่ายอินเทอร์เน็ตโดยเครื่องคอมพิวเตอร์ส่วนตัว หรือ พีซีทั่วโลกที่ติดมัลแวร์ซึ่งรวมตัวกันเรียกว่า บ็อทเน็ต การดาวน์โหลดนี้อาจเป็นการติดมัลแวร์รอบที่สอง หรือ การอัพเกรดเวอร์ชั่นของมัลแวร์นั้นๆ ภายใต้การสั่งการของบ็อทมาสเตอร์ เพื่อเตรียมโจมตีเหยื่อในโลกไซเบอร์ โดยการฟิชชิ่ง ส่งอีเมลสแปม ตั้งเว็บไซต์ไม่ดี การเรียกค่าไถ่ การโจมตีแบบกระจายเพื่อให้เซิร์ฟเวอร์ล่ม เป็นต้น การจัดกลุ่มพฤติกรรมของมัลแวร์มีความสำคัญอย่างมากต่อการตอบสนองต่อภัยคุกคามและเข้าใจพฤติกรรมใหม่ๆของบ็อทเน็ท วิทยานิพนธ์นี้นำเสนอวิธีการคลัสเตอร์มัลแวร์แบบมีผู้สอนโดยอาศัยมิติเชิงเวลา และ เชิงพื้นที่ จากพฤติกรรมการดาวน์โหลดรายสัปดาห์-รายชั่วโมง-รายประเทศ ของมัลแวร์ในชุดข้อมูลปี ค.ศ. 2010 และ 2012 ซึ่งชุดข้อมูลปี 2010 ประกอบด้วยรายการดาวน์โหลดประมาณ 1 ล้านครั้งที่บันทึกโดยเครื่องฮันนี่พ็อทจำนวน 92 ตัว ซึ่งจัดตั้งโดยโครงการ Cyber Clean Center (CCC) ในประเทศญี่ปุ่นตั้งแต่ปี 2009 ถึง 2010 และชุดข้อมูลปี 2012 ประกอบด้วยรายการดาวน์โหลดประมาณ 32 ล้านครั้งที่บันทึกโดยเครื่องฮันนี่พ็อทจำนวน 100 ตัว ซึ่งจัดตั้งโดย Malware Investigation Task Force (MITF) บริษัท Internet Initiative Japan (IIJ) ซึ่งจดทะเบียนในตลาดหลักทรัพย์โตเกียวและนิวยอร์ค

ผลการคลัสเตอร์มัลแวร์ท็อป 30 ในปี 2010 แบบไม่มีผู้สอน แสดงให้เห็นกลุ่มของมัลแวร์ที่แตกต่างกันอย่างเห็นได้ชัดจำนวน 7 กลุ่ม และผลการคลัสเตอร์ประเทศท็อป 30 แบบไม่มีผู้สอน แบ่งเป็นกลุ่มทวีปยุโรป และ กลุ่มนอกทวีปยุโรป ในทำนองเดียวกัน ผลการคลัสเตอร์มัลแวร์ท็อป 20 ในปี 2012 แบบมีผู้สอน ตรงกับมัลแวร์ ตระกูล Conficker.B และ Conficker.C ด้วยค่า Precision และค่า Recall 100.0%, 88.9% และ 91.7%, 100.0%. สำรับคลัสเตอร์ที่ I และ ที่ II ตามลำดับ ยิ่งไปกว่านั้นผลการคลัสเตอร์ประเทศท็อป 20 แบบมีผู้สอน มีความใกล้เคียงกับรายงานฉบับล่าสุด

ในปี 2015 โดย Asghari และคณะซึ่งได้ศึกษาอัตราการแพร่กระจายตัวของมัลแวร์ Conficker จาก 62 ประเทศ ตลอดระยะเวลา 6 ปี ด้วยค่า Precision สูงถึง 75.0% และค่า Recall 86.7%.

Thesis Title	Malware and Country Supervised Clustering from Botnet/Malware Download Behaviors
Student	Khamphao Sisaat
Student ID	54601002
Degree	Doctor of Engineering
Program	Electrical Engineering
Thesis Advisor	Asst. Prof. Dr. Surin Kittitornkun

ABSTRACT

A huge number of botnet malware variants can be downloaded by zombie personal computers as secondary injections and upgrades according to their botmasters to perform different distributed/coordinated cyber attacks such as DDoS, phishing, spam e-mail, malicious Web sites, ransomware, etc. In order to generate a faster response to new threats, to better understand of botnet activities (i.e., easier to write a generic behavioral signature detection, to implement removal procedures, to create new mitigation strategies that work for a whole class of malware, as well as to reduce the size of malware signature database), grouping them based on their malicious behaviors has become extremely important. This dissertation presents a Spatio-Temporal malware and country supervised clustering algorithm based on its (Weekly-Hourly-Country) features of malware download behaviours. The 2010 dataset contains more than 1 million of malware download logs from 92 honeypots set up by Cyber Clean Center (CCC, `https://www.telecom-isac.jp/ccc/en_index.html`), Japan from 2009 to 2010. On the other hand, the 2012 dataset contains more than 32 million of malware download logs from 100 honeypots set up by Malware Investigation Task Force (MITF) of Internet Initiative Japan Inc. (IIJ listed in Tokyo and New York stock markets, `http://www.iij.ad.jp/en/index.html`) from 2011 to 2012.

The Spatio-Temporal malware unsupervised clustering can achieve seven major clusters of Top-30 malware in 2010. The resulted clusters share similar characteristics corresponding to the existing malware databases. In addition, Top-30 source countries can be clustered as the European and non-European group countries. On the other hand, the Top-20 malware supervised clustering results coincidentally correspond to Conficker.B and Conficker.C with relatively high precision and recall rates up to 100.0%, 88.9% and 91.7%, 100.0% for Clusters I and II, respectively. In addition, the resulted two clusters of Top-20 countries are comparable to those with high and

low growth rates recently reported in 2015 by Asghari et al. Therefore, the supervised clustering algorithm can be validated and evaluated to yield precision and recall of up to 75.0% and 86.7%, respectively.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Asst.Prof.Dr. Surin Kittitornkun for his steady guidances and supports throughout my Ph.D study in his laboratory at KMITL (Thailand). I would not be able to accomplish this challenging and interesting topic without his suggestions.

I would like to express my thoughtful gratitude to Prof. Dr. Hiroaki Kikuchi (Meiji University, Japan) and Prof. Dr. Hiroshi Ishii (Tokai University, Japan) for their encouragements, their guidances, advices, supports, and valuable comments on my research. Without them, I would not be able to finish my research work and complete my Ph.D study.

A special thank to my colleague, Mr. Chaxiong Yukonhiatou for being such wonderful helper, discussion and time during my research.

I wish to express my acknowledgement to ASEAN University Network/Southeast Asia Engineering Education Development Network (AUN/SEED-Net) for awarding me the scholarship with the financial support three and a half years for my Ph.D studies in Thailand and Japan. Furthermore, I extend my sincere appreciation to KMITL and Tokai University for giving me the great opportunity to do research in a warmly and friendly environment.

My gratefully acknowledge goes also to all professors, lecturers and supporting staff in International College, Faculty of Engineering at KMITL, JICA Yokohama and Tokai University, who always help and guide me during the whole period of my studies at KMITL and Tokai University.

I would like to recognize all friends; international students, Thai students at KMITL and Japanese students at Tokai University for the enjoyable and stimulating atmosphere that they provided with their companion and friendship.

And finally, I would like to acknowledge to the rest of my family; my mom, my siblings, my wife and daughter for always being there for me, their unconditional support and cheer me up every time I am exhausted and wanted to give up. Their sincere encouragement, their loves and stimulation enable me to go through my degree.

Bangkok, August 2016

Khamphao Sisaat

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation

As computer systems and the Internet become increasingly ubiquitous, the cyber ssecurity threats have also undergone a profound transformation from unstructured and sporadic attacks, to more organized multi-target attacks on a global scale, where the goal is financial profits. The lack of sophisticated protection on average users computers and the high value of enterprise and household targets have attracted skilled and motivated cyber-criminals to launch a wide range of security attacks. These attacks compromise computers, penetrate networks, steal confidential information, send out lots of spam emails, bring down servers and cripple critical infrastructures, leading to severe damages and significant financial losses. According to a recent CSI (Computer Security Institute) survey [10], the average loss from a variety of security attacks was about $100,000 per incident. A recent sample of DDoS attack has been posted in [11] and [12]. Spamhaus came under the biggest attack on 18 March 2013. During this attack, the attack traffic peaked from 10Gbps to 300Gbps, some attacks on major banks may have reached over 150Gbps, global Internet slows after attacking (reported by BBC news), no Internet connection, and its website was unreachable. Another sample of damages has been reported in [13], China faced the largest Distributed Denial of Service (DDoS) attack on 26 August 2013. During the attack, it was leading to a two-to-four hour shutdown of IP addresses using ".cn", the average DDoS attack bandwidth totaled 48.25Gbps, the average packet-per-second rate reached 32.4 million, and the ISPs overwhelmed with huge packet-per-second floods.

The main engine for most organized cyber crimes is various types of malware. Malware, or malicious software, generally refers to various forms of hostile, intrusive and annoying software designed to infiltrate a computer system and to subvert the system for unintentional uses. Typical malware types include viruses, worms, spyware, trojan horses, rootkits, and bots. Spreading a destructive payload, they infect and take control of vulnerable computer systems, using them to facilitate other criminal activities and gain illegal profits [2]. For example, bots typically spread through exploiting software vulnerabilities or employing social engineering techniques to allure naive users to execute malware binaries or intstall. Once a system has been infected, the malware can install spyware and backdoors, transforming these individual victimized systems into a vast network, called a botnet, controlled by the attackers. Botnets are commonly used in launching

cyber attacks such as phishing, spam e-mail, malicious Web sites, ransomware, DDoS (Distributed Denial of Service), etc.

In addition, most malware programs are continuously mutated to evade anti-virus (AV) detectors. Instead of the time-consuming and expensive process of creating a malware program from scratch, malware authors often pursue a more cost-effective solution; reusing existing malware (either binaries or source codes) by slightly altering them to evade AV detectors. Because of this success, such malware variants have evolved into a streamlined process [14], where malware authors employ a broad spectrum of tools and technologies to automatically create malware variants to elude detectors. Figure 1.1 shows the typical techniques including equivalent code substitution, instruction re-ordering, noise insertion and runtime packing (i.e., encrypting or compressing the original binaries into random-looking data and decrypting the content when the malware is executed)

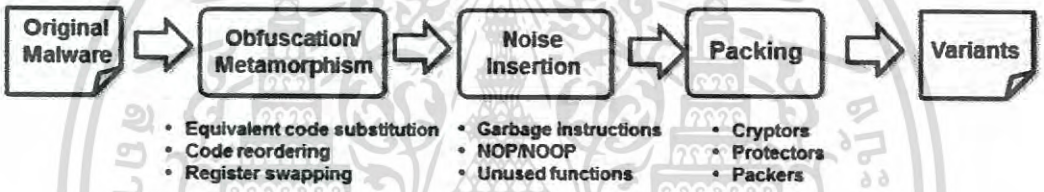

Figure 1.1 Creation of malware variations (Source: Doctoral Dissertation [1])

The ability to automatically and rapidly create variants allows malware authors to replace outdated malware as soon as they become less effective, granting them an advantageous attack window before new detection signatures can be created and deployed. The ease of this malware-mutation process has led to an exponential growth in the number of new malware samples seen in the field as indicated in Figure 1.2.

From Figure 1.2, it can be observed that the number of malware has nearly doubled annually year-to-year. The total number of new malware created in 2009 has reached 2.9 million, which is equivalent to over 8,000 new variants appearing daily. In fact, the total number of malware programs created only in 2009 is more than the total of all malware created over the previous 20 years. However, the release of new variants keeps going. According to the Symantec 2015 Internet Security Threat Report [15], more than 317 million new variants of malware were created in 2014, 26% increase from 2013, or close to 1 million new pieces of unique malware each day. The overall total number of malware is now 1.7 billion and even more. Unfortunately, this trend is likely to continue, and malware will remain the greatest security threat faced by computer users.

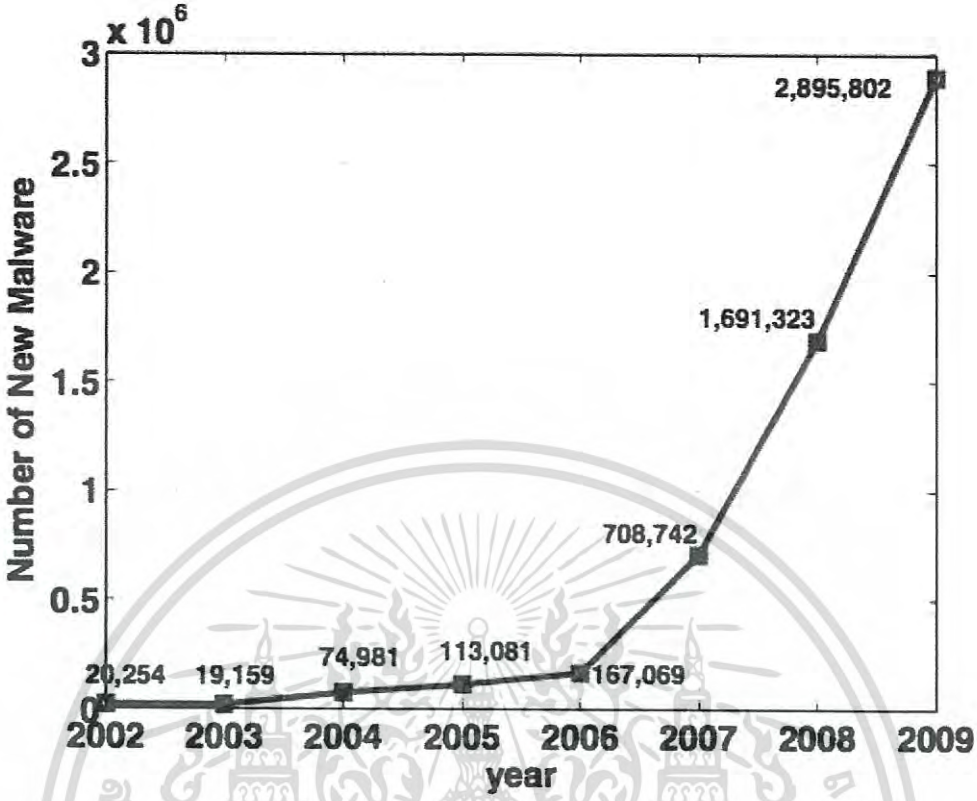

Figure 1.2 Exponential growth in the number of new malware samples (Source: Symantec [2], and Doctoral Dissertation [1])

1.2 Problem Definition

The exponential growth of malware samples/variants has created a major challenge for AV vendors to efficiently analyze and cluster this huge influx of new malware samples and accurately labels them based on their malicious behaviors to protect end-users. An Anti-virus vendor typically receives thousands of suspicious samples every day. These samples are collected from tools such as honeypots and global monitoring sensors [15] or submitted by their partners (e.g., other Anti-virus companies that share malware samples), clients and third-party collection channels [16], [17]. These suspicious samples are typically processed with the following steps [1].

- Malware analysts have to determine if the incoming suspicious samples are indeed malicious, separating malicious programs from benign ones.
- For malicious samples, analysts have to establish which malware family each sample belongs to, and then create family labels for these samples.
- New virus signatures can be generated and distributed to end-users for their protection.

All the above steps require some level of human intelligence through manual analysis, which is expensive, time-consuming and error-prone. The overwhelming number of new malware binary programs has severely strained the scarce human resources of AV companies, making them less responsive to new threats and even allowing some malware to slip through and remain undetected for a significant period of time. For example, there is a typical time window of 54 days between a malwares release and its detection by AV software, and 15% of samples remain undetected after 180 days [18]. As a result, manual analysis has become the major bottleneck in the malware processing workflow, calling for automatic techniques to analyze incoming samples and produce high quality signatures. Such techniques can allow AV vendors to keep up with rapid malware generation and deployment, thus reducing their response time to new security threats.

Numerous countermeasures against botnets including malware analysis, clustering and classification are summarized in [19, 20]. One possible solution is to automatically cluster bot/malware samples and label (name) them according to their similarities [21]. This process is particularly useful because once a number of different variants of the same malware have been identified and grouped together. It is easier to write a generic behavioral signature detection that can be helpful to detect future malware variants with low false positives/false negatives, implement removal procedures, create new mitigation strategies that work for a whole class of programs [22], as well as reduce the size of malware signature database [23].

Based on several recent studies, malware clustering can be divided into Pre-Labeled (based on bot/malware hash values) [22, 24–31] and Post-Labeled (labeled those hash values by virus scanner) [23, 32–38] approaches. On the other hand, the clustering features can be classified as host based such as file, registry key, system call, etc. and network based, e.g. HTTP, IRC, SMTP, DNS, network flow characteristics. However, the existing approaches were targeted at different levels of characteristics of malware ranging from binary contents (physical) to network (temporal) behaviors but somewhat limited to geographical locations.

This dissertation proposes a novel method called *"Spatio-Temporal Supervised Clustering"*, which is a combination of malware features in terms of temporal and spatial (country) download behaviors . Both of them can characterize each malware better thus yield sooner malware prevention/mitigation. The method is applied to cluster the most frequently downloaded Top-30 bots or malware and Top-30 source countries from more than 1 million logs in 2010 CCC dataset as well as Top-20 malware and Top-20 source countries from more than 32 million logs in 2012 IIJ MITF dataset. These logs were recorded due to honeypots set up by Cyber Clean Center

(CCC) and Internet Initiative Japan, Inc. Malware Investigation Task Force (IIJ MITF) in 2010 and 2012 respectively.

1.3 Research Goals

Based on the problem definition, the main goals of this dissertation are as follows.

1. To group suspicious samples of bot/malware into the same family based on their similar behaviors.
2. To group the source countries of spreading malware into the same group based on the similarity of malware spreading behaviors in particular geographical regions.

In order to achieve the goals, this dissertation focuses on the following studies:

- Identify temporal malware/bot download behaviors in 2010 CCC (Cyber Clean Center) and 2012 IIJ MITF (Internet Initiative Japan, Inc. - Malware Investigation Task Force) datasets.
- Perform malware clustering according to Weekly, Hourly, and Country download behaviors in 2010 (unsupervised) and 2012 (supervised).
- Perform country clustering according to Weekly, Hourly, and Malware download behaviors in 2010 (unsupervised) and 2012 (supervised).

1.4 Contributions

The main contributions of this dissertation are as follows:

- The post-labeled malware supervised clustering scheme in this research relies on Temporal (Weekly, Hourly) and Spatial (Country) download behaviors.
- It formally introduces a new malware source country clustering method that incorporates both temporal (Weekly) and malware download behaviors.
- The proposed method can be applied to assist malware analysts and AV vendors in P2P botnet detection/mitigation/prevention efforts.

1.5 Organization of the Dissertation

This dissertation consists of five main chapters, which covers all of our research on 2010 CCC and 2012 IIJ MITF datasets. Its structure can be briefly summarized as follows.

Chapter 2 describes botnets and backgroud involved in this dissertation. Botnets can be mainly categorized as Centralized Command and Control (C&C) and P2P C&C Botnets. Previous work with similar objectives can be summarized in Bot/Malware Clustering and Country Clustering sections. Clustering algorithms including partitional clustering and hierarchical clustering are provided. Finally, CCC and IIJ MITF organizations and datasets are explained.

Chapter 3 presents the proposed method and tools that support this dissertation.

Chapter 4 gradually shows the results and compare the clustering results with the existing malware databases for malware clustering. In addition, country clustering results can be achieved and compared with resent references.

Chapter 5 is the last chapter providing conclusions and future work.

Appendix A describes the statistics of Top-30 downloaded malware and countries in 2010 CCC dataset.

Appendix B gives a brief description of the statistics and details of Top-20 most downloaded malware and countries in 2012 IIJ MITF dataset.

Appendix C provides a summary of malware and country hierarchical clustering in 2010 with a variety of feature (i.e., temporal, spatio/malware, and temporal+spatio/malware), dissimilarity (i.e., correlation and cosine), and linkage (i.e., single, average, and complete) options.

CHAPTER 2

BACKGROUND

This chapter presents a review of the research background associated with botnets and their countermeasures in order to fulfill the objective of this dissertation. This chapter is structured into three main sections; Bot/Botnets, Related Work, and Datasets as following.

2.1 Bot/Botnet

This section begins with the definitions of "bot" and "botnet", the key words of this dissertation. A bot in its original meaning is a "software robot", which performs specific tasks semi-automatically according to the human commands. Not all bots are malicious [39], and there are actually many practical bots in the computer fields such as the followings.

- **Web crawler:** A software that automatically travel (visit) from Web site to Web site to gather information of those sites. Web crawlers are the basis of most of Web search engines.
- **Game Bot:** In computer games, a bot means a computer-controlled player with some AI (Artificial Intelligence).

In security field, however, a bot means a compromised personal computer (PC) by attackers. By means of infection to viruses, worms, and most likely Trojan horses, normal PC becomes a robot for the attackers, and performs attacks instead of the attacker's computer. The owner of the infected PC often does not realize that his/her PC has been infected, since he/she can use the PC just as normally. But the same PC will secretly join the attackers activities. The bot source code is usually professionally written by some (funded) criminal groups and includes a rich set of supported functionalities [40] to carry out many malicious attacks and activities. The infected PC (bot) is often called a "Zombie".

On the other hand, a botnet (short for robot network) is essentially a network of bots that are under the control of an attacker (usually referred to as "botmaster" or "botherder") via some command and control (C&C) channels (protocols). Attackers can control hundreds or thousands zombies as a "Botnet" to cooperatively and simultaneously perform malicious activities, such as DDoS attack, spam email, malware infection, and so on.

Figure 2.1 depicts an example of a simple botnet being commanded to launch a DDoS attack against a competitor or individual. The numbered steps illustrate a timeline from a new botclient joining the botnet and then participating in the DDoS attack.

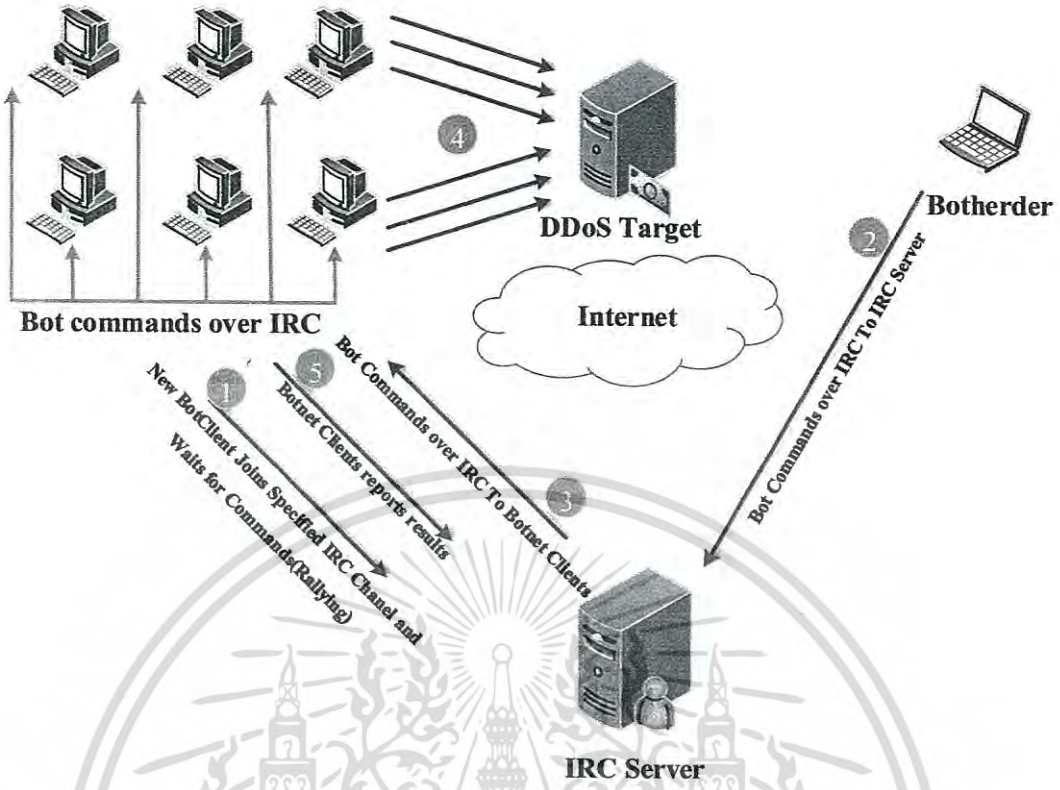

Figure 2.1 Simple botnet being commanded to launch a DDoS attack [3]

2.1.1 Botnet Life Cycle

Botnets follow a similar set of steps throughout their existence. Figure 2.2 illustrates the common life cycle of a botnet client. Our understanding of the botnet life cycle can improve our ability to both detect and respond to botnet's threats. The life of a botnet client, or bot begins when it has been exploited. A prospective botclient can be exploited via malicious code that a user is tricked into running, attacks against unpatched vulnerabilities, backdoors left by Trojan, worms, or remote access Trojans, and password guessing or brute force access attempts, etc.

During the Initial Infection phase, the attacker scans a target subnet for known vulnerability and infects victim hosts through different exploitation methods such as:

- **Unpatched vulnerabilities**

  To support spreading via an attack against unpatched vulnerabilities, most botnet clients include a scanning capability so that each client can expand the botnet. These scanning tools first check for open ports. Then they take the list of systems with open ports and use vulnerability-specific scanning tools to scan those systems with open ports associated with known vulnerabilities. Botnet scans for host systems that have one of a set of known

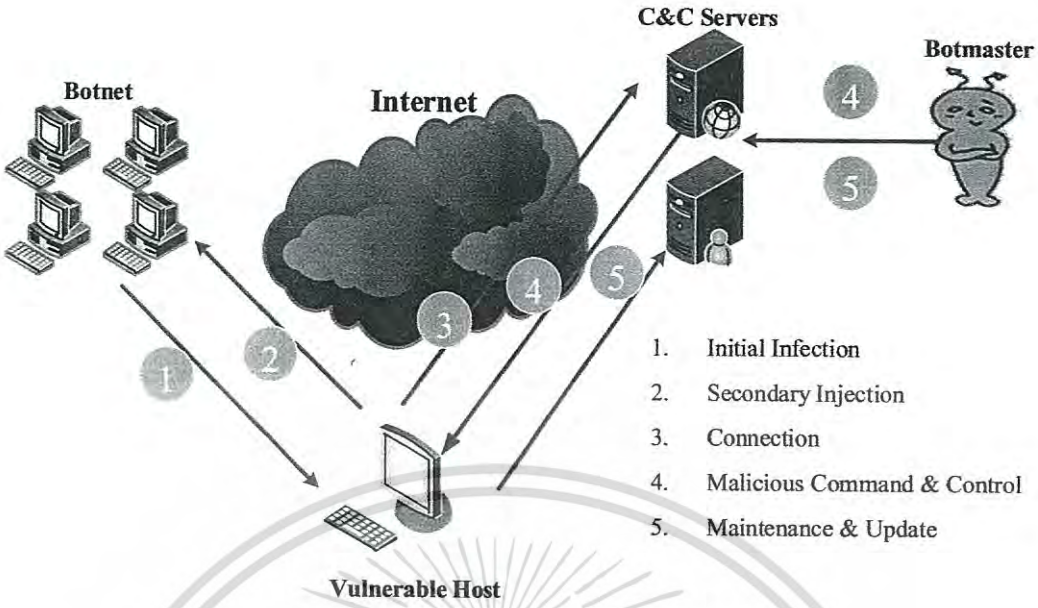

Figure 2.2 A typical botnet life cycle [4]

vulnerabilities, when compromised, it permits remote control of the vulnerable host. A new development is the use of Google to search for vulnerable systems. *Vulnerabilities Commonly Exploited by Bots:*

- Agobot spreads via several methods including:
  Remote Procedure Call (RPC), Distributed Component Object Model (DCOM) (TCP ports 135, 139, 445, 593, and others) to Windows XP systems. RPC Locator vulnerability File shares on port 445. If the target is a Web server: The IIS5 WEBDAV (Port 80) vulnerability [41].

- SDBot Spreads through the following exploits:
  NetBios (port 139), NTPass (port 445), DCom (ports 135, 1025), DCom2 (port 135), MS RPC service and Windows Messenger port (TCP 1025), ASN.1 vulnerability, affects Kerberos (UDP 88), LSASS.exe and Crypt32.dll (TCP ports 135, 139, 445), and IIS Server using SSL, UPNP (port 5000), Server application vulnerabilities, WebDav (port 80), MSSQL (port 1433), Third-party application vulnerabilities such as DameWare remote management software (port 6129) or Email IMAPD Login username vulnerability (port 143) [42].

IRCBot, Botzori, Zotob, Esbot, a version of Bobax, and a version of Spybot attempt to spread by exploiting the Microsoft Plug and Play vulnerability (MS 05-039).

- **Backdoors left by Trojans**

  Some botnets look for backdoors left by Remote Access Trojans. Remote Access Trojans include the ability to control another computer without the knowledge of the owner.

  – SDBot exploits the following backdoors:

    Optix backdoor (port 3140), Bagle backdoor (port 2745), Kuang backdoor (port 17300), Mydoom backdoor (port 3127), NetDevil backdoor (port 903), SubSeven backdoor (port 27347) [42].

- **Password guessing and brute force attacks**

  RBot and other bot families employ several varieties of password guessing. According to the Computer Associates Virus Information Center, RBot spreading is started manually through remote control. It does not have an automatic built-in spreading capability. RBot starts by trying to connect to ports 139 and 445. If successful, RBot attempts to make a connection to the windows share (\\<target>\ipc\$), where the target is the IP address or name of the potential victims computer. If unsuccessful, the bot gives up and goes on to another computer. It may attempt to gain access using the account it is using on the attacking computer. Otherwise, it attempts to enumerate a list of the user accounts on the computer. It will use this list of users to attempt to gain access. If it can't enumerate a list of user accounts, it will use a default list that it carries.

After the Initial Infection, in Secondary Injection phase, the infected PCs execute a script known as shell-code. The shell-code fetches the binary image of the actual bot from the specific location via FTP, HTTP, or P2P. The bot binary installs itself on the target host. Once the bot program is installed, the PC turns in to a 'zombie' and runs the malicious code. The bot application starts automatically each time the zombie is rebooted.

In the Connection phase, the bot program establishes a Command and Control (C&C) channel and connects the zombie to the C&C Server. Upon the establishment of C&C channel, the zombie becomes a part of attacker's botnet army. Within the Connection phase, the actual botnet C&C activities will be started. The botmaster uses the C&C channel to disseminate commands to his bot army. Bot programs receive and execute commands sent by botmaster. The C&C channel enables the botmaster to remotely control the action of a large number of bots to conduct various illicit activities.

The last phase is to maintain and update bots. In this phase, bots are commanded to down-

load an updated binary image (file). Bot controllers may need to update their botnets for several reasons. For instance, they may need to update the bot binary to evade detection techniques, or they may in tend to add new functionalities to their bot army. Moreover, sometimes the updated binary moves the bots to a different C&C Server. This process is called 'server migration' and it is very useful for botmasters to keep their botnets alive. Botmasters try to keep their botnets invisible and portable by using Dynamic DNS (DDNS) to facilitate frequent updates and changes of server locations. In case authorities disrupt a C&C Server at a certain IP address, the botmaster can easily setup another C&C Server instance with the same name at a different IP address. IP address changes in C&C Servers propagate almost immediately to bots due to short time-to-live (TTL) values for the domain name set by DDNS providers. Consequently, bots will migrate to the new C&C Server location and will stay alive.

2.1.2 Botnet Architectures

As with normal networks, botnets are also structured in various architectures. One inherent property of all botnet architectures is that the network allows a botmaster to send commands to the bots in some way. Similarly, although not a strict requirement in every botnet, most botnet designs also allow the bots to send feedback to the botmaster.

2.1.2.1 Centralized Botnets

In centralized botnets as illustrated in Figure 2.3(a), all bots connect to a single command and control (C&C) server. Anytime attackers wish to launch some attack such as a DDoS attack by sending special commands to C&C servers with instructions to perform an attack on a particular target, and any infected machines communicating with the contacted C&C server will comply by launching a coordinated attack.

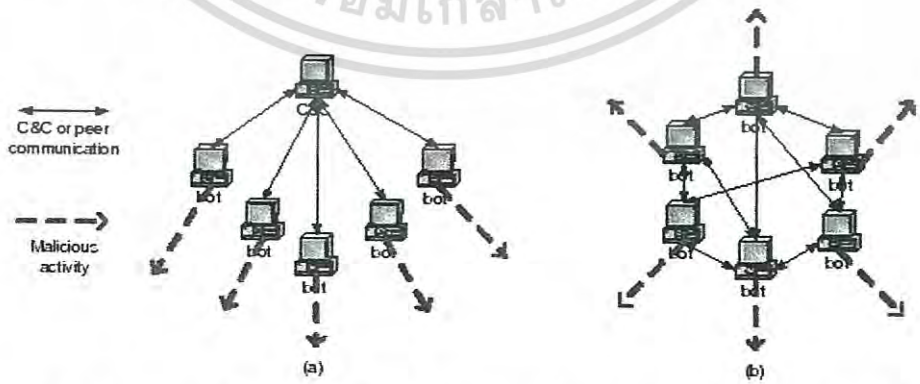

Figure 2.3 Possible structures of a botnet: (*a*) Centralized; (*b*) Peer-to-Peer [5]

Botnet C&C servers often exist in one of four structures each with pros and cons: star,

multi-server, hierarchical, and random:

- Star topology botnets rely on one central C&C server, which sends commands to every bot in the botnet. This configuration allows for reliable, low-latency communication. However, it can be easy to take down, as there is only one C&C server.
- Multi-server topology botnets are very similar to star topology botnets, except that the central server consists of a series of interconnected servers that allow for redundancy (preventing the single point of failure problem of star topology botnets); however, setting up multiple connected C&C servers may require more planning and overall be more difficult than just using a single server.
- Hierarchical topology botnets (involving a series of C&C servers in a hierarchy) allow for botnet owners to more easily divide their botnet up into separate chunks for re-sale or renting, as well as prevent researchers from enumerating the location of all other C&C servers and bots within a network with only a few captured C&C servers due to the restricted visibility of the entire botnet from lower hierarchy certain servers. Additionally, commands that have to travel through a large hierarchy of C&C servers in order to reach bots may add to latency.
- Random topology botnets do not rely on any C&C servers; rather, all botnet commands are sent directly from one bot to another if they are deemed to be signed by some special means indicating that they have originated from the botnet owner or another authorized user. Such botnets have very high latency, and will often allow for many bots within a botnet to be enumerated by a researcher with only one captured bot. Many times special forms of encrypted bot to bot communication over public P2P networks is used in conjunction with a more complex C&C server topology in order to render such botnets that are particularly difficult to dismantle.

(1) **Communication Protocols:** One of the more popular botnet C&C channel is the IRC (Internet Relay Chat) protocol. Its original version was RFC 1459 [43] with latest update in RFC 2813 [44].

The basic idea is that an IRC user must connect to an IRC server at a certain port (traditionally port 6667), select a nickname (handle), and join one or more channels with a possibly optional password. The important thing is that the logic that glues IRC together is the IRC channel name. Which is a logical chat room.

Figure 2.4 shows two IRC networks, both organized around channels. Network 1 is organized around the Linux chat channel and consists of two servers and a number of client hosts. Network 2 has one server (which happens to be a botnet C&C) and a couple of clients. With Network 2, the channel name is Local Security Authority Subsystem Service 455 (lsass 455). Using the IRC protocol, a client sends a data (PRIVMSG) message to an IRC channel, which is an abstraction for a set of users on possibly different client computers and one or more servers. Channel names are ASCII strings with a little bit of syntax sugar possible.

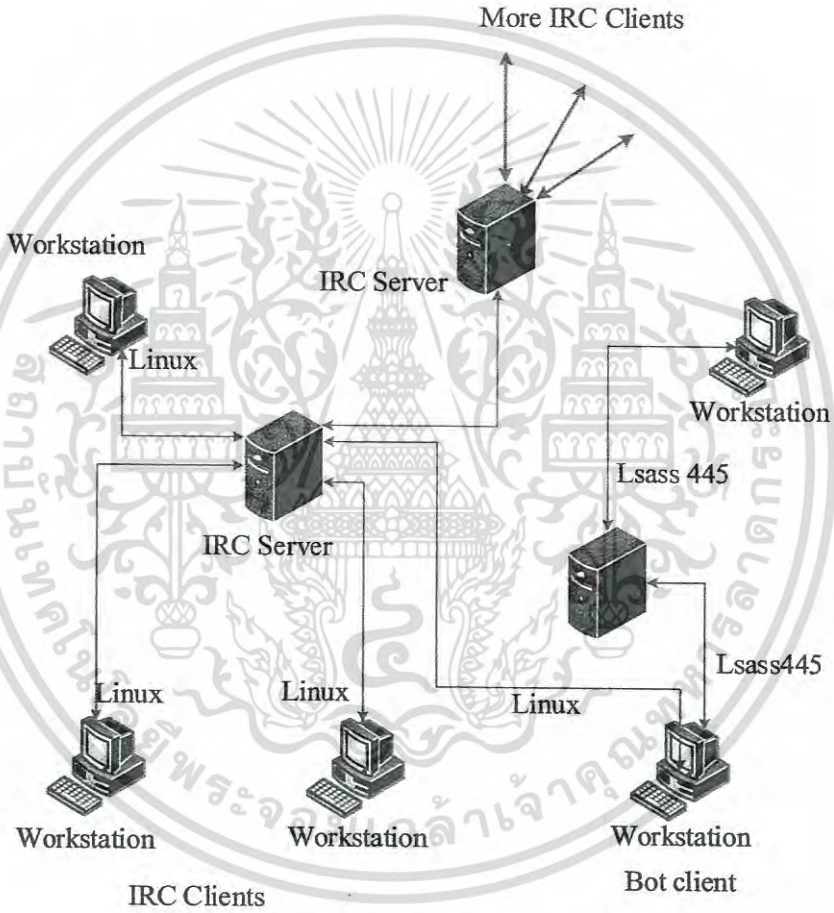

Figure 2.4 The Two IRC Networks [3]

The server that the client is directly connected to take the message and forwards it to other directly connected clients as long as the client has logged into the channel. IRC is said to be a logical mesh network and the data is flooded to other potential recipients in the mesh. This means data goes one way to all the logical clients through all the servers. In other words, the servers make sure the message is not sent twice to any client interested in the channel.

The four kinds of IRC protocol messages are as follows:

- **JOINS** are used by an IRC client to log into a channel on a server. The channel name and password are part of the JOIN message.
- **PINGS** are sent from a server to a client to discover if the client is still interested in the channel and has not for example crashed or gone away otherwise. Typically, PINGS are sent in a periodic fashion at some multiple of 30 seconds.
- **PONGS** are returned from the client to the server to show that it does not want to be logged out and still exist.
- **PRIVMSG** contains both the channel name and data sent to the channel name.

(2) **Botnet Families:** Figure 2.5 shows the activity of some 25+ prevalent botnets in terms of consecutive C&C activity by family. The x-axis reflects the time period since February, 1st, 2010 until February, 13th, 2013, while the y-axis lists the botnet families. A star depicts a dedicated takedown action. Note that, in case of Mariposa and Mega-D, the takedown actions have taken place before the beginning of the time period in this figure. The Mariposa takedown occurred on December 23rd, 2009, and Mega-D has been taken down since November 2009.

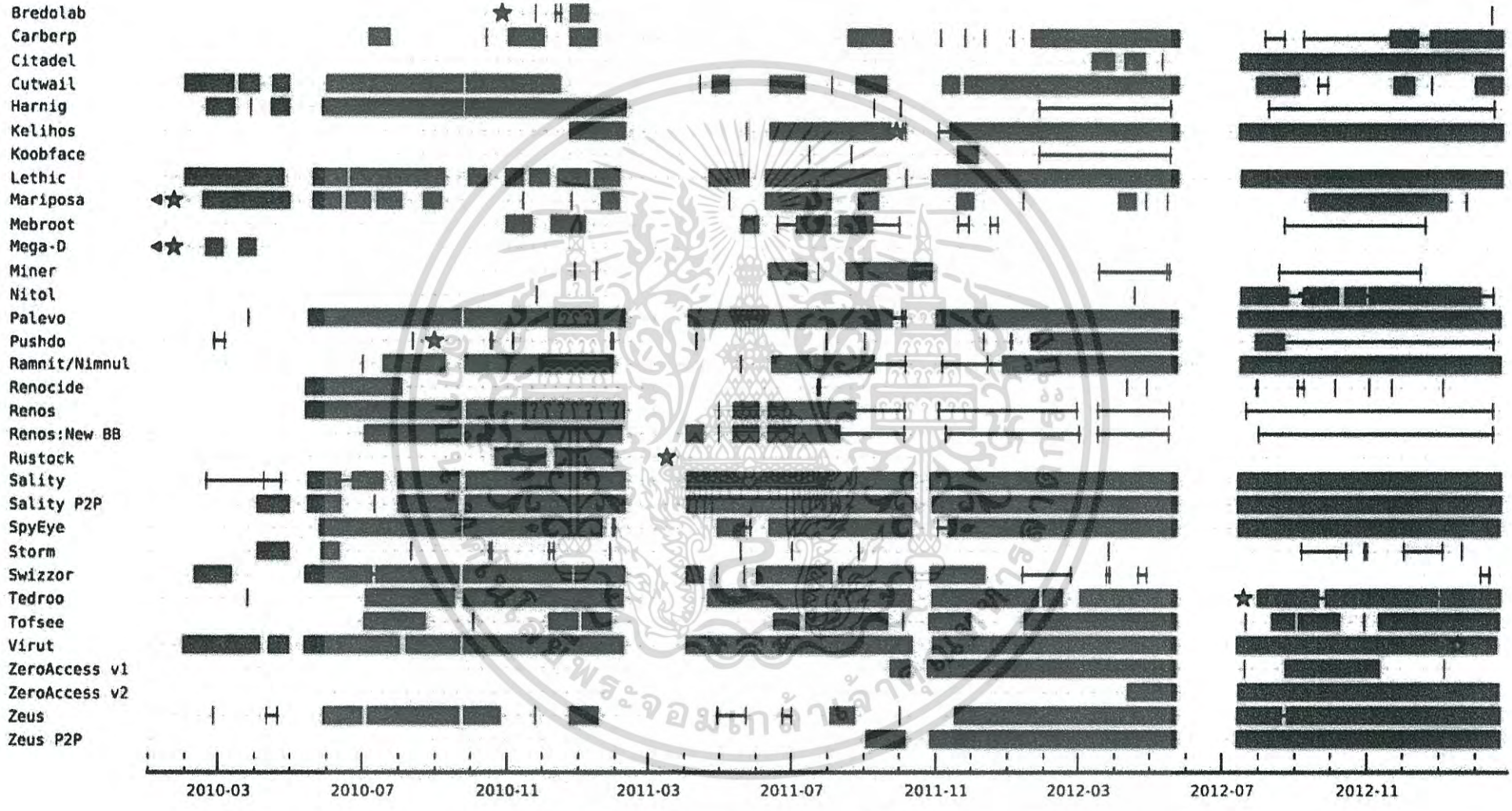

Figure 2.5 Major Botnet Families including Centralized Botnets [6]

In these cases, the stars are placed on the start of the time period in order to visualize the preceding takedowns. A thin black line with black markers represents the time period where new binaries are distributed, but none of the binaries exhibit an active C&C channel. Inactive C&C channels are caused, for example, by outdated binaries, unreachable C&C servers or sinkholing (possibly after successful takedowns). The Rustock takedown operation by Microsoft Digital Crime Unit is an example of such a successful takedown. Similarly, the Harnig botnet, which is believed to be the main distributor of Rustock (most likely via pay-per-install), exhibits inactive C&C since March, 2011. This correlates to the time of the Rustock takedown operation.

In contrast to inactive C&C, a thick red bar represents the time periods where active C&C communication has been observed. Cutwail, Virut, Sality, Palevo and Lethic are examples of such long-living botnets. While Sality has had a builtin P2P component for years, Virut uses number of domains in order to contact the C&C servers. In case the domains cannot be resolved, a domain generation algorithm (DGA) gets involved. However, only recently, a takedown operation addressed Virut. Palevo seems to keep it simple, neither P2P nor DGA nor significantly low TTLs on the DNS responses (which could indicate fast flux). Instead Palevo seems to rely on new and migrating domains. Similar to Palevo, Lethic manages to bootstrap its C&C by plain old DNS resolution.

2.1.2.2 Peer-to-Peer Botnets

Peer-to-Peer or P2P botnets are fully distributed botnets, in which the bots retrieve their commands from other bots via the P2P network. As Figure 2.3(b) shows P2P bots keep track of other bots in the botnet, following architecture without central servers. The lack of central components makes the resilience of P2P architectures attractive for botmasters. In particular, P2P botnet continues to operate even if a large number of bots are removed from it, and the P2P network quickly heals itself from sudden network changes. On the other hand, P2P networks can be subjected to other mitigation techniques, ranging from enumerating all infected bots to P2P-based sinkholing or P2P network partitioning.

The first P2P botnet to be spotted was Sinit (aka Calyps.a or Calypso) in 2003, by Joe Stewart at LURHQ (now SecureWorks) [45]. Later on, Agobot variants had a P2P option and Phatbot made the leap to P2P for real.

(1) **Technologies and Protocols:** P2P protocols became more common with Napster whose

nodes in a network act as both clients and servers. Other P2P protocols followed such as Gnutella, eDonkey, BitTorrent and Kademlia. These protocols are completely decentralized and attractive for botmasters. Botnets are so named because they use P2P technologies and protocols either for spreading, for C&C or both. Most of the P2P botnets that encrypt their communications use asymmetric encryption for C&C and symmetric key encryption for client-to-client communication. This makes it harder to detect malicious code (the IDS has to be able to read it to recognize it as malicious).

P2P technology presented botnet controllers with both pros and cons. On the plus side, the bots were decentralized and not reliant on single point of failure. On the negative side, programming could potentially be injected from any peer in the botnet. It can be solved by introducing cryptographic keys. Another type of P2P botnet relies on a centralized location for tracking, much like P2P networks. For using one of the public P2P networks, this has to be the case. The main problem with advancing control channel technology over the years is that the more complex it is, the easier to track down the botnet. In P2P, this would be especially true, as by being a simple peer, one can discover other bots without taking any action. Actually, in P2P botnets they are: *no centralized C&C servers acting as C&C server and client and much more resilient against takedown.*

(2) **Families:** Several P2P botnets have been deployed and detected for several years. Figure 2.6 illustrates the history developments of P2P botnets since 2007 until 2012 [46].

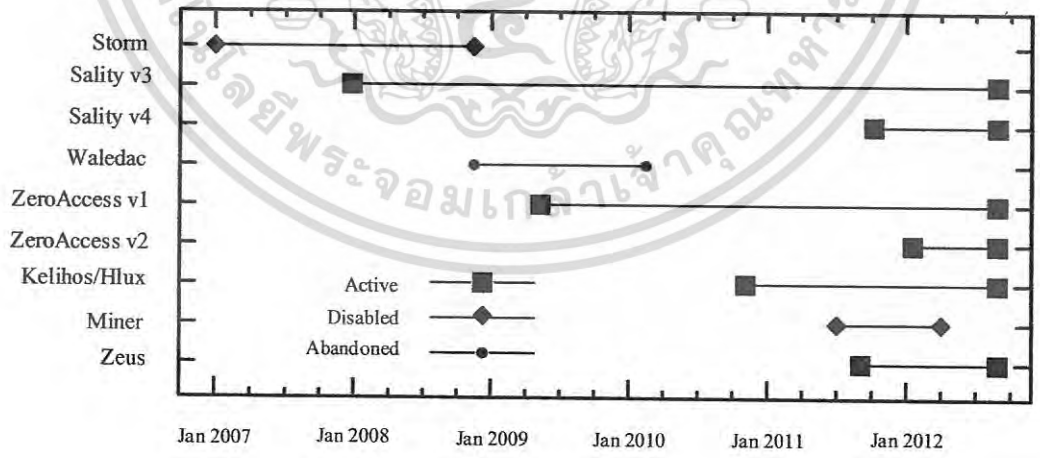

Figure 2.6 History development of P2P botnets [7]

Table 2.1 describes the advantages or purposes of owning a botnet for cybercriminals used by well-known bots [47].

– **Nugache** is an email, P2P, and IM worm. It relates to two other botnets, Storm and

Table 2.1 P2P botnet families of evolution

Name	Released	Threats
Nugache	2006	Theft of financial credentials via keystroke logging
Storm	2007	Overnet, MS OS via spam
Sality P2P	2008	Pay per install
Waledac	2008	Email spam
Miner	2009	Query neighbors for new malware
Kelihos	2010	Spamming and ID theft
ZeroAccess	2011	Generating a digital currency
Zeus P2P	2011	Steal financial credentials

Rizo. The Nugache worm can spread via email, using a variety of subject lines and message text from lists contained in the worm's code. It may also spread via AOL Instant Messenger or Windows Messenger. It sends IM contacts a link pointing to a copy of itself. According to antivirus vendor McAfee, the names of the files pointed to may include one of the self nude.scr, my pic.sc. The Nugache worm opens a back-door on TCP port 8, attempts to connect to a specific IRC server, and awaits remote commands from the worm's author.

- **Storm** (a.k.a. Peacomm) was a structured P2P botnet using the Overnet protocol, a Kademlia implementation. In fact, the first version of Storm used an existing Overnet network, and the bots added themselves to the existing DHT (distributed hash table). In Storm, botmasters stored spam templates at deterministically computable IDs in the DHT. In turn, the bots requested these commands by looking up the computable IDs. Storm was significantly disrupted in 2008.
- **Sality P2P** appeared with its second version in 2008 and is a variant of the centralized Sality malware downloader. Sality uses an unstructured P2P network in a pull-based manner. Peers regularly check if their neighbors promote previously unseen files, and if so, they download and install these files. Two separate Sality networks consist of peers only with version 3 or version 4, respectively. Both networks share the same P2P protocol and differ mainly in the file downloading mechanism.
- **Waledac** is assumed to be the successor of Storm. Waledac had centralized peers in its upper layer, which served spam templates to peers in the lower hierarchy. The lower peers form the majority in the network and were connected via an unstructured P2P network. The lower peers exchanged lists of peers in the upper layer using pull-based communication. In February 2010, Waledac was sink holed via manipulated peer lists

by Stock et al.

- **ZeroAccess** (a.k.a. Sirefef) is a malware downloader with an unstructured P2P C&C architecture. It exists in at least two variants and is organized in at least seven disjoint networks. The older variant, ZeroAccess v1, is pull-based, which bots regularly consult their neighbors (peers) for malware installation instructions. The newer variant, ZeroAccess v2, uses both push- and pull-based communication. New peers are broadcast to the network (push), whereas the malware download commands are requested from other peers (pull).
- **Miner** was an unstructured P2P botnet that included three libraries for mining Bitcoins (a digital currency). The Miner botnet consisted of two almost disjoint networks with about 38,000 non-NATed peers according to Kaspersky. According to Crowdstrike, Miner ceased to operate around 03/2012, presumably due to insufficient monetization of mining Bitcoins.
- **Kelihos/Hlux** is tactic of using P2P communication rather than a centralized command and control server or servers also contributes to its staying power. This P2P botnet is resilient against not only law enforcement, but also security analysts who want to enumerate these networks of compromised computers or disrupt their services. At RSA Conference 2013, CrowdStrike researcher Tillmann Werner did a live takedown of Kelihos on stage during a presentation. He managed to poison a middle layer of P2P proxy servers that communicate with the attacker by writing a sinkhole daemon that behaved like a bot. The daemon would send poisoned peer lists to the other bots it communicated with, specifically blacklisted sets of IP addresses, sending them toward a sinkhole and oblivion.
- **Zeus P2P** is an unstructured P2P network having a pull-and push-based command architecture. Zeus's configuration files are pulled from peers with more recent versions, containing, for example, browser hooks used to steal personal data. Drop zone locations for receiving the stolen data are pushed via gossiping.

Reviews of bot/malware clustering and country clustering are given here.

2.2 Bot/Malware Clustering

Malware clustering has been lately studied by several researchers and can be divided into *Pre-Labeled* and *Post-Labeled* approaches based on similar Host based and Network based features as summarized in Table 2.2. Note that the features in each quadrant are ordered in a top-down fashion.

Table 2.2 Summary of Clustering Approaches (Pre-Labeled & Post-Labeled) vs Clustering Features (Host Based & Network Based).

Behaviors	Pre-Labeled Approaches	Post-Labeled Approaches
Network Based	HTTP, SMTP, IRC Oriented [22] & [30] C&C traffic [31] & DNS Oriented [28] Network Flow Oriented [24, 48] & [27] IP Network Layer Oriented [29]	Daily and Hourly Malware Download [34] HTTP Oriented ([23] DNS Oriented [37] SrcIP/DstPort [32]
Host Based	Dynamic System Call, File, Control Flow [22] Dynamic File, Registry, Process [29] Dynamic System Call Order [26] Static Portable Executable [25] Malware Binaries [30]	AntiVirus Label Graph [33] Dynamic File, Registry, Process [32] Static Packed Portable Executable [35] Malware Binaries [36] & [38]

2.2.1 Pre-Labeled Approaches

Pre-Labeled approaches utilize features from both host and network based features to cluster malware identified by unique hash values based on similar behaviors/distance metrics. The following approaches are reviewed in chronological order.

Initially in 2006, Pouget et al. [24] introduced a new notion, namely *cliques of clusters*, as an automated knowledge discovery method for attack traces from Leu-rre.com dataset (network) of 40 honeypot sensors. These sensors had been active for more than 12 months in 25 countries. They clustered the traffic with a number of features in order to achieve eight different distance matrices based on an algorithm detailed in [48]. As a result of experimental validation, Sasser and Dabber worms could be identified by clustering *A_CommonIPs* matrix.

In 2009, Wicherski [25] introduced a novel malware clustering based on a generic hash function for Portable Executable files. His practical evaluation on different malware sets (including the malware samples collected by Nepenthes sensors) shows a significant reduction of sample counts.

In the same year, Apel et al. [26] presented a technique for clustering malware behavior based on the sequence of dynamic system calls executed by the malware samples. To compare and group malware traces with using the single-linkage hierarchical clustering algorithm, several

appropriate distance metrics have been investigated, including Edit Distance, Approximated E-dit Distance, Normalized Compression Distance, and Manhattan Distance Using Tries.

Bayer et al. [22]'s clustering algorithm is based on locality sensitive hashing (LSH) of malware program file, registry key, operations, e.g. read, write, create, and network activities. Based on LSH, they were able to compute single-linkage hierarchical clustering of more than 75 thousand malware samples (obtained from AN-UBIS) in less than three hours.

In 2011, Lu et al. [27] clustered malware generating the anomalous traffic. Their approach applied three different clustering algorithms for botnet detection; K-means, unmerged X-means, and merged X-means. Evaluation on IRC community can successfully detect two IRC botnet traffic traces with a high detection/classific-ation rate and a low false positive rate.

Due to high computation demands, a light-weight mechanism called BotGAD (Botnet Group Activity Detector) was proposed by Choi and Lee [28]. It is based on a small number of data from DNS traffic without all the traffic contents or known signatures to detect botnets. BotGAD can automatically detect botnets in large scale network providing over 95% detection rates while generating less than 0.4% false positive rates and 5% false negative rates.

Late 2012, Chandramohan et al. [29] presented a scalable malware behavior modeling technique that models the interactions between malware and sensitive system resources (e.g. File, Registry, Process, and Network) in order to perform malware clustering. They used F-Measure to evaluate the clustering accuracy of their approach. Their experiment result improves the average clustering accuracy by 6.20% while reducing the feature spaces by 289 times against a state-of-the-art malware clustering technique.

In 2013, Rafique et al. [30] presented FIRMA, a tool that given a large pool of network traffic (e.g., HTTP, IRC, SMTP, TCP, UDP) obtained by executing unlabeled malware binaries, generates clusters of malware binaries. They have implemented FIRMA and evaluated it on two recent datasets comprising nearly 16,000 unique malware binaries. Their results show very high Precision and Recall.

Recently in 2015, Barthakur et al. [31] proposed CluSiBotHealer, a novel framework for detection of P2P botnets through data mining technique. EM (Expectation Maximization) clustering algorithm is used to cluster C&C flows based on Jaccard Similarity Coefficient. Three different C&C flows namely Nugache, Waledac and Zeus can be identified with good Accuracy, Sensitivity (Recall) and Positive Predictive Value (Precision).

2.2.2 Post-Labeled Approaches

On the other hand, Post-Labeled clustering approaches rely on features extracted directly/indirectly from *labeled* (named) malware detected by anti-viruses or detection systems. The following approaches are reviewed in chronological order as well.

Bailey et al. [32] proposed an automated malware classification and analysis based on a behavioral fingerprint of the malware's activities. The fingerprint is a set of system changes due to malware execution including files modified, processes created (host based) and network connections (network based), i.e., connects to port 80, scans port 80, and so on. Single-linkage hierarchical clustering is applied by using normalized compress distance as a distance metric. 3,700 malware samples are collected in a six-month period and detected by major anti-viruses; McAfee and Trend Micro, and compared with the clusters generated by existing malware classification.

Perdisci et al. [23] presented a network-level behavioral malware clustering system that focuses on HTTP-based malware and clusters malware samples based on a notion of structural similarity between the malicious HTTP traffic. They applied single-linkage hierarchical clustering algorithm with more than 25,000 malware samples in a period of six months from a number of different malware to confirm the effectiveness of the proposed clustering system.

Two years later, Perdisci and U [33] proposed VAMO system to provide a fully automated assessment of the quality of malware clustering results. Based on AntiVirus Label Graph according to existing AntiVirus scanners, namely McAfee, Avira, and Trend Micro. By applying average-linkage hierarchical clustering, VAMO performs better than majority voting-based approaches, and provides a better way for malware analysts to automatically assess the quality of their malware clustering results.

In 2013, Yukonhiatou et al. [34] made use of both daily and hourly malware downloads correlation coefficients to cluster malware/bots download behavior of Top-10 malware (given by anti-virus, Trend Micro) based on 2010 [49] (with more than one million download logs), and 2011 [50] (with almost two hundred thousand download logs) CCC datasets. Their approach can cluster 3 and 4 groups of Top-10 malware/bots in 2010 and 2011, respectively.

On the host based side, Hu et al. [35] presented a comprehensive malware clustering system based on static features of unpacked/disassembled/extracted from packed portable executable. The experiment result shows that their approach can process a database of more than 130,000 malware samples to correctly cluster over 80% within a few hours, achieving a good balance between accuracy and scalability.

In 2014, Biggio et al. [36] proposed a clustering method focusing on behavioral malware (malware binaries) and investigate whether and to what extent an attacker may be able to subvert these approaches through a careful injection of sample with poisoning behavior. Their single-linkage hierarchical clustering results clearly show that even a small fraction of 3% of poisoning samples may completely subvert the clustering process, leading to poor clustering results.

In the same year of 2014, Thomas et al. [37] relied on similarity of domain name system (DNS) traffic, such as non-existent domain queries at several premier top level domain authoritative name servers. They applied single-linkage hierarchical clustering with $O(n^2)$ complexity, where n is the number of domains.

Recently in 2015, Narra et al. [38] applied clustering analysis to the challenging problem of classifying previously unknown malware families. Based on the Hidden Markov Model (HMM) scoring samples of malware families, the malware samples are grouped into clusters using K-means and EM clustering algorithms. The new malware families can be classified based on clustering results of a set of previously-known malware families.

2.3 Country Clustering

To illustrate that country clustering plays a critical role in malware spreading, the notion of source IP's country clustering has been mentioned in [24] as *A_Geo* matrix since 2006. They reported that very limited number of countries hosted frequent attacks to their honeypot sensors. However, some malware tends to target particular geographical regions, corresponding to different market segments for vulnerable software (i.e., a language edition of an operating system) [51] in 2006 as well.

In 2013, Sisaat et al. [52] proposed a simple source country clustering method based on *Time Zone Correlation* of hourly malware downloads in order to locate the source countries of C&C server(s). Their approach is based on 2010 CCC dataset [49]. Their analysis shows that botnet compromised hosts are located in two country groups; group *J*: malware downloads are synchronized with C&C servers located in Japan, while group *L*: malware downloads are synchronized with local C&C servers' time zones.

Based on the Symantec World Intelligence Network Environment (WINE) Intrusion Prevention System (IPS) telemetry data, Mezzour et al. [53] concluded in 2014 that some developed countries are major targets of exploits, web attacks and fake applications to gain computing and monetary resources. In addition, Eastern Europe hosts attack computers due to good computing infrastructure and high corruption rate.

Recently in 2015, Asghari et al. [9] reported a 6-year long dataset of Conficker. Sixty two countries infected by Conficker were characterized by growth, peak and decay rates of infections. The high growth rate of Conficker in a particular country corresponds to both low Information and Communication Technology (ICT) development and high software piracy rates.

However, the existing approaches were targeted at different levels of characteristics of malware ranging from binary contents (physical) to network (temporal) behaviors but somewhat limited to geographical locations.

2.4 Clustering Algorithms

Data clustering is an unsupervised learning problem, where the learning algorithms task is to group samples into clusters based on unlabeled features. This allows us to find similarities and differences among samples and to derive useful conclusions about them [54], [55]. There are specific steps when applying a clustering scheme. Here, six basic steps are needed:

In order to prepare the features for the clustering algorithm we need to perform some (1) *preprocessing*, for example, converting to a supported feature type. Also, by applying (2) *feature selection* techniques it will ensure that the chosen features are highly relevant for the task of interest. A (3) *proximity measure* is applied in order to decide how similar (or dissimilar) two feature vectors are. Additionally, defining what type of clusters that are most sensible for the underlying dataset is stated by a (4) *clustering criterion*. Next step involves applying the (5) *clustering algorithm*, for example, partitional or hierarchical clustering algorithm. Finally, the clustering outcome is (6) *validated* and evaluated to verify its correctness.

2.4.1 Partitional Clustering

In partitional clustering, samples are grouped into partitions, as can be seen in Figure 2.7 (a). Depending on the specific algorithm, for example, K-means clustering [56], the number of clusters needs to be specified beforehand. This is the general disadvantage with partitional clustering, since the number of clusters will influence the clustering outcome. Thus, it may be necessary to apply the same clustering algorithm several times and evaluate the results to find the best number of clusters that suit the specific task best.

2.4.2 Hierarchical Clustering

There are two types of hierarchical algorithms, namely, bottom-up (agglomerative) and top-down (divisive). A standard representation of hierarchical clustering is with a dendrogram as indicated in Figure 2.7 (b), which is a binary tree where the leaves correspond to each individual

sample and the root corresponds to all the samples. When a button-up algorithm is executed it starts with the individual samples and iteratively merges the most similar ones. On the other hand, top-down algorithms starts with all samples in one cluster and iterative splits them into smaller clusters [57].

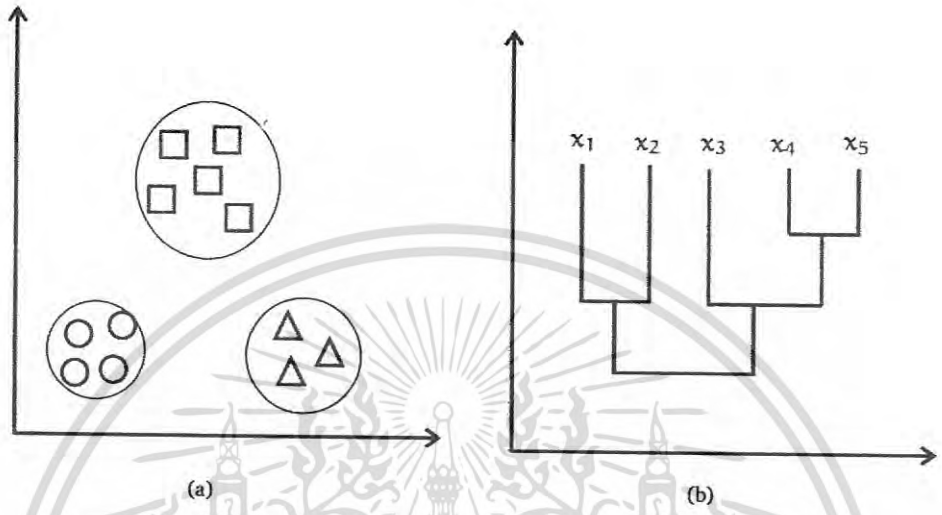

Figure 2.7 (a) Partitional clustering, (b) Hierarchical clustering (Source: Master Thesis [8])

2.5 2010 CCC and 2012 IIJ Datasets

2.5.1 Overview of CCC

Cyber Clean Center or CCC has been established by MIC (Ministry of Internal Affairs and Communications) and METI (Minister of Economy, Trade and Industry) for the purpose of reducing the number of botnet-infected computers to zero.The CCC is a five-year project from fiscal year 2006 to 2010. Totgether with 76 ISPs, 7 anti-virus software vendors and research institutes were working together to promote anti-bot activities in April 2010.

CCC is active in analyzing characteristics of bots, which have been a threat against the Internet, and providing information on disinfestation of bots from users' computers. In addition, the CCC is a core organization playing a major role to promote bot cleaning and prevention of re-infection of users' computers, which are once infected by bots, based on cooperation with ISPs (Internet Service Providers).

Under the Cyber Clean Center-Steering Committee (CCC-SC), CCC consists of three groups covering different purposes and conducting daily activities as shown in Figure 2.8.

1. Bot Countermeasure System Operation Group (Telecom ISAC Japan): The Bot Counter-

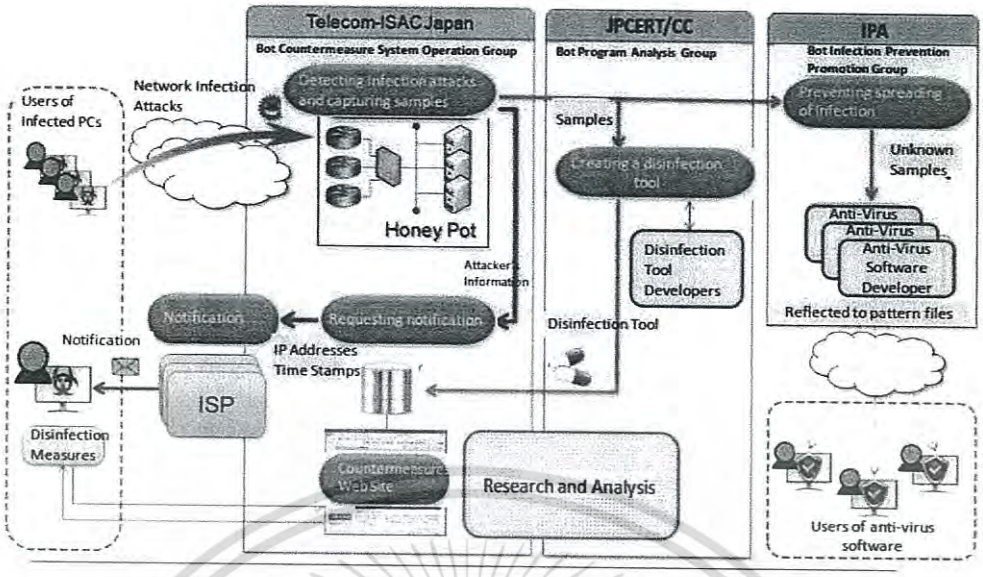

Figure 2.8 General workflow of CCC

measure System Operation Group operates the main system of this project, including the Honeypot System and Warning System to collect and analyzes bots, and notifies users of bot-infected PCs through the ISPs participating in the project. With the aim of countering the new thread of bots and implementing effective measure, the group collaborates with security vendors to conduct surveys on the latest malware threats.

2. Bot Program Analysis Group (JPCERT Coordination Center): The Bot Program Analysis Group analyzes the characteristic and technology of the bot samples collected by the Bot Countermeasure System Operation Group. This group works with disinfection tool developers to provide the CCC Cleaner disinfection tool. They also study effective analysis methods and coordinate with security vendors to develop countermeasure technologies.

3. Bot Infection Prevention Promotion Group (Information-Technology Promotion Agency, Japan): The Bot Infection Prevention Promotion Group maintains bot samples collected by the Bot Countermeasure System Operation Group. The samples are quickly provided to security vendors so that they can reflect them in the creation of pattern files. Therefore, the users of anti-virus software can disinfect unknown bots before their infection spreads. This group promotes infection prevention by reducing the risk of infection.

2.5.2 Overview of IIJ

Internet Initiative Japan, Inc. or IIJ is a telecommunications company based in Tokyo, Japan. Established on December 3, 1992, it employs 1,715 people to provide Internet connec-

tivity and WAN services, network-related services, network systems construction, operation and maintenance, development and sales of telecommunication equipment [58].

The Malware Investigation Task Force or MITF uses honeypots that emulate the behavior of functions such as Windows File Sharing (SMB) and RPC to monitor the activity of malware that attempts to exploit vulnerabilities. Because honeypots are simply connected to the Internet and do not communicate actively, attempts to communicate with them are likely to be either attacks or precursors to attacks.

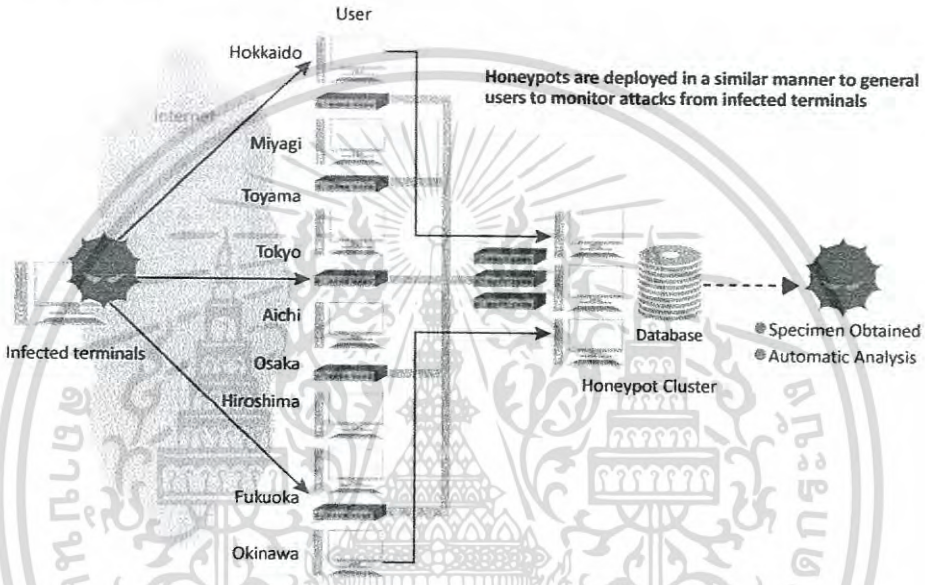

Figure 2.9 IIJ MITF Honeypots

This research work relies the 2012 IIJ MITF dataset [58], which observes malware traffic on 100 independent honeypots. Figure 2.9 illustrates overall architecture of the IIJ MITF honeypots in Japan. These honeypots are virtual hosts running Windows XP SP2 as operating system with vulnerabilities and no human interactions. The initial infections automatically force these honeypots to download secondary injections and upgrades from compromised hosts [59].

IIJ MITF dataset is one of the datasets provided with Anti-Malware Research and Research Achievements Shared at the Workshop (MWS 2012) [60] similar with the CCC datasets in MWS2009 [61], MWS2010 [49], and MWS2011 [50].

2.5.3 Log Structure and Datasets

This dissertation relies the 2010 CCC dataset, which investigates more than 90 independent Honeypots to observe malware traffic on the Japanese tier-1 backbone network. Figure 2.10 is the overall architecture of our experimental setup for this research work. A Honeypot is a virtual host

running Windows XP as an operating system with vulnerabilities, which is rebooted periodically in order to avoid an infection from being active for a long time. In other words, time to reboot is negligible short and hence a Honeypot is supposed to be always online.

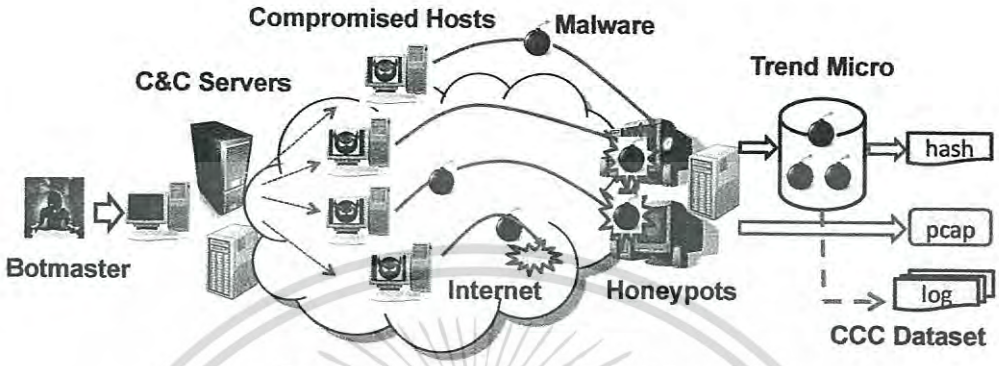

Figure 2.10 Experimental setup for 2010 CCC dataset

The CCC coordinates the observations. 2010 CCC dataset comprises the access logs of the botnet attacks between May 1, 2009 and April 30, 2010 Japan fiscal year. Each Honeypot records every packet as an item in an access log comprising Timestamp, Honeypot ID, Source/Destination IP address, Source/Destination port number, Hash value (SHA1: Secure Hashing Algorithm Version 1.0), Malware name, and Malware file name.

Table 2.3 shows the summary and comparison of 2010 CCC and 2012 IIJ MITF datasets, which were extracted. According to the datasets, full record logs with 92 Honeypots for 2010 CCC dataset, and 10 months with 100 Honeypots for 2012 IIJ MITF dataset.

Table 2.3 Summary of malware downloads in 2010 and 2012

Details	2010	2012
Time period	52 Wks (12 Mths)	44 Wks (10 Mths)
# of honeypots	92	100
# of records (including unknown malware)	1,162,093	32,070,143
# of unique IPs (including honeypots)	176,981	2,322,224
Based on TCP	1,053,977	32,070,143
Based on UDP	108,116	0
Unique hash values	29,858	20,743
Unique malware names (excluding unknown malware)	979	456
Anti-Virus Scanner	Trend Micro	ClamAV

All of the logs in this research work are stored in binary format. Two datasets have been identified; 2010 CCC and 2012 IIJ MITF datasets. Table 2.4 shows the log structure in CCC and IIJ datasets detected by Trend Micro AntiVirus and ClamAV scanner, respectively. It consists of

9 fields in the CCC log file, and 10 fields in the IIJ log file. Most common fields in both log file is separated by a comma "," which indicated the patterns of malware download behavioral activities.

Table 2.4 Sample of logged structure in CCC dataset 2010 & 2012 IIJ MITF dataset

Column	Logged items	Example	Remark
1	Timestamp	2010-03-05 03:02:41	
2	Source IP address	*.*.166.195/honey060	
3	Source port No.	1028	
4	Destination IP address	*.*.243.167/honey032	
5	Destination port No.	5824	
6	Protocols	TCP/UDP	
7	Hash value (SHA1)	***bc3c8***	
8	Malware name	WORM_DOWNAD.AD	
9	File name	C:\WINNT\system32\dhnlr.dll	
10	File type	PE32 executable for MS Windows	Only 2012 IIJ
11	Vulnerability	MS08-67	Only 2012 IIJ
12	Path to download malware	http://*.*.166.195:5229/fqqxttq	Only 2012 IIJ

- **Timestamp:** stores the phenomenon time (ordering by year, month, date, hour, minute and second. For example, 2009-05-01 00:01:05) of malware download among C&C Server and Honeypots.
- **SRC IP:** stores the source IP addresses of malware. This will be occurred upon the malware download between remote Server and Honeypots activities.
- **SRC Port:** stores the source ports of malware originating. Actually, there are in the range of registered ports, private ports, ports 1024 through 49151 and ports 49152 through 65535, respectively.
- **DST IP:** stores the target IP addresses of malware or victim Hosts (most of the them are Honeypots ID).
- **DST Port:** stores the vulnerabilities ports of target Hosts that use for communicating /downloading of malware.
- **Protocols:** stores the protocols used by malware (tcp, udp).
- **Hash value (SHA-1):** stores the hash values, which generated by antivirus scanner according to malware detected. And there are variants for each unique malware (a single malware has multiple hash values)

- **Malware name:** stores the malware names, which derived from the malware signature used by commercial anti-virus software Trend Micro and ClamAV in 2010 and 2012, respectively.

- **File name:** stores the file names of malware. Each file name can be unique.

CHAPTER 3

PROPOSED METHOD AND TOOLS

This chapter will take a more elaborate look at millions of binary (malware) download logs which were recorded in Japan.

3.1 Tools and facilities

2010 CCC and 2012 IIJ MITF datasets contain millions of malware download records uniquely identified by Timestamp, Source IP address/port number, Destination IP address (Honeypot ID)/port number, Hash value (SHA1), Malware name, and Malware file name as mentioned in Chapter 2. Based on a huge amount of logged information of malware download activities in the datasets, the total downloads of malware can be explored and enumerated on day d and the total downloads of malware at hour h, using some powerful tools such as **R** application and some Linux commands shuch as **cat, grep, awk**, and **sed** as shown in Figure 3.1 to obtain Top-20/30 malware, Top-20/30 countries, and so on as follows.

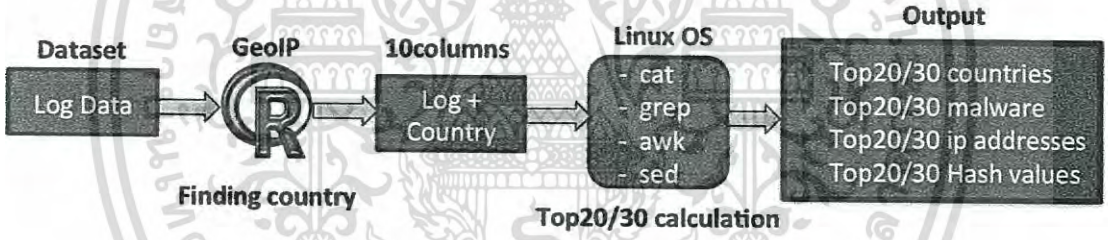

Figure 3.1 Top-20/30 processing of 2010 and 2012 datasets

- **R Application:** R is a free software environment for statistical computing and graphics. It is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes
  - An effective data handling and storage facility,
  - A suite of operators for calculations on arrays, in particular matrices,
  - A large, coherent, integrated collection of intermediate tools for data analysis,
  - Graphical facilities for data analysis and display either on-screen or on hardcopy, and
  - A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

- **GeoIP:** implements a lookup using the InfoDB API to determine point coordinates for a given IP address. GeoIP(x), x is the IP address in IPv4 format. A list is returned containing the elements as summarized in Table 3.1.

Table 3.1 Structure of IPv4 GeoIP Results

Element	Descriptions
IPaddress	Input IP address
statusCode	Returned status code from lookup
latitude	Point coordinate-latitude
longtitude	Point coordinate-longtitude
statusMessage	Returned status message from lookup
countryCode	Country code from IP lookup
countryName	Country name from IP lookup
regionName	State/region/province from IP lookup
cityName	City from IP lookup
zipCode	Postal code from IP lookup
TimeZone	Timezone from IP lookup

The imported initial logs in to R are looked up the countryName using GeoIP function based on the malware sources IP address. Furthermore, the log data are processed for the next processing as indicated in Figure 3.1

- **cat:** is a Unix and Linux command, concatenate FILE(s), or standard input. cat program is given files in a sequence as arguments; it will output their contents to the standard output in the same sequence. To extract any patterns of malware from the entire logs data 2010 and 2011, cat command can merge all monthly logs data of each year into single file and continue next step by using awk command.

- **awk:** the AWK language is useful for manipulating data files, text retrieval and processing, and for prototyping and experimenting with algorithms. An AWK program is a sequence of pattern action pairs and function definitions. This command extracts and counts all unique malware names, IP addresses, countries, etc., including number duplicated of them. From this, another subcommand is called such as sort, head, and especially is grep to arrange the obtained output into Top-malware/country.

- **grep:** searches the input file name(s) for lines containing a match to the given PATTERN. By default, grep prints the matching lines. Hence, this command is used for two main purposes: First, search for quantities of unique malware containing in logs data. Second, create and save the greped lines into a new specific file for each Top-malware/country.

Furthermore, this command can be combined with others such as sed to correct the data format before computing in R Project.

- **sed:** a stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline). The command sed can filter texts and symbols in a pipeline that particularly distinguishes of some output files.

3.2 Supervised Clustering Algorithm

This dissertation proposes a combination of malware features in terms of temporal and spatial (country) download behaviors. Both of them can characterize each malware better thus yield sooner malware prevention/mitigation.

The malware clustering method falls into the Post-Labeled (a number of unique hash values have been labeled and classified with malware names) category as detected by ClamAV. It differs from previous network based approaches due to the country, weekly, and hourly download feature vectors as depicted in Figure 3.2. In addition to malware clustering, the by product country clustering can summarize malware spreading behaviors in particular geographical regions. The country clustering method may be able to predict future botnet propagation characteristics, for those botnets using similar vulnerabilities (i.e., regional viruses or worms). This section explains how to obtain the Spatio-Temporal features and clustering method in details.

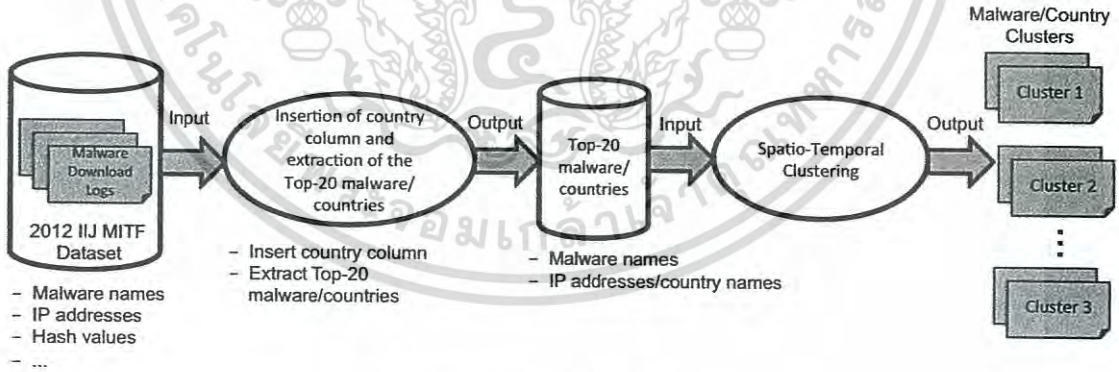

Figure 3.2 System Overview of the Spatio-Temporal Malware/Country Clustering.

To ease the presentation of how the statistical features are computed, notations are first introduced and used throughout this paper as provided in Table 3.2.

The download behaviors of the malware in 2010 CCC and 2012 IIJ MITF datasets can be investigated. To analyze the download logs of each malware to the Honeypots, the temporal and spatio behaviors in terms of number of downloads per day/per hour of a given dataset can be

Table 3.2 Notations (#DLs: Number of Downloads, Norm: Normalized).

Notation	Description
$l_m(p, e, k)$	#DLs of Top-p country on day e, in hour k of Top-m malware
$l_c(n, e, k)$	#DLs of Top-n malware on day e, in hour k from Top-c country
$l_c^h(k)$	#DLs in hour k from Top-c country
$l_c^m(n)$	#DLs of Top-n malware from Top-c country
$l_c^w(u)$	#DLs in week u from Top-c country
$l_m^h(i)$	Total hourly DLs of Top-m malware in hour i
$l_m^h(k)$	#DLs of Top-m malware in hour k
$l_m^c(p)$	#DLs of Top-m malware from Top-p source country
$l_m^w(u)$	#DLs of Top-m malware in week u
$l_n^c(p)$	#DLs of Top-n malware from Top-p country
$l_p^h(i)$	#DLs from Top-p country in hour i
$l_p^m(n)$	#DLs of Top-n malware from Top-p country
$l_c'^h(k)$	Norm #DLs in hour k from all Top-20/30 countries
$l_c'^m(n)$	Norm #DLs of Top-n malware from all Top-20/30 countries
$l_m'^c(p)$	Norm #DLs of all Top-20/30 malware from Top-p country
$l_m'^h(k)$	Norm #DLs of all Top-20/30 malware in hour k
$Ł_m$	Spatio-Temporal feature vector of Top-m malware
$Ł_c$	Malware-Temporal feature vector of Top-c country
$Ł_m^c$	Country DL vector of Top-m malware
$Ł_m^h$	Hourly DL vector of Top-m malware
$Ł_m^w$	Weekly DL vector of Top-m malware
$Ł_c^h$	Hourly DL vector from Top-c country
$Ł_c^m$	Malware DL vector of Top-c country
$Ł_c^w$	Weekly DL vector of Top-c country

formulated. Thus, clustering features can be divided into malware clustering features and country clustering features as described in following subsections. However, the variation of malware downloads can be too large to compare. Therefore, the normalization of the number of malware downloads should be investigated.

3.2.1 Malware Clustering Features

Malware clustering features include weekly download, hourly download, and country download vectors. These features can be obtained from the following definition.

Definition 1: Let $l_m(p, e, k)$ be number of Top-m malware downloads from Top-p source country, on day e and in hour k, respectively. The country p can be derived from source IP address of each record with the GeoIP database [62]. Therefore, *the Spatio-Temporal feature vector of malware download logs can be expressed consecutively.*

A Spatio-Temporal feature vector, $Ł_m$ of Top-m malware is a row vector consisting of weekly download, hourly download and country download vectors as follows:

$$Ł_m = \left[Ł_m^w \quad Ł_m^h \quad Ł_m^c\right]. \tag{3.1}$$

The weekly download vector of Top-m malware is

$$Ł_m^w = \left[l_m^w(u)|_{u=1}^{52} \cdots\right] \tag{3.2}$$

where u is the weekly number ranging from 1 to 52 per year. Based on Definition 1, the number of downloads of Top-m malware in weekly u from all Top-20/30 source countries can be derived as

$$l_m^w(u) = \sum_{e=7(u-1)+1}^{7u} \sum_{k=0}^{23} l_m(p, e, k), \quad \forall p. \tag{3.3}$$

Similarly, the hourly download vector of Top-m malware is

$$Ł_m^h = \left[l_m^h(k)|_{k=0}^{23} \cdots\right] \tag{3.4}$$

where k is the hour number in each day ranging from 0 to 23. The number of downloads of Top-m malware in hour k from all Top-20/30 source countries is

$$l_m^h(k) = \sum_{e=1}^{365} l_m(p, e, k), \quad \forall p. \tag{3.5}$$

Finally, the malware country download vector of Top-m malware from each Top-p country is

$$Ł_m^c = \left[l_m^c(p)|_{p=1}^{20/30} \cdots \right] \tag{3.6}$$

where p is 1 to 20/30. Hence, the number of downloads of Top-m malware from Top-p source country is

$$l_m^c(p) = \sum_{e=1}^{365} \sum_{k=0}^{23} l_m(p, e, k). \tag{3.7}$$

3.2.2 Country Clustering Features

Country clustering features include weekly download, hourly download, and malware download vectors. These features can be obtained from the following definition.

Definition 2: Let $l_c(n, e, k)$ be number of Top-n malware downloads from Top-c country, on day e and in hour k, respectively.

A Malware-Temporal feature vector $Ł_c$ of Top-c country is a row vector consisting of weekly download, hourly download and malware download vectors as follows:

$$Ł_c = \left[Ł_c^w \quad Ł_c^h \quad Ł_c^m \right]. \tag{3.8}$$

The weekly download vector of Top-c country is

$$Ł_c^w = \left[l_c^w(u)|_{u=1}^{52} \cdots \right] \tag{3.9}$$

where u is the weekly number ranging from 1 to 52 per year. Based on Definition 2, the number of downloads of Top-c country in weekly u from all Top-20/30 malware can be derived as

$$l_c^w(u) = \sum_{e=7(u-1)+1}^{7u} \sum_{k=0}^{23} l_c(n, e, k), \quad \forall n. \tag{3.10}$$

Similarly, the hourly download vector from Top-c country is

$$Ł_c^h = \left[l_c^h(k)|_{k=0}^{23} \cdots \right] \tag{3.11}$$

where k is the hour number in each day ranging from 0 to 23. The number of downloads of Top-c

country in hour k from Top-20/30 malware is

$$l_c^h(k) = \sum_{e=1}^{365} l_c(n, e, k), \quad \forall n. \tag{3.12}$$

Finally, the malware country download vector of Top-c country from each Top-n malware is

$$Ł_c^m = \left[l_c^m(n)|_{n=1}^{20/30} \cdots \right] \tag{3.13}$$

where n is the malware number 1 to 20/30. The number of downloads of Top-c country from malware n is

$$l_c^m(n) = \sum_{e=1}^{365} \sum_{k=0}^{23} l_c(n, e, k). \tag{3.14}$$

3.2.3 Dissimilarity

In this dissertation, the hierarchical clustering with different dissimilarity and linkage options is utilized and compared to achieve more balanced clusters (with equal diameter) and be less susceptible to noise. Both Top-20/30 malware and Top-20/30 countries can be clustered according to Correlation Dissimilarity and Cosine Dissimilarity as follows.

The Malware Correlation Dissimilarity between Top-m and Top-m' malware can be obtained as

$$\mathbf{D_{cor}}(Ł_m, Ł_{m'}) = 1 - \frac{\mathbf{Cov}(Ł_\mathbf{m}, Ł_{\mathbf{m'}})}{\sqrt{\mathbf{Var}(Ł_\mathbf{m})\mathbf{Var}(Ł_{\mathbf{m'}})}} \tag{3.15}$$

where $\mathbf{Cov}(Ł_m, Ł_{m'})$ and $\mathbf{Var}(Ł_m)$ are the covariance and variance of Spatio-Temporal feature vectors, $m = 1, ..., 19/1, ..., 29$, and $m' = m + 1$, respectively.

On the other hand, the Country Correlation Dissimilarity between Top-c and Top-c' countries can be obtained as

$$\mathbf{D_{cor}}(Ł_c, Ł_{c'}) = 1 - \frac{\mathbf{Cov}(Ł_\mathbf{c}, Ł_{\mathbf{c'}})}{\sqrt{\mathbf{Var}(Ł_\mathbf{c})\mathbf{Var}(Ł_{\mathbf{c'}})}} \tag{3.16}$$

where $\mathbf{Cov}(Ł_c, Ł_{c'})$ and $\mathbf{Var}(Ł_c)$ are the covariance and variance of Malware-Temporal feature vectors, $c = 1, ..., 19/1, ..., 29$, and $c' = c + 1$, respectively.

Similarly, the Malware Cosine Dissimilarity between Top-m and Top-m' malware can be obtained as

$$\mathbf{D_{cos}}(Ł_m, Ł_{m'}) = 1 - \frac{Ł_\mathbf{m} \cdot Ł_{\mathbf{m'}}}{||Ł_\mathbf{m}|| ||Ł_{\mathbf{m'}}||} \tag{3.17}$$

where $Ł_m, Ł_{m'}$ and $||Ł_m||, ||Ł_{m'}||$ are the vector and magnitude of Spatio-Temporal feature vectors, $m = 1, ..., 19/1, ..., 29$, and $m' = m + 1$, respectively.

On the other hand, the Country Cosine Dissimilarity between Top-c and Top-c' countries can be obtained as

$$\mathbf{D_{cos}}(Ł_c, Ł_{c'}) = \mathbf{1} - \frac{Ł_c \cdot Ł_{c'}}{||Ł_c||||Ł_{c'}||} \tag{3.18}$$

where $Ł_c, Ł_{c'}$ and $||Ł_c||, ||Ł_{c'}||$ are the vector and magnitude of Malware-Temporal feature vectors, $c = 1, ..., 19/1, ..., 29$, and $c' = c + 1$, respectively.

3.2.4 Evaluation and Flow Chart

To evaluate the effectiveness of proposed malware/country clustering with two Dissimilarities and different linkage options, P (Precision) and R (Recall) can be computed based on Reference Data as follows.

$$P = \frac{TP}{TP + FP} \tag{3.19}$$

and

$$R = \frac{TP}{TP + FN} \tag{3.20}$$

where TP, FP and FN are True Positive, False Positive and False Negative, respectively.

The proposed Spatio-Temporal Clustering method can be described as a flowchart in Figure 3.3. The Hierarchical Clustering with (Single, Average and Complete) Linkage option is applied. Given the maximum Dissimilarity value, the resulted cluster (dendrogram) can be compared with Reference Data such that Presicion (P) and Recall (R) are computed. In order to achieve better P and R results, the algorithm gradually decreases the Dissimilarity value resulting in a range of Dissimilarity values with the same dendrogram. The method stops until the best P and R have been found.

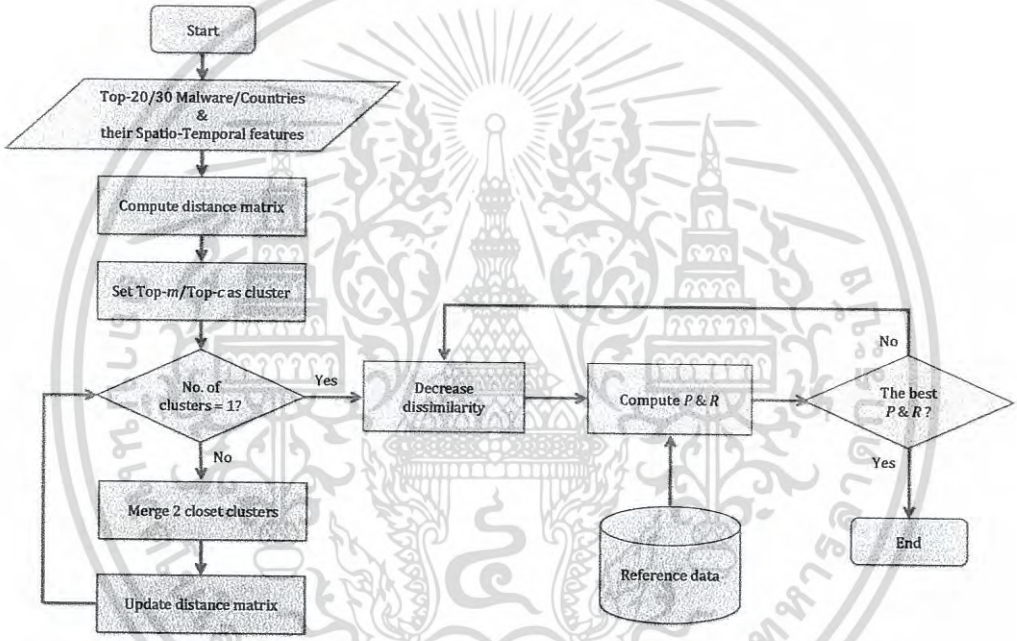

Figure 3.3 Flowchart of Hierarchical Spatio-Temporal Supervised Clustering Algorithm.

CHAPTER 4

RESULTS AND DISCUSSIONS

This section is divided into three subsections; The first one summarizes the statistics of Top-10 and Top-20 malware/countries. The second and third sections are the malware clustering and (source) country clustering, respectively. Clustering results of each section will be compared with other previous work as Reference Data in Figure 3.3.

4.1 Statistics of Top-30 and Top-20 Malware/Country in 2010 & 2012

After Top-30 and Top-20 processing shown in Figure 3.1, Top-30 and Top-20 malware and countries can be achieved in 2010 and 2012, respectively. Only Top-10 and an interesting malware have been listed in Table 4.1 presenting number of downloads, percent of downloads, and number of unique IP addresses. Similarly, those of Top-20 malware are listed in Table 4.2.

In 2010, it can be noticed that PE_VIRUT.AV (Top-1) has been heavily downloaded. While other top malware's download is fewer than one half of PE_VIRUT.AV. Similarly, Trojan.Dropper-18535 is the Top-1 malware in 2012 with more than twice the downloads of other malware.

Table 4.1 Statistics of Top-10 and interesting malware downloads in 2010 CCC dataset. Note that complete Top-30 malware downloads are provided in Appendix A.

Top	Malware Name, w	#Downloads $\sum_d l_w(d)$	%Downloads	Unique IPs
1	PE_VIRUT.AV	194,557	16.7	37,481
2	BKDR_VANBOT.RG	83,757	7.2	6,851
3	WORM_AUTORUN.CZU	46,313	4.0	1
4	WORM_RBOT.SMA	36,171	3.1	26,160
5	TROJ_BUZUS.BEZ	32,172	2.8	2,230
6	WORM_KOLABC.ET	31,967	2.8	2,119
7	BKDR_RBOT.ASA	31,404	2.7	23,744
8	BKDR_NEPOE.CW	30,118	2.6	1,474
9	WORM_KOLAB.EA	28,909	2.5	4
10	WORM_KOLAB.CV	28,586	2.5	2
21	WORM_MAINBOT.MCL	11,298	1.0	1,750

4.2 Malware Clustering 2010 & 2012

This dissertation thoroughly examines and clusters Top-30 malware names that were most frequently downloaded in 2010 CCC dataset. Top-30 malware accounts for 820,260 of total 1,162,093 downloads, representing 70% of the entire dataset. Furthermore, this thesis also thoroughly examines and clusters Top-20 malware names that were most frequently downloaded in 2012 IIJ dataset. Top-20 malware accounts for 17,926,169 of total 32,070,143 downloads, rep-

resenting 55.9% of the entire dataset, roughly 18 million records. The details on these Top-20 malware are summarized in Table 4.2, which contains number of downloads (#DLs), and download percentage (%).

Table 4.2 Top-20 Malware downloads (%) and Top-20 source countries downloads (%) in 2012 IIJ MITF dataset.

Top	Malware Name	#DLs(%)	Country Code	#DLs(%)
1	Trojan.Dropper-18535	3,738,829(23.4)	RU	3,224,066(20.2)
2	Worm.Kido-20	1,766,252(11.0)	TW	2,541,603(15.9)
3	Worm.Kido-102	1,451,967(9.1)	US	1,996,188(12.5)
4	Worm.Kido-99	1,169,602(7.3)	BR	1,468,410(9.2)
5	Trojan.Agent-71049	961,731(6.0)	RO	939,753(5.9)
6	Worm.Agent-194	868,378(5.4)	HU	596,561(3.7)
7	Worm.Kido-367	642,253(4.0)	BG	586,321(3.7)
8	Worm.Kido-182	617,509(3.9)	JP	503,315(3.1)
9	Trojan.Agent-71068	557,115(3.5)	PL	469,245(2.9)
10	Worm.Kido-24	498,357(3.1)	KR	463,310(2.9)
11	Worm.Kido-119	497,163(3.1)	IT	451,485(2.8)
12	Worm.Kido-223	492,787(3.1)	AR	434,514(2.7)
13	Worm.Kido-25	481,447(3.0)	DE	377,545(2.4)
14	Worm.Kido-128	425,643(2.7)	CN	331,364(2.1)
15	Worm.Kido-175	339,416(2.1)	UA	308,955(1.9)
16	Worm.Kido-85	328,762(2.1)	IL	293,313(1.8)
17	Worm.Kido-51	322,627(2.0)	IN	292,117(1.8)
18	Worm.Downadup-113	305,763(1.9)	CA	265,543(1.7)
19	Trojan.Agent-71228	277,149(1.7)	TR	257,814(1.6)
20	Worm.Kido-295	245,161(1.5)	FR	186,489(1.2)

4.2.1 Clustering Feature: Weekly Downloads

To briefly compare the Weekly download behaviors, only Top-10 malware's Weekly Downloads from all countries, $l_m^w(u)$ in Equation (3.3) are plotted in Figure 4.1 and Figure 4.2 for the 2010 and 2012, respectively.

In 2010, the Weekly Download behaviors of each malware are quite different as indicated in Figure 4.1. Some are active throughout the observing period and some are temporarily active. PE_VIRUT.AV(1) may lead to spreading of WORM_RBOT.SMA(4). BKDR_RBOT.ASA(7) has been downloaded before 2010 and BKDR_VANBOT.RG(2) started around week 8 in 2010.

On the other hand, to briefly compare the weekly download behaviors in 2012, only Top-10 malware's Weekly Downloads from Top-20 countries, $l_m^w(u)$ in Equation (3.3) are plotted in Figure 4.2 (note that Top-11 to Top-20 malware's weekly downloads have similar fashion of behaviors, but due to limited space). It can be noticed that in general their weekly download activities are quite active in the beginning. Even though they share similar download trend, as

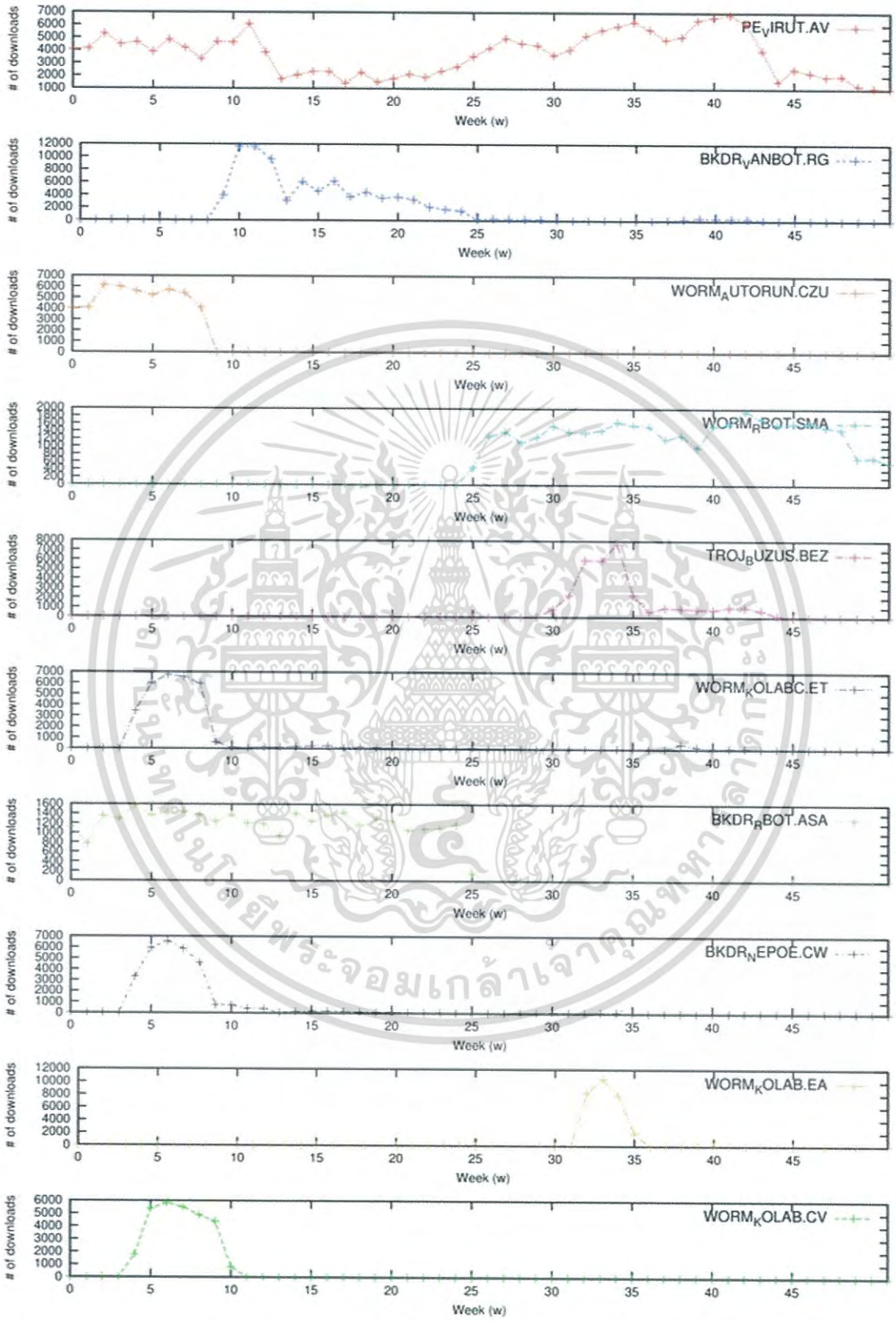

Figure 4.1 Weekly download of Top-10 malware in 2010, $l_m^w(u)$ in Equation (3.3)

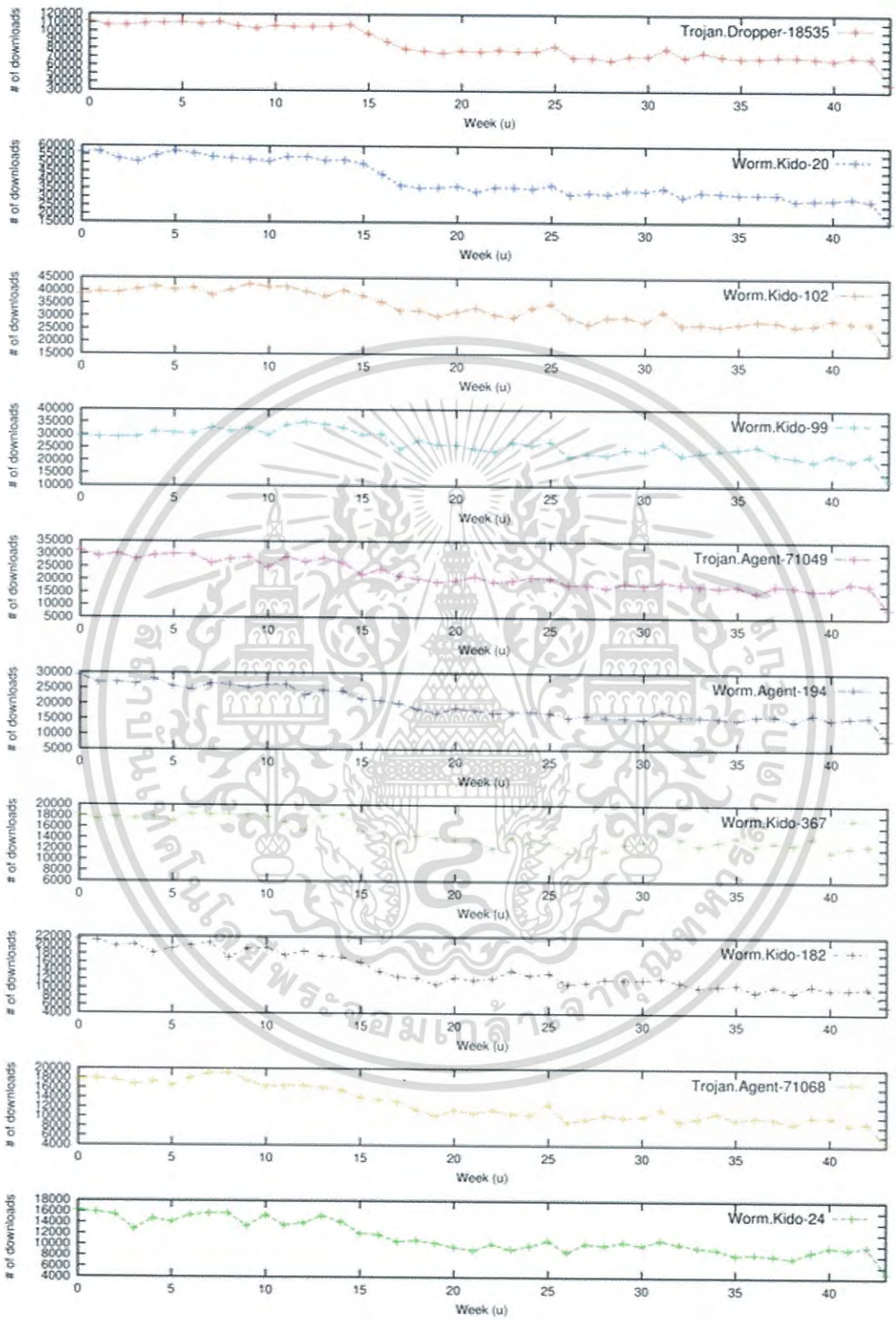

Figure 4.2 Weekly downloads of Top-10 malware from Top-20 source countries, $l_m^w(u)$ in Equation (3.3) where u ranges from 1 to 44 (10 months). Note that Top-11 to Top-20 malware's weekly downloads have similar fashion of behaviors, but due to limited space.

their downloads slightly decrease from the first 15 weeks period till week 44, their quantities are different on the Y axis.

4.2.2 Clustering Feature: Hourly Downloads

The average hourly malware downloads of 2010 and 2012 are summarized in Table 4.3. This table can be associated with the Top-10 malware normalized hourly downloads for visulaiztion as follows.

Table 4.3 Average hourly downloads of Top-10 malware in 2010 and 2012, $l'^{h}_{m}(k) = \sum_{k=0}^{23} l'_m(k)/24$.

Top	2010		2012	
	Malware	$l'^{h}_{m}(k)$	Malware	$l'^{h}_{m}(k)$
1	PE_VIRUT.AV	8,106	Trojan.Dropper-18535	177,383
2	BKDR_VANBOT.RG	3,489	Worm.Kido-20	80,402
3	WORM_AUTORUN.CZU	1,929	Worm.Kido-102	68,990
4	WORM_RBOT.SMA	1,507	Worm.Kido-99	54,323
5	TROJ_BUZUS.BEZ	1,340	Trojan.Agent-71049	42,804
6	WORM_KOLABC.ET	1,331	Worm.Agent-194	42,077
7	BKDR_RBOT.ASA	1,308	Worm.Kido-367	30,240
8	BKDR_NEPOE.CW	1,254	Worm.Kido-182	29,028
9	WORM_KOLAB.EA	1,204	Trojan.Agent-71068	25,434
10	WORM_KOLAB.CV	1,191	Worm.Kido-24	24,416

Normalized hourly downloads of Top-10 malware represent the portion of download activies on any day in particular year in hours 0 to 23. The normalized hourly downloads of 2010 and 2012 are illustrated in Figure 4.3 for 2010 and Figure 4.4 for 2012, respectively.

It can be noticed in Figure 4.3 that the malware are relatively downloaded more frequently at nights (from 19.00-23.00) until midnights. Although the Honeypots are almost always available, malware was relatively less downloaded after midnights until dawns. The plot shows that all of malware have similar download behaviors with the malware dominators activities. Hence, all the malware have coordinated downloading and may relate to each other. On the other hand, these coordinated malware downloads may be infected by one/group of them. Thus, other dependent malware may be downloaded by previous infections as well.

On the other hand, to visually compare the hourly download behaviors of each malware, m in 2012, the Normalized Hourly Downloads of Top-20 Malware from Top-20 source countries are plotted in Figure. 4.4. It can be obviously noticed that relatively high download activities are performed in night time rather than in day time with respect to the Japanese Local Time in spite of the always-ON honeypots.

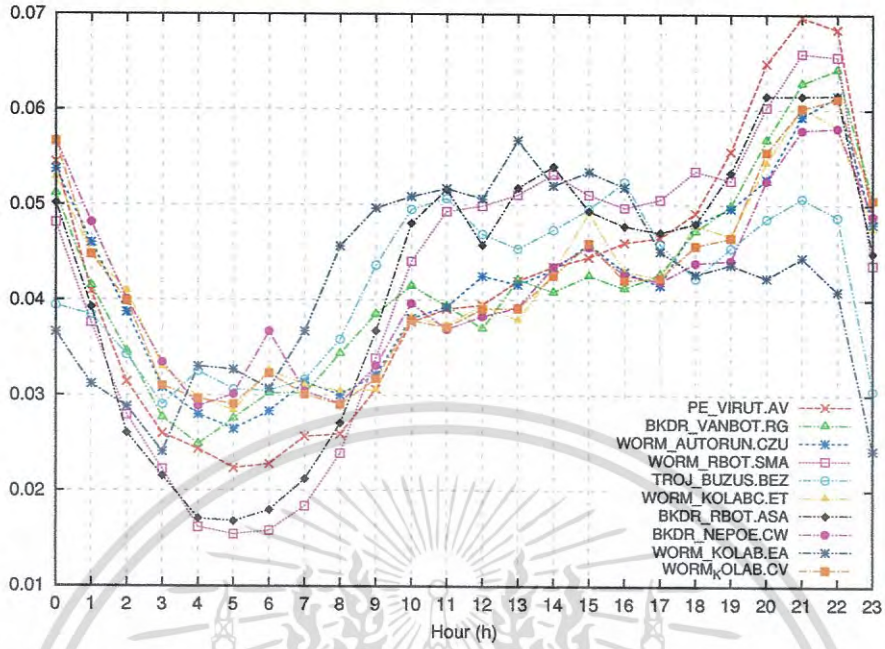

Figure 4.3 Normalized Hourly Downloads of Top-10 Malware from all countries in 2010, $l'^{h}_{m}(k)=l^{h}_{m}(k)/\sum_{i=0}^{23} l^{h}_{m}(i)$, where $k = 0...23$ is the Japanese Local Time and m=[1,2,3,...,10].

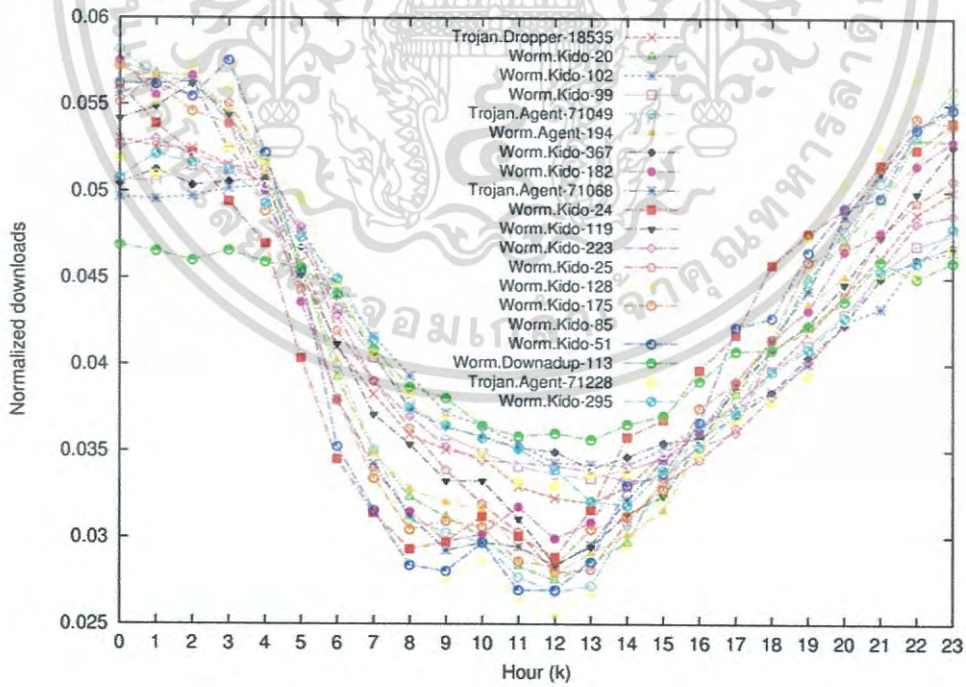

Figure 4.4 Normalized hourly downloads of Top-m malware from Top-20 source countries in 2012, $l'^{h}_{m}(k)=l^{h}_{m}(k)/\sum_{i=0}^{23} l^{h}_{m}(i)$, where $k = 0...23$ is the Japanese Local Time and m=[1,2,3,...,20].

The noticeable observations between 2010 and 2012 Normalized Hourly Downloads of malware are as follows. The off-peak period is in night time for 2010 and in day time for 2012. The peak period of both 2010 and 2012 is in night time. However, the 2012 peak is delayed by 2-3 hours from 2010.

4.2.3 Clustering Feature: Country Downloads

Some malware is unevenly hosted by certain countries. Its variants may show similar behaviors. This spatial clustering feature can represent the geographical distribution of malware around the globe. Based on 2010 CCC dataset, the Normalized Country Downloads of Top-30 Malware can be visualized in Figure 4.5. It shows how the individual Top-30 malware is relatively downloaded from each Top-30 source country. It can be noticed that some malware has been heavily downloaded from certain countries, such as JP and CA according to their ranks in Table A.2. However, some malware with lower ranks are hosted relatively comparable with top ranks malware in certain countries, such as Top-13, 16, 20, etc. as indicated in Figure 4.5.

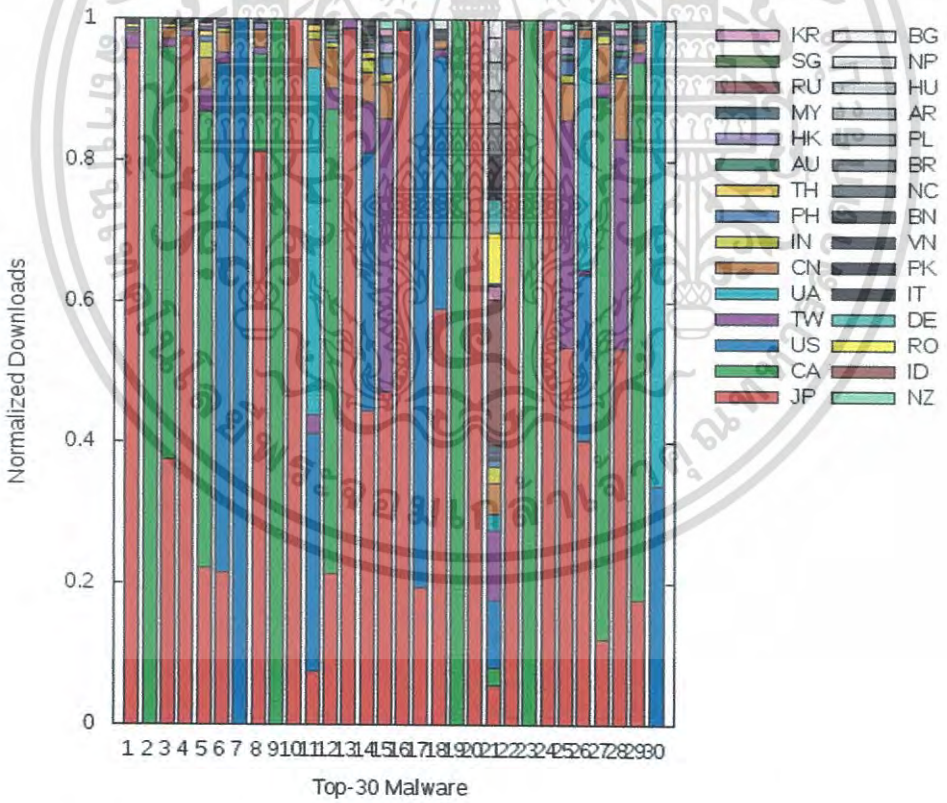

Figure 4.5 Normalized Country Downloads of Top-30 Malware in 2010, $l'^{c}_{m}(p)=l^{c}_{m}(p)/\sum_{n=1}^{30} l^{c}_{n}(p)$, where m=(1,2,3,...,30) and p=[JP,CA,US,...,NZ].

On the other hand, the 2012 Normalized Country Downloads of Top-20 Countries can be visualized in Figure 4.6. It shows how the individual Top-20 malware is relatively downloaded

Table 4.4 Top-10 and interesting source countries of Top-30 malware in 2010 CCC dataset. Note that complete Top-30 source countries downloads are provided in Appendix A.

Top	Country (Code), c	#Downloads, $\sum_d l_c(d)$	%Downloads	Time Zone (UTC/GMT)	Time Diff. wrt Japan, p
1	JAPAN (JP)	399,641	48.86	+9	0
2	CANADA (CA)	235,375	28.78	-5	14
3	USA (US)	67,459	8.25	-5	14
4	TAIWAN (TW)	36,886	4.51	+8	1
5	UKRAINE (UA)	35,655	4.36	+2	7
6	CHINA (CN)	17,634	2.16	+8	1
7	INDIA (IN)	3,572	0.44	+5	4
8	PHILIPPINES (PH)	3,476	0.43	+8	1
9	THAILAND (TH)	2,212	0.27	+7	2
10	AUSTRALIA (AU)	1,783	0.22	+10	-1
16	NEW ZEALAND (NZ)	1,055	0.13	+12	-3
18	ROMANIA (RO)	550	0.07	+2	7
20	ITALY (IT)	436	0.05	+1	8
25	BRAZIL (BR)	347	0.04	-3	12
26	POLAND (PL)	345	0.04	+1	8

from each Top-20 source country. It can be noticed that some malware has been heavily downloaded from certain countries, such as Russia (RU), US, Taiwan (TW) according to their ranks in Table 4.2. However, some malware with lower ranks is hosted comparable to top ranks malware in certain countries, such as Trojan.Agent-71068 (Top-9) in China (CN), Worm.Kido-223 (Top-12) and Worm.Kido-25 (Top-13) in Bulgaria (BG), Worm.-Kido-25 (Top-13) in Argentina (AR) and Brazil (BR), Worm.Kido-128 (Top-14) in Poland (PL), and Worm.-Kido-85 (Top-16) and Worm.Downad-up-113 (Top-18) in Korea (KR).

4.2.4 Malware Clustering Results

Hierarchical clusters (dendrogram) of Top-30 malware in 2010 can be obtained according to Malware Dissimilarity in Equation (3.15) illustrated in Figure 4.7. At $\mathbf{D_{cor}}$=0.45, there are 7 major clusters as listed and explained in Table 4.5. Malware is usually named as TYPE_FAMILY.EXTENSION. TYPE can be BKDR for Backdoor, WORM for worm, TROJ for trojan horse, PE for Portable Executable, etc. Within the same cluster, some of them share the same FAMILY name with different EXTENSIONs.

On the other hand, a dendrogram of Top-20 hierarchical malwarely clustered by $\mathbf{D_{cor}}(Ł_m, Ł_{m'})$ in Equation (3.15) with Complete Linkage, where B and C are Conficker.B and Conficker.C, respectively. Two major clusters can be noticed at $\mathbf{D_{cor}}$=0.4 which is quite large. The left-hand side cluster consists of Top-6, 9, 15, 10, 14, 2, 8, 5, 17, 11 and 13. The right-hand side cluster

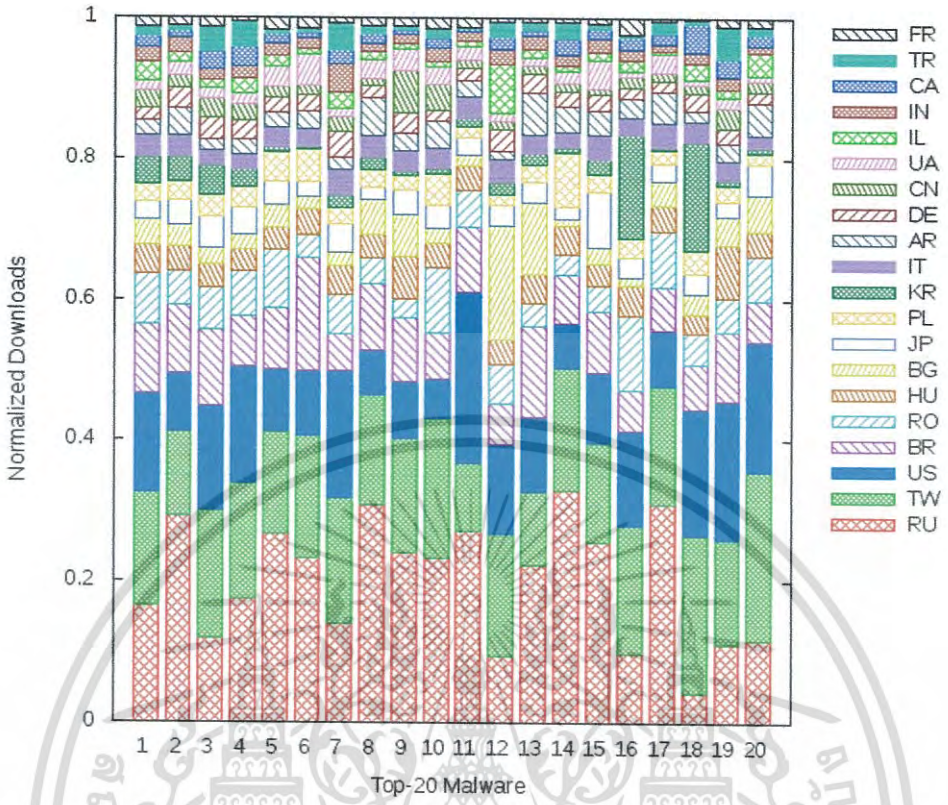

Figure 4.6 Normalized country downloads of each Top-20 malware in 2012, $l'^{c}_{m}(p)=l^{c}_{m}(p)/\sum_{n=1}^{20} l^{c}_{n}(p)$, where m=(1,2,3,...,20) and p=[RU,TW,US,...,FR].

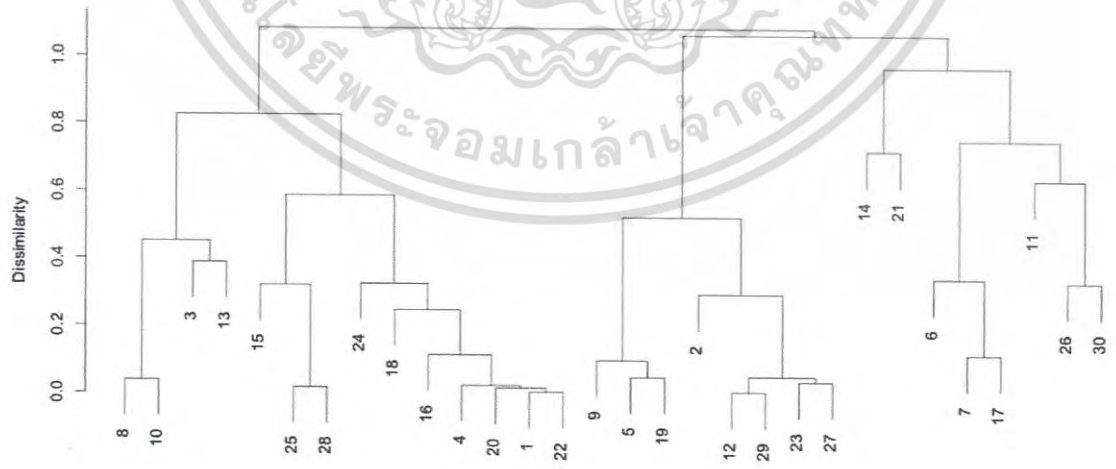

Figure 4.7 Hierarchical clusters (dendrogram) of Top-30 malware according to Malware Dissimilarity in 2010, $D_{w1,w2}$ in Equation (3.15) with both Temporal and Spatioal features and Complete Linkage.

Table 4.5 Similar behaviors of malware clusters in 2010

Cluster	Malware (Top)	Similar behaviors
1	WORM_AUTORUN.CZU(3), BKDR_NEPOE.CW(8), WORM_KOLAB.CV(10), TROJ_IRCBRUTE.BW(13)	They modify system settings to allow remote access to MS Windows PC, giving the attacker full control of the system.
2	WORM_PALEVO.SMD(15), Mal_Swzr3(25), WORM_DOWNAD.AD(28)	Most of them are WORM from infected removable drives, MSN Instant Messenger, software vulnerabilities, and P2P network download. When exploited, it allows a remote user to execute arbitrary code on the infected system in order to propagate across networks.
3	PE_VIRUT.AV(1), WORM_RBOT.SMA(4), BKDR_NEPOE.DM(16), WORM_ALLAPLE.IK(18), PE_VIRUT.PAU(20), BKDR_MYBOT.AH(22), WORM_PALEVO.BE(24)	Use network shares port:135/139/445/593.
4	TROJ_BUZUS.BEZ(5), WORM_KOLAB.EA(9), TROJ_MALWARE.VTG(19)	They shared the same alias name as Trojan.Win32.Buzus(KAV) and copy themselves to system Root (Windows).
5	BKDR_VANBOT.RG(2), BKDR_VANBOT.AHH(12), BKDR_VANBOT.HG(23), WORM_SPYBOT.AWX(27), TSPY_ONLINEG.TKJ(29)	They are backdoors and can spread through network using Net-Worm as a transporter
6	WORM_KOLABC.ET(6), BKDR_RBOT.ASA(7), WORM_PALEVO.AZ(17)	It can be attached with e-mail and infect installed software in the system, and also from using outdated web browser. It typically modifies system settings to automatically start, leads to degrade the system performance as well as allows malicious users to remotely manipulate affected systems.
7	WORM_PALEVO.AK(26), PE_VIRUT.XV(30)	They are usually downloaded from the Internet or malicious web sites and installed by unsuspecting users. They typically carry payloads or other malicious actions that can range from the mildly annoying to the irreparable destructive. They may also modify system settings to automatically start.

consists of Top-16, 18, 12, 7, 20, 19, 3, 1 and 4. Table 4.6 summarizes the results with their aliases according to Figure 4.8. Based on Microsoft definition [17], each malware is either aliased as Conficker.B or Conficker.C.

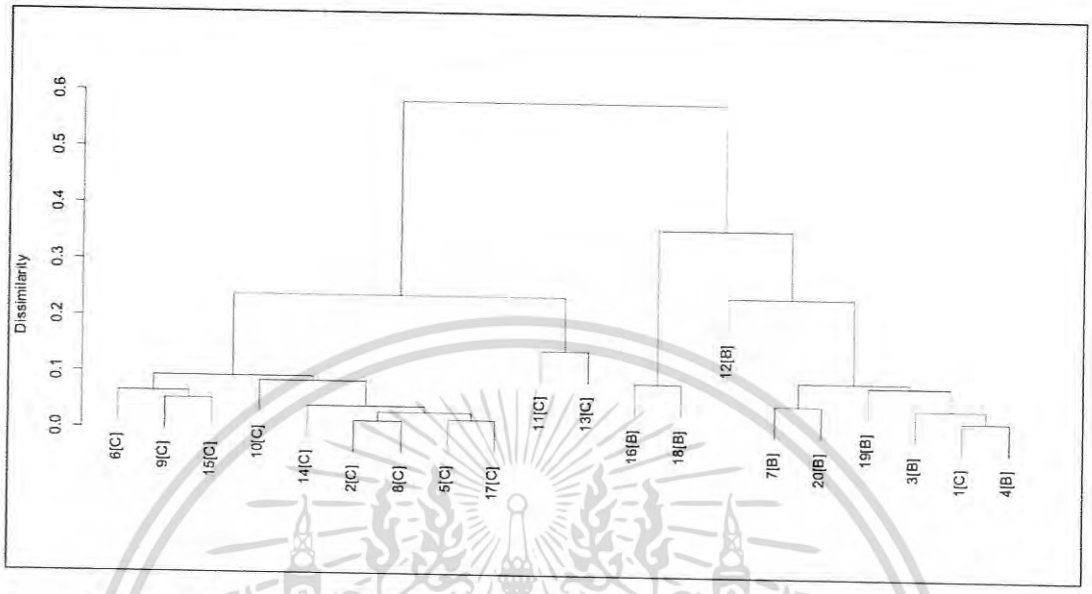

Figure 4.8 A dendrogram of Top-20 hierarchical malwarely clustered by D_{cor} in Eq.(3.15) with both Temporal and Spatial features and Complete Linkage, where B and C are Conficker.B and Conficker.C, respectively.

The intention of this thesis is to study Top-20 malware accounted for 55.9% of the entire dataset as mentioned earlier. Surprisingly, two discovered malware clusters turn out to be two prominent families of Conficker, Conficker.B and Confiker.C. With respect to this fact, the effectiveness of each malware clustering in terms of P (Precision) and R (Recall) from Eq.(3.19) and Eq.(3.20) can be presented in Table 4.8. Among two different dissimilarities (Correlation and Cosine) and three different linkages (Single, Average, and Complete) of hierarchical clustering, Complete Linkage of Correlation/Cosine Dissimilarity, equally provides good P and R at 100.0, 88.9% and 91.7, 100.0% for both Clusters I and II, respectively.

In order to explore the proposed method further, Temporal-only, Spatial-only and Temporal+Spatial feature vectors are tested with both Correlaton and Cosine Dissimilarity metrics and all the linkage options in Appendix C. Due to lack of Reference Data, the resulted malware clusters of Correlation Disimilarity and Complete Linkage are provided in Table 4.7 for 2010 CCC dataset. On the other hand, both P and R can be achieved in Table 4.8 for 2012 IIJ MITF dataset.

Based on the results in Table 4.6, malware clustering is eveluated with a variety of features and different Dissimilarity thresholds. The resulted P and R are tabulated in Table 4.8. The best P and R are highlighted in bold face. It can be observed that Complete Hierarchical clustering

Table 4.6 Two Major Clusters of Top-20 Malware and their Aliases according to Fig. 4.8 with $\mathbf{D}_{\mathbf{cor}}(Ł_m, Ł_{m'})$=0.4, Eq.(3.15) and Complete Linkage.

Top	Malware	Cluster	Alias
2	Worm.Kido-20	I	Conficker.C
5	Trojan.Agent-71049		Conficker.C
6	Worm.Agent-194		Conficker.C
8	Worm.Kido-182		Conficker.C
9	Trojan.Agent-71068		Conficker.C
10	Worm.Kido-24		Conficker.C
11	Worm.Kido-119		Conficker.C
13	Worm.Kido-25		Conficker.C
14	Worm.Kido-128		Conficker.C
15	Worm.Kido-175		Conficker.C
17	Worm.Kido-51		Conficker.C
1	Trojan.Dropper-18535	II	**Conficker.C**
3	Worm.Kido-102		Conficker.B
4	Worm.Kido-99		Conficker.B
7	Worm.Kido-367		Conficker.B
12	Worm.Kido-223		Conficker.B
16	Worm.Kido-85		Conficker.B
18	Worm.Downadup-113		Conficker.B
19	Trojan.Agent-71228		Conficker.B
20	Worm.Kido-295		Conficker.B

option can achieve the best P and R of malware clustering. In general, Correlation Dissimilarity yields better results than Cosine Dissimilarity. For feature selection, Temporal+Spatial-feature provides better P and R than those of other features. The Spatial feature yield almost the same results as those of Temporal+Spatial feature due to its $\mathbf{D}_{\mathbf{cor}}(Ł_m, Ł_{m'})$=0.45.

Table 4.7 Summary of hierarchical malware clustering in 2010 with a variety of feature options, Correlation Dissimilarity and Complete Linkage by Equation (3.15).

$\mathbf{D_{cor}}$	Temporal (Weekly+Hourly)		Spatial (Country)		Temporal+Spatial	
	Clusters	Top-m	Clusters	Top-m	Clusters	Top-m
$0.3 < x \leq 0.4$	I	27,9,5,19	I	11,30	I	8,10
	II	15,25	II	7,6,17	II	3,13
	III	4,22	III	3,9,23,2,19,5,12,27,29	III	15,25,28
	IV	21,24	IV	16,14,18	IV	24,18,16,4,20,1,22
	V	10,6,8,3,23,12,29	V	8,1,13,24,4,22,16,10,20,15,25,28	V	9.5,19
	VI	11,13	-	-	VI	2,12,29,23,27
	VII	1,18	-	-	VII	6,7,17
	VIII	2,30	-	-	VIII	26,30
$0.4 < x \leq 0.5$	I	27,9,5,19	I	11,30	**I**	**8,10,3,13**
	II	15,25	II	7,6,17	**II**	**15,25,28**
	III	20,4,22	III	3,9,23,2,19,5,12,27,29	**III**	**24,18,16,4,20,1,22**
	IV	21,24	IV	26,14,18,8,1,13,24,4,22,16,10,20,15,25,28	**IV**	**9,5,19**
	V	10,6,8,3,23,12,29	-	-	**V**	**2,12,29,23,27**
	VI	11,13	-	-	**VI**	**6,7,17**
	VII	1,18	-	-	**VII**	**26,30**
	VIII	7,2,30	-	-	-	-

Table 4.8 Precision (P) and Recall (R) of supervised hierarchical malware clustering in 2012 with a variety of feature, dissimilarity and linkage options compared by Equation (3.19) and Equation (3.20).

Linkage	$\mathbf{D_{cor}}$ / $\mathbf{D_{cos}}$	Clusters	Temporal (Weeky+Hourly)				Spatial (Country)				Temporal+Spatial			
			$\mathbf{D_{cor}}(Ł_m, Ł_{m'})$		$\mathbf{D_{cos}}(Ł_m, Ł_{m'})$		$\mathbf{D_{cor}}(Ł_m, Ł_{m'})$		$\mathbf{D_{cos}}(Ł_m, Ł_{m'})$		$\mathbf{D_{cor}}(Ł_m, Ł_{m'})$		$\mathbf{D_{cos}}(Ł_m, Ł_{m'})$	
			P (%)	R (%)	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)
Single	$0.3 < x \leq 0.4$	I	60.0	100.0	60.0	100.0	60.0	100.0	60.0	100.0	60.0	100.0	60.0	100.0
		II	40.0	100.0	40.0	100.0	40.0	100.0	40.0	100.0	40.0	100.0	40.0	100.0
	$0.4 < x \leq 0.5$	I	60.0	100.0	60.0	100.0	60.0	100.0	60.0	100.0	60.0	100.0	60.0	100.0
		II	40.0	100.0	40.0	100.0	40.0	100.0	40.0	100.0	40.0	100.0	40.0	100.0
Average	$0.3 < x \leq 0.4$	I	60.0	100.0	60.0	100.0	66.7	100.0	60.0	100.0	66.7	100.0	60.0	100.0
		II	40.0	100.0	40.0	100.0	100.0	25.0	40.0	100.0	100.0	25.0	40.0	100.0
	$0.4 < x \leq 0.5$	I	60.0	100.0	60.0	100.0	60.0	100.0	60.0	100.0	60.0	100.0	60.0	100.0
		II	40.0	100.0	40.0	100.0	40.0	100.0	40.0	100.0	40.0	100.0	40.0	100.0
Complete	$0.3 < x \leq 0.4$	I	60.0	100.0	60.0	100.0	100.0	91.7	**100.0**	**91.7**	**100.0**	**91.7**	**100.0**	**91.7**
		II	40.0	100.0	40.0	100.0	85.7	75.0	**88.9**	**100.0**	**88.9**	**100.0**	**88.9**	**100.0**
	$0.4 < x \leq 0.5$	I	60.0	100.0	60.0	100.0	**100.0**	**91.7**	60.0	100.0	**100.0**	**91.7**	60.0	100.0
		II	40.0	100.0	40.0	100.0	**88.9**	**100.0**	40.0	100.0	**88.9**	**100.0**	40.0	100.0

The proposed malware clustering method differs from [34] in terms of clustering algorithm, features and dissimilarity measures. Reference [34] clustered Top-10 malware download behavior of the 2010 and 2011 CCC datasets based on both weekly and hourly download correlation coefficients only. In addition to that, the proposed method is based on three features of Top-20 malware including weekly, hourly, and country download behaviors as indicated in Fig. 4.6. Therefore, the method should be able to achieve more accurate results due to both spatial and temporal characteristics.

Regarding the hierarchical clustering, several researchers including [22,23,26,32,33,36,37] have employed it with mostly single linkage option as summarized in Table 4.9. However, most methods rely on almost entirely different datasets, with various durations and mostly small number of records. The dataset is quite reliable compared to others' in terms of dataset/year, duration and number of records.

Although two Conficker families have been identified in Top-20 malware, recent security threat reports ([63] and [64]) revealed that Conficker worms were still alive as one of the top 3 malware affecting enterprise and small/medium businesses until 2014. This can also be confirmed in the dataset of [9] that the 2012 IIJ dataset is reliable and not biased.

Table 4.9 Comparison of malware datasets and study methods (Hier: Hierarchical, SL: Single-Linkage, AL: Average-Linkage, CL: Complete-Linkage, Statistics: Statistical Analysis, Pro: Protocol-aware, EM: Expectation Maximization, NA: Not Available).

Ref/Yr	Dataset/Year	Duration	#Records	Method
[24]/2006	Leurre.com/2003	More than12 Mths	1,240,000 Uniq. IPs	Graph-based
[32]/2007	Legacy/2004, Small/2006, Large/2007	1 Yr, 6 Wks, 6 Mths	3,637, 893, 3,698	Hier/SL
[26]/2009	Honeypots/NA	NA	1,195	Hier/SL
[22]/2009	ANUBIS/NA	NA	75,692	Hier/SL
[23]/2010	NA/2009	6 Mths	25,720	Hier/SL
[33]/2012	NA/2011	1Yr	1,108,289	Hier/AL
[65]/2012	Sinkhole/2010	7 Days	24,912,492 Uniq. IPs	Statistics
[34]/2013	CCC/2010&2011	1 Yr & 11 Mths	1,162,093 & 158,734	Correlation
[30]/2013	Malicia&Mixed/2012&2012	11 Mths & 1 Mth	10,600&5,250	Pro & Generic
[53]/2014	WINE IPS telemetry/2009-2011	2 Yrs	10 million Uniq. IPs	Statistics
[36]/2014	Malheur/2009&2013	3 Yrs & NA	3,131& 657	Hier/SL
[37]/2014	Verisign/2012	31 Days	1,565,500 Domains	Hier/SL
[38]/2015	Malicia/2012	11 Mths	11,688	K-means & EM
[9]/2015	Sinkhole/2009-2014	6 Yrs	178×10^6 IPs/Yr	Curve Fitting
[31]/2015	Nugache, Waledac, Zeus /NA	NA	2×10^4, 2×10^4, 2×10^4	EM
Ours	IIJ MITF/2012	10 Mths	32,070,143	Hier/SL, AL, CL

4.3 Country Clustering 2010 & 2012

In addition to malware clustering, clustering Top-30 source countries based on Top-30 malware in 2010 CCC dataset, can account for almost 100% of total downloads. In the 2012 IIJ dataset, clustering Top-20 source countries is introduced which amounts to 89.2% of total downloads, roughly 16 million records. GeoIP Databases [62], a commercial geographical IP (GeoIP) service is used to provide information including country code, city name and so on to identify the source country of a given IP address.

4.3.1 Clustering Feature: Weekly Downloads

Weekly Downloads of source countries in 2010 and 2012 are indicated in Figure 4.9 and Figure 4.10, respectively. Figure 4.9 illustrates the Weekly Downloads of Top-30 malware from Top-10 source countries, $l_c^w(u)$ in Equation (3.10) sorted in ranking order. Note that they are plotted on different scales and w ranges from 1 to 52 (12 months) as described in Table 2.3.

On the other hand, Figure 4.10 illustrates the weekly downloads of Top-20 malware from Top-10 source countries, $l_c^w(u)$ in Equation (3.10) sorted in ranking order. Note different unit scales and u from 1 to 44 (10 months) as described in Table 2.3. Please also note that only weekly downloads of Top-20 malware from Top-10 source countries are shown, the weekly downloads of Top-20 malware from Top-11 to Top-20 source countries have similar fashion of behaviors.

4.3.2 Clustering Feature: Hourly Downloads

The average hourly malware downloads from source countries of 2010 and 2012 are summarized in Table 4.10. This table can be associated with the Top-10 normalized hourly downloads for visulaiztion as follows.

Table 4.10 Average hourly malware downloads from Top-10 source countries in 2010 and 2012, $l_c'^h(k) = \sum_{k=0}^{23} l_c'(k)/24$.

Top	2010		2012	
	Country Name	$l_c'^h(k)$	Country Name	$l_c'^h(k)$
1	JAPAN	16,652	RUSSIA	134,336
2	CANADA	9,807	TAIWAN	105,900
3	UNITED STATES	2,811	UNITED STATES	83,175
4	TAIWAN	1,537	BRAZIL	61,184
5	UKRAINE	1,486	ROMANIA	39,156
6	CHINA	735	BULGARIA	24,857
7	INDIA	149	HUNGARY	24,430
8	PHILIPPINES	145	JAPAN	20,971
9	THAILAND	92	POLAND	19,552
10	AUSTRALIA	74	KOREA	19,305

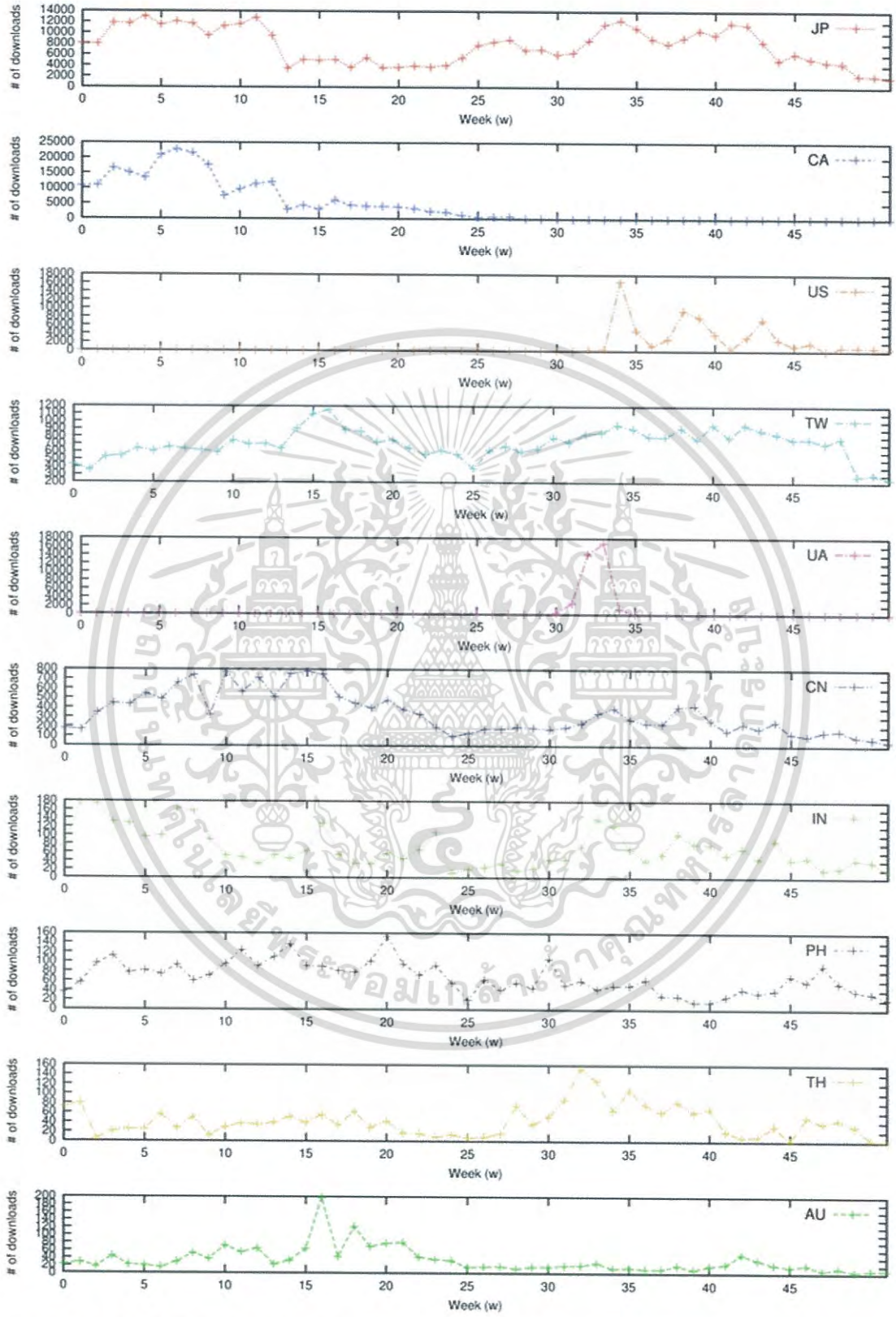

Figure 4.9 Weekly Downloads of Top-30 malware from Top-10 source countries in 2010, $l_c^w(u)$ in Equation (3.10)

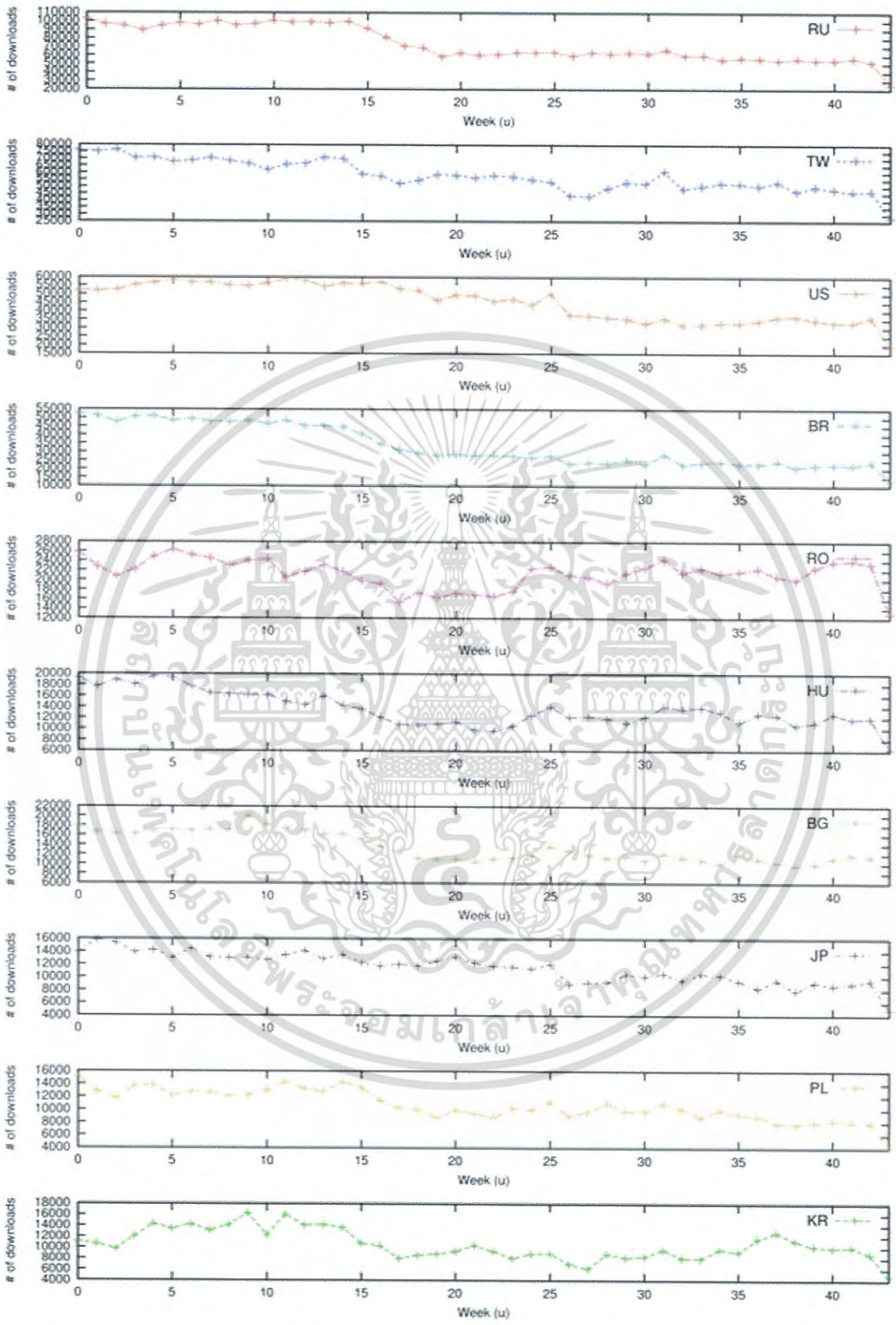

Figure 4.10 Weekly downloads of Top-20 malware from Top-10 source countries, $l_m^w(u)$ in Equation (3.3) where u ranges from 1 to 44 (10 months). Note that Top-11 to Top-20 malware's weekly downloads have similar fashion of behaviors, but due to limited space.

Normalized hourly malware downloads from Top-10 source countries represent the portion of download activies on any day in particular year in hours 0 to 23. The normalized hourly downloads of 2010 and 2012 are illustrated in Figure 4.11 for 2010 and Figure 4.12 for 2012, respectively.

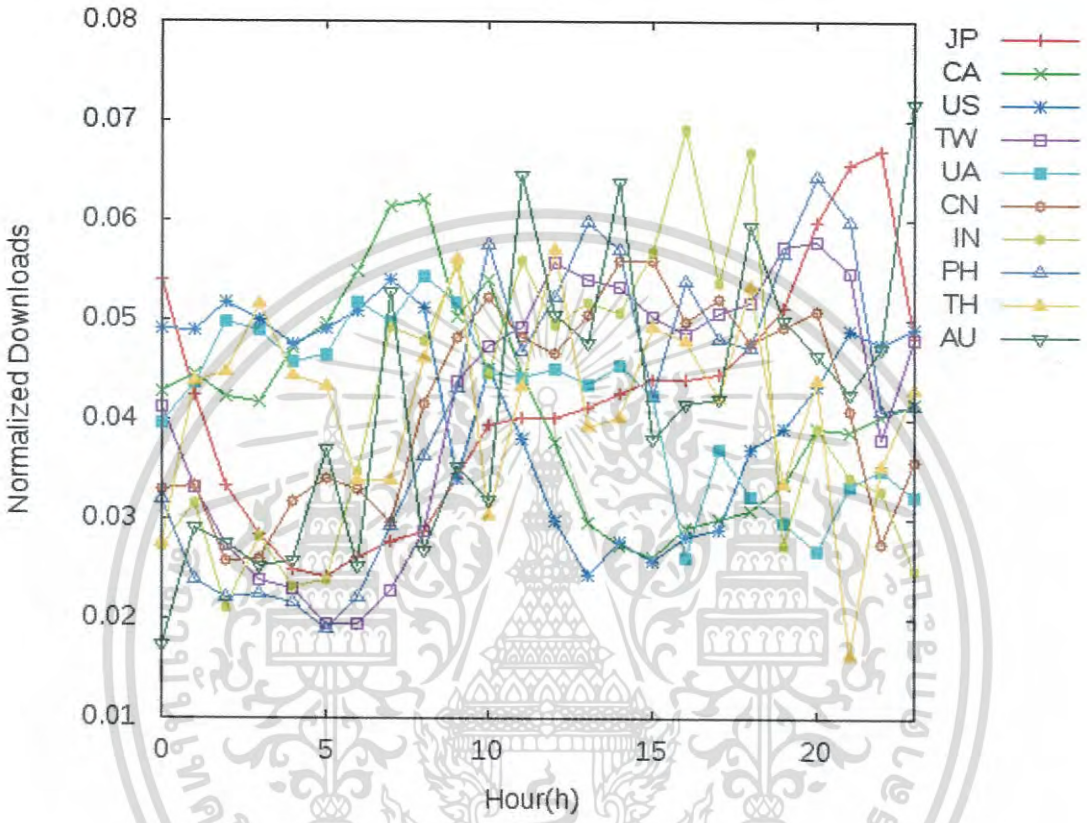

Figure 4.11 Normalized Hourly Downloads of Top-30 Malware from Top-10 Source Countries in 2010, $l_c'^h(k)=l_c^h(k)/\sum_{i=0}^{23} l_p^h(i)$, where $k = 0...23$ is each country's Local Time.

On the ohter hand, to visualize the Normalized Hourly Download among the Top-20 source countries in 2012, Figure 4.12 compares and contrasts the variations of Top-20 malware downloads in hour k ranging 0 to 23 of each country's local time. It can be observed that US and Korea (KR) show almost constant number of downloads all day. On the other hand, several countries show high local time dependency such as Hungary (HU), Bulgaria (BG), etc. It can be described that many zombie PCs are online all day in some countries. On the contrary, some PCs are online especially in the afternoon till midnight.

4.3.3 Clustering Feature: Malware Downloads

Based on 2010 CCC dataset, the Normalized Malware Downloads from Top-30 source countries can be visualized in Figure 4.13. It can be noticed that the Top-1 country hosts a wide

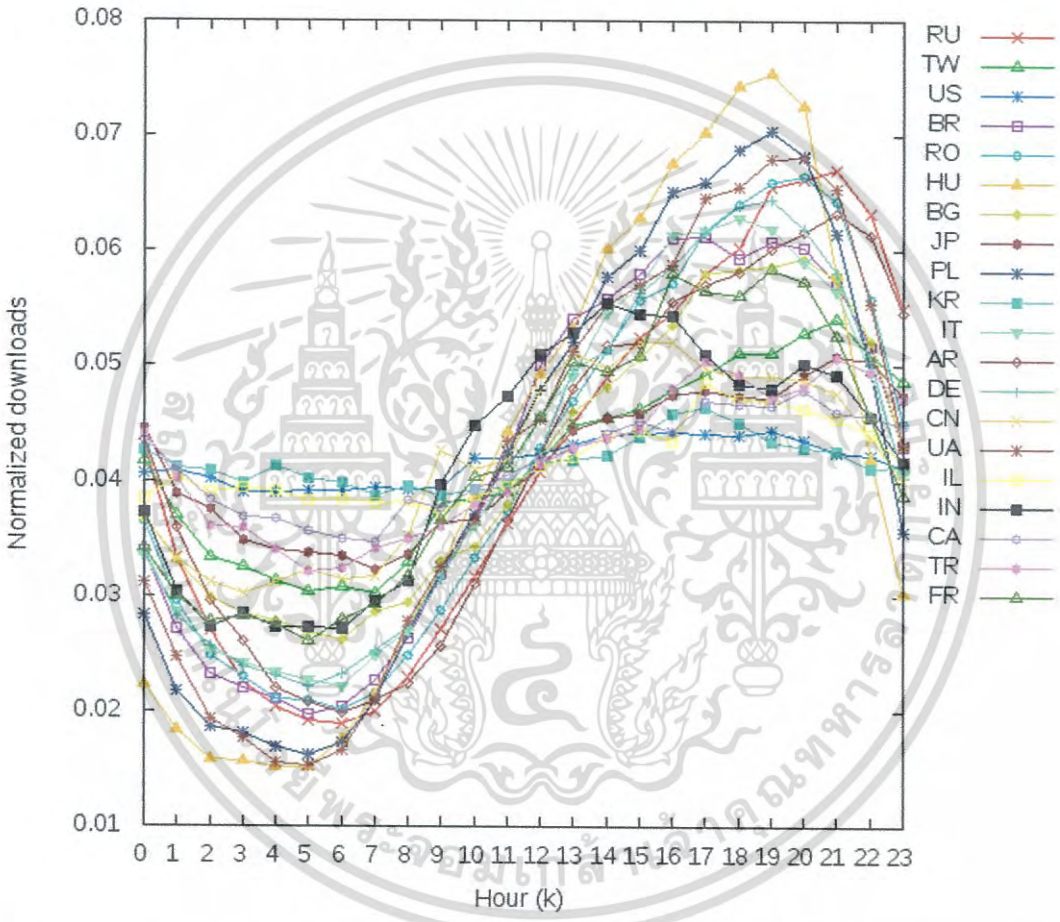

Figure 4.12 Normalized hourly downloads of Top-20 malware from Top-20 source countries in 2012, $l_c'^h(k)$=$l_c^h(k)/\sum_{i=0}^{23} l_p^h(i)$, where $k = 0...23$ is each country's Local Time.

variety of malware while some lower ranking countries do not. PE_VIRUT.AV (Top-1) is the most popular malware among these countries. PE_VIRUT.AV (Top-1) is a major portion of malware distribution (almost 50%) in Japan (Top-1) and Thailand (Top-9). BKDR_VANBOT.RG (Top-2) can only be spotted in Canada (Top-2). Top-18 malware is about 50% of all malware in Top-29 country. Top-13, Top-18 to Top-20, Top-25 to Top-28 and Top-30 countries are major hosts of Top-21 malware.

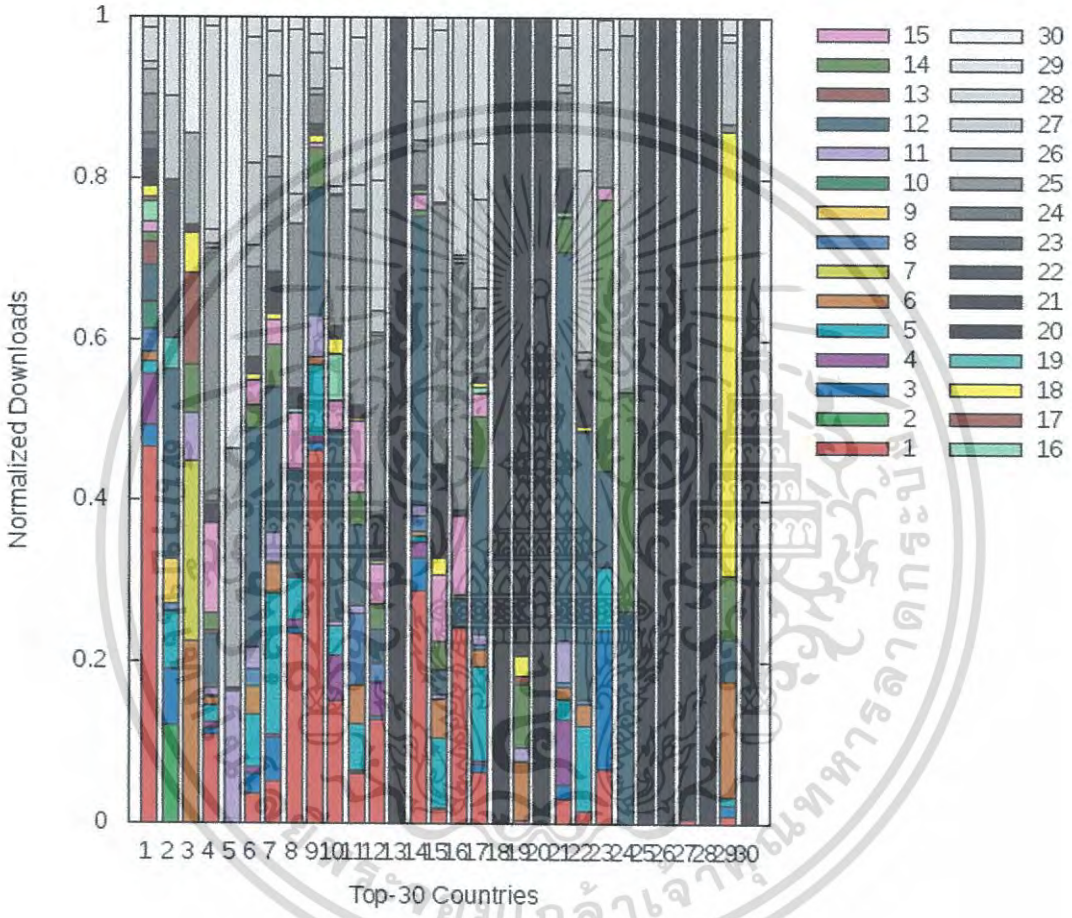

Figure 4.13 Normalized Malware Download from Top-30 Source Countries in 2010, $l_c'^m(n)=l_c^m(n)/\sum_{p=1}^{30} l_p^m(n)$, where c=[JP,CA,US,...,NZ] and n=[1,2,3,...,20].

On the other hand, the Normalized Malware Download behavior of Top-20 Malware in 2012 can be visualized in Figure 4.14. The normalized malware download distribution patterns are quite similar in some countries. However, some observations can be noted as follows. Top-1 malware is hosted in every Top-20 country, relatively high in Israel (IL) and Korea (KR) with very high portions, 34% and 30%, respectively. Likewise, Top-3, Top-4 and Top-7 malware can be visually spotted in Turkey (TR). Significant portions of Top-16 and Top-18 malware can be spotted in Korea. Finally, Bulgaria (BG) and Israel (IL) share significant portion of Top-12

malware, which is similar to the Top-13 in BG and AR.

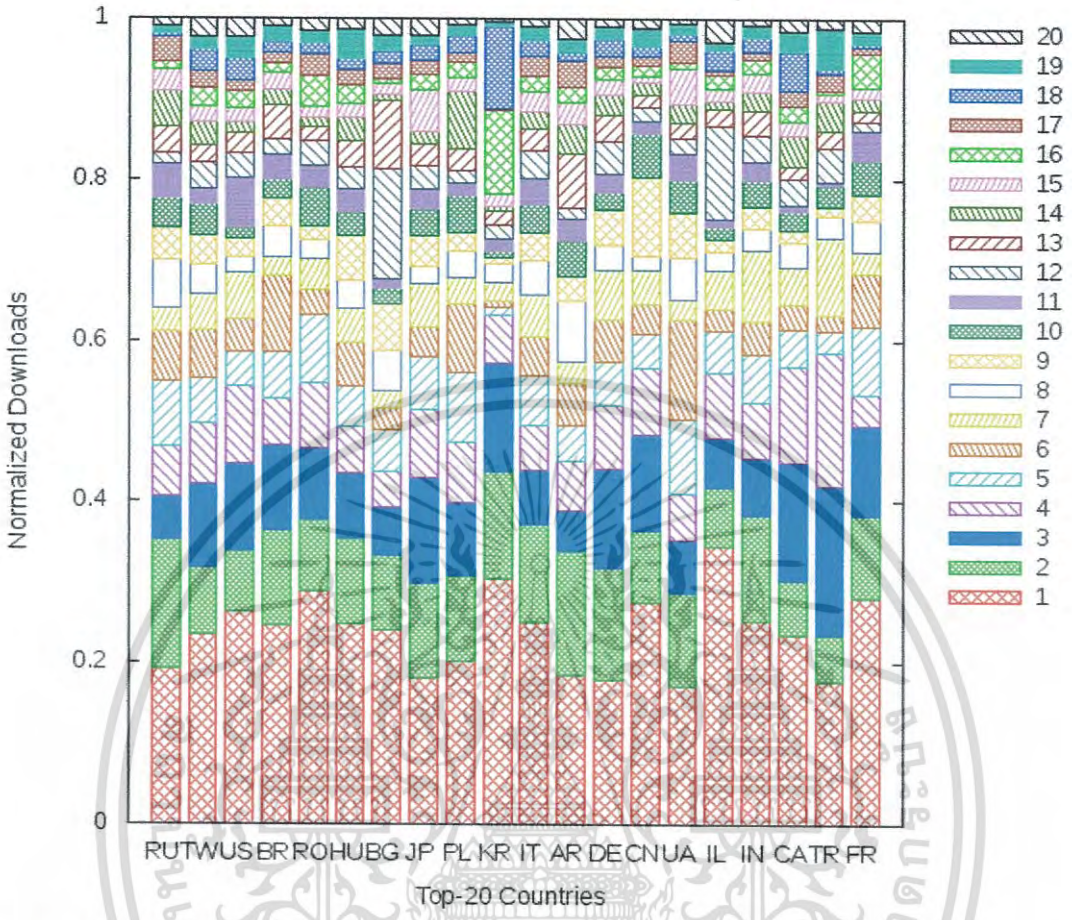

Figure 4.14 Normalized malware downloads of each Top-n malware from Top-20 source countries in 2012, $l_c'^m(n)=l_c^m(n)/\sum_{p=1}^{20} l_p^m(n)$, where c=[RU,TW,US,...,FR] and n=[1,2,3,...,20].

4.3.4 Country Clustering Results

The resulted dendrogram of Top-30 countries in 2010 based on $\mathbf{D_{cor}}$ in Eq.(3.16) with Complete Linkage can be depicted in Figure 4.15. At $\mathbf{D_{cor}}$=0.39, it can be noticed that Top-30 source countries can be clustered into the European (i.e., DE, IT, PL, RO, HU, BG) and non-European (i.e., HK, KR, MY, PH, TW) countries groups as indicated in Figure 4.15.

On the other hand, the resulted dendrogram of Top-20 countries in 2012 based on $\mathbf{D_{cor}}$ in Eq.(3.16) with Complete Linkage can be depicted in Figure 4.16. At $\mathbf{D_{cor}}$ =0.39 which is relatively high, two clusters can be obtained as listed in Table 4.11 in the first row. Note that TR (Turkey) is excluded due to high $\mathbf{D_{cor}}$.

According to the statistical analysis and curve fitting in [9], the growth rates (ϕ_g) of Conficker in 62 countries can differentiate them apart where its median=0.08. These countries are

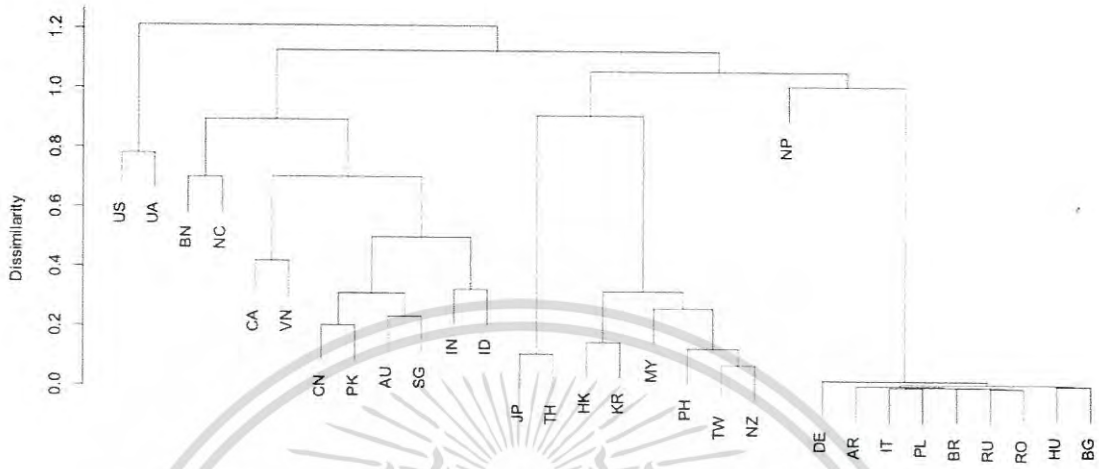

Figure 4.15 Hierarchical clusters (dendrogram) of Top-30 Countries according to Country Dissimilarity in 2010, $D_{c1,c2}$ in Equation (3.16) with both Temporal and Malware features and Complete Linkage.

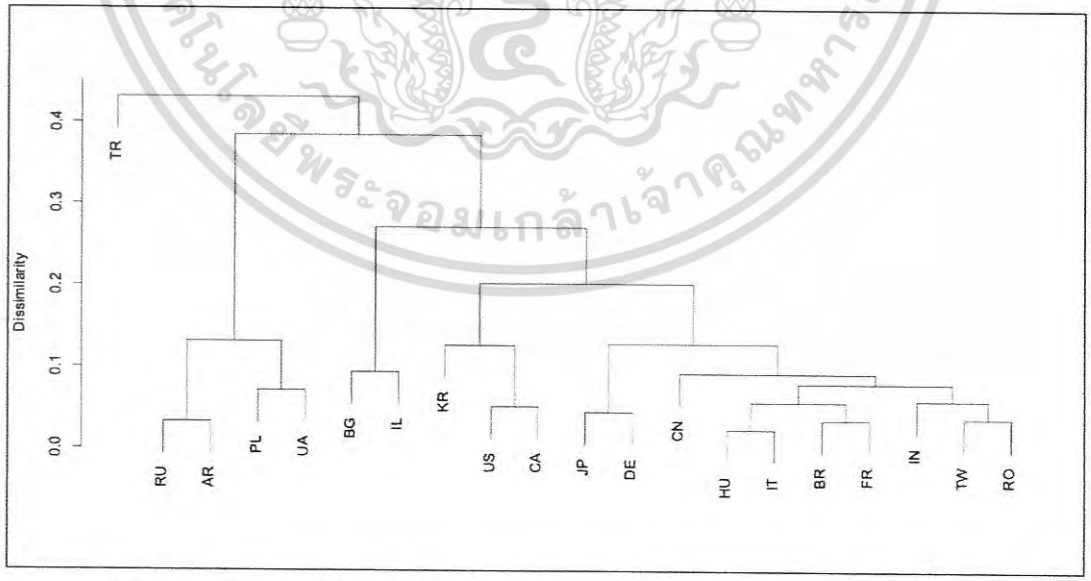

Figure 4.16 A Dendrogram of Top-20 countries hierarchically clustered by $\mathbf{D_{cor}}(Ł_c, Ł_{c'})$ in Equation (3.16) with both Temporal and Malware features and Complete Linkage.

ranked according to their growth rates and can be visually separated at $\phi_g \geq 0.16$ and $\phi_g < 0.16$. Note that only countries corresponding to Top-20 countries are listed in Table 4.11 as the second row. The only difference between these two rows of Table 4.11 is that BR and PL are misplaced between Clusters I and II with respect to [9]. The high growth rate of Conficker in a particular country corresponds to both low ICT development and high software piracy [9]. In addition, Shin et al. [65] reported during 7 days of 2010 that mostly (99%) Conficker can be spread via ADSL/dial-up (less than 1 Mbps) rather than high-speed connections. Their top 10 source countries were CN, BR, RU, IN, IT, VN, TW, DE, AR, ID where only VN (Vietnam) and ID (Indonesia) are not included in Top-20.

Table 4.11 Comparison of country clusters obtained from ours in Fig.4.16 and [9] with Growth Rate (ϕ_g). Note that India (IN) was not included in [9] dataset and Turkey (TR) is excluded.

Method	Cluster	
	I ($\phi_g \geq 0.16$)	II ($\phi_g < 0.16$)
Ours $D_{cor}(Ł_c, Ł_{c'})$=0.39	AR, *PL*, RU, UA	*BR*, BG, CA, CN, DE, FR, HU, IL, IN, IT, JP, KR, RO, TW, US
[9] Clustered by Growth Rate (ϕ_g)	AR, BR, RU, UA	BG, CA, CN, DE, FR, HU, IL, IT, JP, KR, PL, RO, TR, TW, US

Similarly, P and R with respect to [9] in the second row of Table 4.11 are summarized in Table 4.13 to show the effectiveness of our proposed country clustering. Among two different dissimilarities and three different linkages of hierarchical clustering, the Correlation Dissimilarity & Complete Linkage option provides the best P and R at 75.0%, 86.7% for Clusters I and II, respectively. That is because PL and BR are exchanged as shown in the first row of Table 4.11.

According to [9], (AR, RU) and (BR, UA) are similar in terms of Growth Rate and Peak Height. In Cluster I: RU & UA, AR & BR can be regarded in the same cluster as they are located in Eastern Europe and South America, respectively. RU and UA are from Eastern Europe AR and BR represent South America On the other hand, in Cluster II, the obvious subcluster in Fig. 4.16 is US & CA that both are neighboring countries. Several pairs of European countries are neighbors as well, for example, FR & IT, BG & RO, and HU & IT (same time zone).

Regarding to the previous country clustering [52], the proposed country clustering method has been improved in terms of method (i.e., the hierarchical clustering) and features i.e., weekly, hourly, and malware downloads as depicted in Figures 4.10, Figure 4.12 and Figure 4.14, respec-

tively. This improvement may yield more accurate results as summarized in Table 4.13.

In order to explore the proposed method further, Temporal-only, Malware-only and Temporal+Malware feature vectors are tested with both Correlaton and Cosine Dissimilarity metrics and all the linkage options in Appendix C. Due to lack of Reference Data, the resulted country clusters of Correlation Disimilarity and Complete Linkage are provided in Table 4.12 for 2010 CCC dataset. On the other hand, both P and R can be achieved in Table 4.13 for 2012 IIJ MITF dataset.

Based on the results in Table 4.11, country clustering is eveluated with a variety of features and different Dissimilarity thresholds. The resulted P and R are tabulated in Table 4.13. The best P and R are highlighted in bold face. It can be observed that Complete Hierarchical clustering option can achieve the best P and R of country clustering. In general, Correlation Dissimilarity yields better results than Cosine Dissimilarity. For feature selection, Malware-only feature provides even better P and R than those of Temporal+Malware-feature. The Temporal (Weekly+Hourly) feature yield the worst results due to its D_{cor} =0.39.

For practical use of country clustering, as Confickers prefer to infect nearby hosts in their /24 subnets rather than random hosts globally [65,66]. Country clustering is a coarse-grained view to complement the reputation-based detection systems to black list individual hosts. The ISPs shall be more proactive to detect, notify, and educate their own users since the broadband Internet connections are becoming popular. For example, at the ISPs, incoming traffic from suspicious countries shall be logged for realtime/near-realtime analysis. Coordination of major ISPs in the Netherlands named AbuseHUB has accelerated botnet mitigation and was ranked top in the world [66]. In addition, M3AAWG [67] describes the best practices for ISPs and Mailbox Providers by Messaging, Malware, Mobile Anti-Abuse Working Group members to combat phishing attacks.

Although Confickers were in the slow Internet era, FTTX, Cable, or ADSL connections shall be warned or disrupted by the ISPs due to malicious contents or anomaly behaviours detected earlier in the last mile. Even more, broadband routers with some Anti Malware capabilities including Bitdefender's BOX shall be bundled with the subscriptions to stop the malware from/to the consumers. Unfortunately, the routers themselves can be infected or compromised especially SoHo (Small office/Home office) routers [68].

Table 4.12 Summary of hierarchical country clustering in 2010 with a variety of feature options, a Correlation Dissimilarity and Complete Linkage by Equation (3.16).

$\mathbf{D_{cor}}$	Temporal (Weekly+Hourly)		Malware		Temporal+Malware	
	Clusters	Top-c	Clusters	Top-c	Clusters	Top-c
$0.3 < x \leq 0.4$	I	BR,AR,IT,PL,HU,RU,RO,DE,BG	I	DE,AR,PL,RO,RU,BG,HU,IT,BR	I	**CN,PK,AU,SG**
	II	CN,BN	II	JP,TH	II	**IN,ID**
	III	KR,JP,IN	III	HK,KR,MY,PH,TW,ZN	III	**JP,TH**
	IV	HK,CN,TW,PH	IV	CN,PK,AU,SG	IV	**HK,KR,MY,PH,TW,NZ**
	-	-	V	IN,ID,VN	V	**DE,AR,IT,PL,BR,RU,RO,HU,BG**
$0.4 < x \leq 0.5$	I	BR,AR,IT,PL,HU,RU,RO,DE,BG	I	DE,AR,PL,RO,RU,BG,HU,IT,BR	I	CA,VN
	II	AU,SG	II	JP,TH	II	CN,PK,AU,SG,IN,ID
	III	CA,BN	III	HK,KR,MY,PH,TW,ZN	III	JP,TH
	IV	UA,TH	IV	CN,PK,AU,SG,IN,ID,VN	IV	HK,KR,MY,PH,TW,NZ
	V	KR,JP,IN	-	-	V	DE,AR,IT,PL,BR,RU,RO,HU,BG
	VI	HK,CN,TW,PH,MY,NZ	-	-	-	-

Table 4.13 Precision (P) and Recall (R) of supervised hierarchical country clustering in 2012 with a variety of feature, dissimilarity and linkage options compared by Equation (3.19) and Equation (3.20).

Linkage	$\mathbf{D_{cor}}$ / $\mathbf{D_{cos}}$	Clusters	Temporal (Weeky+Hourly)				Malware				Temporal+Malware			
			$\mathbf{D_{cor}}(\mathbf{Ł_m}, \mathbf{Ł_{m'}})$		$\mathbf{D_{cos}}(\mathbf{Ł_m}, \mathbf{Ł_{m'}})$		$\mathbf{D_{cor}}(\mathbf{Ł_m}, \mathbf{Ł_{m'}})$		$\mathbf{D_{cos}}(\mathbf{Ł_m}, \mathbf{Ł_{m'}})$		$\mathbf{D_{cor}}(\mathbf{Ł_m}, \mathbf{Ł_{m'}})$		$\mathbf{D_{cos}}(\mathbf{Ł_m}, \mathbf{Ł_{m'}})$	
			P (%)	R (%)	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)
Single	$0.3 < x \leq 0.4$	I	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0
		II	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0
	$0.4 < x \leq 0.5$	I	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0
		II	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0
Average	$0.3 < x \leq 0.4$	I	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0
		II	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0
	$0.4 < x \leq 0.5$	I	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0	20.0	100.0
		II	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0	75.0	100.0
Complete	$0.3 < x \leq 0.4$	I	25.0	100.0	25.0	100.0	**80.0**	**100.0**	20.0	100.0	**75.0**	**75.0**	20.0	100.0
		II	100.0	26.7	100.0	26.7	**92.9**	**86.7**	75.0	100.0	**86.7**	**86.7**	75.0	100.0
	$0.4 < x \leq 0.5$	I	20.0	100.0	20.0	100.0	21.1	100.0	20.0	100.0	20.0	100.0	20.0	100.0
		II	75.0	100.0	75.0	100.0	73.7	93.3	75.0	100.0	75.0	100.0	75.0	100.0

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Clustering by analyzing a huge number of variants of the same bot/malware has been a challenging problem until now. This dissertation intends to propose a novel *Spatio-Temporal* Clustering of post-labeled bot/malware whose feature vectors are derived from their weekly, hourly and country download behaviors. The algorithm was evaluated by clustering Top-30 malware and source countries into groups based on *Malware and Country Dissimilarity* derived from a number of correlation coefficients in 2010 CCC (`https://www.telecom-isac.jp/ccc/en_index.html`) dataset as well as Top-20 bot/malware and source countries of 2012 IIJ dataset. Note that IIJ (Internet Initiative Japan, `http://www.iij.ad.jp/en/index.html`) is a Tier-1 ISP company in Japan. Therefore, seven clusters of 2010 Top-30 bot/malware were achieved based on their network behaviors as summarized in Table 4.5. However, different options of feature, dissimilarity and linkage yield different results as summarized in Table C.1, Table C.2, and Table C.3. On the other hand, in 2012, those clustered Top-20 botnets eventually correspond to the widely known *Conficker.B* and *Conficker.C* as provided in Table 4.6 without any bias. Similarly, different options of feature, dissimilarity and linkage yield different Precision and Recall percentages as summarized in Table 4.8.

The by-product *Country Clustering* can be applied to distinguish Top-30 and Top-20 source countries according to the Top-30 and Top-20 botnets in 2010 and 2012, respectively. The result of Top-30 source countries can be clustered as the European and non-European group countries. However, different options of feature, dissimilarity and linkage yield different results as summarized in Table C.4, Table C.5, and Table C.6. On the other hand, two clusters can be identified for Top-20 countries in 2012. Similarly, different options of feature, dissimilarity and linkage yield different Precision and Recall percentages as summarized in Table 4.13. According to the recent research efforts on Conficker families especially [9] show relevant results similar to the Top-20 source countries. The resulting country clusters are almost identical to [9]. This can prove that both clustering algorithms are effective. In addition, the 2012 IIJ dataset is quite reliable compared to others'.

5.2 Future Work

As our future work, we need to exploit other clustering features such as URLs, source port and destination port numbers, multiple time zones of bigger countries, e.g. RU, US, CA, etc. The proposed method can be extended to Top-50 or even more to be a Pre-Labeled approach based on Hash values. We believe that our method can be applied to more malware/countries to support detection/mitigati-on efforts against current and future P2P botnets.

REFERENCES

[1] X. Hu, "Large-scale malware analysis, detection, and signature generation," Ph.D. dissertation, The University of Michigan, 2011.

[2] *Symantec, "Symantec Global Internet Security Threat Report", Trends for 2009*, p. 97 pages, April 2010.

[3] C. A. Schiller, J. Binkley, D. Harley, G. Evron, T. Bradley, C. Willems, and M. Cross, *Botnets - THE KILLER WEB APP.* SYNGRESS, 2007.

[4] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *2009 Third International Conference on Emerging Security Information, Systems and Technologies*, 2009, pp. 268–273.

[5] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol - and structure-independent botnet detection," in *Proceedings of the 17th conference on Security Symposium*, July 2008, pp. 139–154.

[6] C. J. Dietrich, *http://blog.cj2s.de/archives/32-Tracking-the-Command-and-Control-Activity-of-Botnets.html*, accessed May. 2014.

[7] C. Rossow, "Using malware analysis to evaluate botnet resilience," Ph.D. dissertation, Vrije Universiteit Amsterdam, 2013.

[8] P. E. Berg, "Behavior-based classification of botnet malware," Master's thesis, Gjøvik University College, July 2011.

[9] H. Asghari, M. Ciere, and M. J. van Eeten, "Post-mortem of a zombie: Conficker cleanup after six years," in *24th USENIX Security Symposium (USENIX Security 15)*, August 2015, pp. 1–16.

[10] *Computer Security Institute, 15th annual 2010/2011 Computer Crime and Security Survey, Technical report, Computer Security Institute*, 2010.

[11] *18 March 2013, DDoS strike on Spamhaus highlights need to close DNS open resolvers, http://www.techrepublic.com/blog/it-security/ddos-strike-on-spamhaus-highlights-need-to-close-dns-open-resolvers/#*, accessed Jul. 2014.

[12] *Hope is Not a Strategy - 2012 Annual DDoS Attack and Impact Survey: A Year-to-Year Analysis*, Neustar, Inc., 2012.

[13] *26 August 2013, China Hit with Biggest DDoS Attack in its History, http://www.infosecurity-magazine.com/view/34160/china-hit-with-biggest-ddos-attack-in-its-history/*, accessed, Jul. 2014.

[14] G. Ollmann, "Serial variant evasion tactics: Techniques used to automatically bypass antivirus technologies," Technical Report, Damballa, Tech. Rep., 2009.

[15] *Symantec, Internet Security Threat Report, Volume 20,*, p. 119 pages, April 2015.

[16] *Anubis - Malware Analysis for Unknown Binaries, https://anubis.iseclab.org*, accessed Jul. 2014.

[17] *VirusTotal, http://www.virustotal.com/*, accessed Mar. 2014.

[18] *Damballa, Inc., 3Bot-driven Targeted Attack Malware, http://www.prnewswire.com/news-releases/3-to-5-of-enterprise-assets-are-compromised-by-bot-driven-targeted-attack-malware-61634867.html*, accessed Jul. 2014.

[19] N. Singh and S. S. Khurmi, "Malware analysis, clustering and classification: A literature review," *InternatIonal Journal of Computer Science and Technology*, vol. 6, issue 1, pp. 68–72, March 2015.

[20] E. Gandotra, D. Bansal, and S. Sofat, "Malware analysis and classification: A survey," *Journal of Information Security*, vol. 5, no. 2, pp. 56–64. doi: 10.4236/jis.2014.52 006, April 2014.

[21] *McAfee, "A Look at One Day of Malware Samples", http://blogs.mcafee.com/mcafee-labs/a-look-at-one-day-of-malware-samples*, accessed Feb. 2014.

[22] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, "Scalable, behavior-based malware clustering," in *Proceedings of the 16th Annual Network and Distributed System Security Symposium (NDSS 2009)*, January 2009, p. 18 pages.

[23] R. Perdisci, W. Lee, and N. Feamster, "Behavioral clustering of http-based malware and signature generation using malicious network traces," in *NSDI'10 Proceedings of the 7th USENIX conference on Networked systems design and implementation*, Sept. 2010, p. 14 pages.

[24] F. Pouget, M. Dacier, J. Zimmerman, A. Clark, and G. Mohay, "Internet attack knowledge discovery via clusters and cliques of attack traces," *Journal of Information Assurance and Security*, vol. 1, pp. 21–32, 2006.

[25] G. Wicherski, "pehash: a novel approach to fast malware clustering," in *LEET'09 Proceedings of the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*, April 2009, p. 8 pages.

[26] M. Apel, C. Bockermann, and M. Meier, "Measuring similarity of malware behavior," in *The 5th LCN Workshop on Security in Communications Networks (SICK 2009)*, October 2009, pp. 891–898.

[27] W. Lu, G. Rammidi, and A. A. Ghorbani, "Clustering botnet communication traffic based on n-gram feature selection," *Computer Communications*, vol. 34, no. 3, pp. 502–514, March 2011.

[28] H. Choi and H. Lee, "Identifying botnets by capturing group activities in dns traffic," *Computer Networks*, vol. 56, no. 1, pp. 20–33, January 2012.

[29] M. Chandramohan, H. B. K. Tan, and L. K. Shar, "Scalable malware clustering through coarse-grained behavior modeling," in *FSE'12 Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, November 2012, p. 4 pages.

[30] M. Z. Rafique and J. Caballero, "Firma: Malware clustering and network signature generation with mixed network behaviors," in *Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2013)*, October 2013, pp. 144–163.

[31] P. Barthakur, M. Dahal, and M. K. Ghose, "Clusibothealer: Botnet detection through similarity analysis of clusters," *Journal of Advances in Computer Networks*, vol. 3, no. 1, pp. 49–55. doi: 10.7763/JACN.2015.V3.141, March 2015.

[32] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of internet malware," in *Proceedings of RAID 2007*, Sept. 2007, pp. 178–197.

[33] R. Perdisci and M. U, "Vamo: Towards a fully automated malware clustering validity analysis," in *Annual Computer Security Applications Conference*, Dec. 2012, pp. 329–338.

[34] C. Yukonhiatou, S. Kittitornkun, H. Kikuchi, K. Sisaat, M. Terada, and H. Ishii, "Clustering top 10 malware/bots based on temporal behavior," in *International Conference on Information Technology and Electrical Engineering*, October 2013, pp. 62–67.

[35] X. Hu, S. Bhatkar, K. Griffin, and K. G. Shin, "Mutantx-s: Scalable malware clustering based on static features," in *2013 USENIX Annual Technical Conference (USENIX ATC'13)*, June. 2013, pp. 187–198.

[36] B. Biggio, K. Rieck, D. Ariu, C. Wressnegger, I. Corona, G. Giacinto, and F. Roli, "Poisoning behavioral malware clustering," in *AISec'14 Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, November 2014, pp. 27–36. doi: 10.1145/2 666 652.2 666 666.

[37] M. Thomas and A. Mohaisen, "Kindred domains: Detecting and clustering botnet domains using dns traffic," in *WWW'14 Companion Proceedings of the 23rd International Conference on World Wide Web*, April 2014, pp. 707–712. doi: 10.1145/2 567 948.2 579 359.

[38] U. Narra, F. D. Troia, V. A. Corrado, T. H. Austin, and M. Stamp, "Clustering versus svm for malware detection," *Journal of Computer Virology and Hacking Techniques*, pp. 12 pages. doi: 10.1007/s11 416–015–0253–z, October 2015.

[39] *Cisco Systems, Inc., "What Is the Difference: Viruses, Worms, Trojans, and Bots?", http://www.cisco.com/web/about/security/intelligence/virus-worm-diffs.html*, accessed Jul. 2014.

[40] P. Barford and V. Yegneswaran, "An inside look at botnets," in *Special Workshop on Malware Detection, Advances in Information Security, Springer Verlag*, 2006.

[41] *T. Micro, Worm AgoBot, 2004, http://about-threats.trendmicro.com/ArchiveMalware.aspx?language=us&name=WORMAGOBOT. XE*, accessed Jul. 2014.

[42] *T. Micro, Worm SDBot, 2003, http://about-threats.trendmicro.com/ArchiveMalware.aspx?language=us&name=WORMSDBOT. AZ*, accessed Jul. 2014.

[43] *RFC - 1549, "Internet Relay Chat Protocol", http://tools.ietf.org/html/rfc1459.html*, accessed Aug. 2014.

[44] *RFC - 2813, "Internet Relay Chat: Server Protocol", http://tools.ietf.org/html/rfc2813*, accessed Aug. 2013.

[45] D. Ha, G. Yan, S. Eidenbenz, and H. Ngo, "On the effectiveness of structural detection and defense against p2p-based botnets," in *IEEE/IFIP International Conference on Dependable Systems Networks, DSN'09*, 2009, pp. 297–306.

[46] C. Rossow, D. Andriesse, T. Werner, B. Stone-Gross, D. Plohmann, C. J. Dietrich, and H. Bos, "Sok: P2pwned — modeling and evaluating the resilience of peer-to-peer botnets," in *IEEE Symposium on Security and Privacy*, May 2013, pp. 97–111.

[47] P. Wang, S. Sparks, and C. C. Zou, "An advanced hybrid peer-to-peer botnet," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 2, pp. 113–127. doi:10.1109/TDSC.2008.35, April-June 2010.

[48] F. Pouget and M. Dacier, "Honeypot-based forensics," in *In Proceeding of AusCERT Asia Pacific Information Technology Security Conference 2004 (AusCERT2004)*, 2004.

[49] M. Hatada, Y. Nakatsuru, M. Akiyama, and S. Miwa, "Datasets for anti-malware research - mws 2010 datasets," *IPSJ Malware Workshop 2010 (MWS 2010)*, no. 1-5, 2010 (in Japanese).

[50] M. Hatada, Y. Nakatsuru, and M. Akiyama, "Datasets for anti-malware research - mws 2011 datasets," *IPSJ Malware Workshop 2011 (MWS 2011)*, no. 1-5, 2011.

[51] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *Proceedings of the 13 th Network and Distributed System Security Symposium NDSS*, Feb 2006, p. 15 pages.

[52] K. Sisaat, H. Kikuchi, S. Matsuo, M. Terada, M. Fujiwara, and S. Kittitornkun, "Time zone correlation analysis of malware/bot downloads," *IEICE Transactions on Communications*, vol. E96-B, No.07, pp. 1753–1763, July 2013.

[53] G. Mezzour, L. R. Carley, and K. M. Carley, "Global mapping of cyber attacks, cmu-isr-14-111," 2014, p. 32 Pages.

[54] I. Kononenko and M. Kukar, *MACHINE LEARNING AND DATA MINING - Introduction to Principles and Algorithms*. Horwood, 2007.

[55] S. Theodoridis and K. Koutroumbas, *Pattern Recognition - Fourth Edition*. Elsevier, 2009.

[56] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining - Practical Machine Learning Tools and Techniques*. Elsevier, 2005.

[57] L. Breiman, "Random forests, http://oz.berkeley.edu/ breiman/randomforest2001.pdf," accessed Jul. 2014.

[58] *Internet Initiative Japan Inc., "Malware Investigation Task Force", https://sect.iij.ad.jp/en/mitf.html*, accessed Jan. 2014.

[59] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon, "Peer-to-peer botnets: Overview and case study," in *HotBots'07 Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, April 2007, p. 8 pages.

[60] MWS, *Anti malware engineering workshop 2012 (MWS-2012), http://www.iwsec.org/mws/2012/about.html,*, 2012, accessed Apr. 2014.

[61] M. Hatada, Y. Nakatsuru, M. Terada, and Y. Shinoda, "Dataset for anti- malware research and research achievements shared at the workshop," *IPSJ Malware Workshop 2009 (MWS 2009)*, vol. 1-8, 2009 (in Japanese).

[62] *MaxMind, "GeoIP Databases", http://www.maxmind.com/en/city?pkit_lang=en*, accessed Jan. 2014.

[63] *Bach Seat, Conficker Worm - Still Alive, http://rbach.net/blog/index.php/conficker-worm-still-alive/,*, July 2014, accessed Oct. 2015.

[64] *F-Secure, Threat Report H1 2014, https://www.f-secure.com/documents/996508/1030743/Threat_Report_H1_2014.pdf,*, 2014, accessed Oct. 2015.

[65] S. Shin, G. Gu, N. Reddy, and C. P. Lee, "A large-scale empirical study of conficker," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 7, no. 2, pp. 676–690. doi: 10.1109/TIFS.2011.2 173 486, April 2012.

[66] G. C. M. Moura, Q. Lone, H. Asghari, and M. J. van Eeten, "Evaluating the impact of abusehub on botnet mitigation interim deliverable 1.0," Master's thesis, Delft University of Technology, March 2015.

[67] M3AAWG, *Anti-Phishing Best Practices for ISPs and Mailbox Providers*, Version 2.01, June 2015.

[68] TRIPWIRE, *SOHO WIRELESS ROUTER (IN)SECURITY*, 2014.

APPENDICES

APPENDIX A

STATISTICS AND DETAILS OF TOP MALWARE AND TOP COUNTRIES IN 2010

This appendix describes the statistics of Top downloaded malware and countries in 2010 CCC dataset. Moreover, some details of Top-30 malware and Top-30 countries are provided.

Table A.1 indicates the statistics of Top-30 malware downloads in 2010 including malware name (w), their number downloads (#Downloads), download percentage (%Downloads), and their unique IP addresses. In this reseach, Top-30 malware are analyzed and clustered. Top-30 malware account for 820,260 of total 1,162,093 downloads, representing 70% of the entire dataset. It can be seen that most of malware names contain many unique IP addresses, and only few malware names contain few unique IP. For example, WORM_AUTORUN.CZU contains only one unique IP address. That means the downloads may come from only one compromised host.

On the other hand, Table A.2 illustrates the statistics of Top-30 source countries of Top-30 malware in 2010, which contains country (c), their number downloads (#Downloads), download percentage (%Downloads), time zone, and time difference with respect to Japan (p).

Table A.3 summarizes some details of Top-10 malware/bot downloads in 2010 including malware name, their associated threats, infection, and their aliases.

Table A.1 Statistics of Top-30 malware downloads in 2010 CCC dataset

Top	Malware Name, w	#Downloads $\sum_d l_w(d)$	%Downloads	Unique IPs
1	PE_VIRUT.AV	194,557	16.7	37,481
2	BKDR_VANBOT.RG	83,757	7.2	6,851
3	WORM_AUTORUN.CZU	46,313	4.0	1
4	WORM_RBOT.SMA	36,171	3.1	26,160
5	TROJ_BUZUS.BEZ	32,172	2.8	2,230
6	WORM_KOLABC.ET	31,967	2.8	2,119
7	BKDR_RBOT.ASA	31,404	2.7	23,744
8	BKDR_NEPOE.CW	30,118	2.6	1,474
9	WORM_KOLAB.EA	28,909	2.5	4
10	WORM_KOLAB.CV	28,586	2.5	2
11	TROJ_ADCQ.A	27,985	2.4	-
12	BKDR_VANBOT.AHH	26,709	2.3	-
13	TROJ_IRCBRUTE.BW	26,256	2.3	-
14	WORM_PALEVO.SMJF	21,068	1.8	-
15	WORM_PALEVO.SMD	15,107	1.3	-
16	BKDR_NEPOE.DM	14,721	1.3	-
17	WORM_PALEVO.AZ	13,333	1.1	-
18	WORM_ALLAPLE.IK	12,559	1.1	-
19	TROJ_MALWARE.VTG	11,881	1.0	-
20	PE_VIRUT.PAU	11,811	1.0	-
21	WORM_MAINBOT.MCL	11,298	1.0	1,750
22	BKDR_MYBOT.AH	10,497	0.9	-
23	BKDR_VANBOT.HG	10,406	0.9	-
24	WORM_PALEVO.BE	9,557	0.8	-
25	Mal_Swzr-3	9,215	0.8	-
26	WORM_PALEVO.AK	9,061	0.8	-
27	WORM_SPYBOT.AWX	9,020	0.8	-
28	WORM_DOWNAD.AD	9,015	0.8	-
29	TSPY_ONLINEG.TKJ	8,454	0.7	-
30	PE_VIRUT.XV	8,420	0.7	-

Table A.2 Top-30 source countries of Top-30 malware in 2010 CCC dataset

Top	Country (Code), c	#Downloads, $\sum_d l_c(d)$	%Downloads	Time Zone (UTC/GMT)	Time Diff. wrt Japan, p
1	JAPAN (JP)	399,641	48.86	+9	0
2	CANADA (CA)	235,375	28.78	-5	14
3	USA (US)	67,459	8.25	-5	14
4	TAIWAN (TW)	36,886	4.51	+8	1
5	UKRAINE (UA)	35,655	4.36	+2	7
6	CHINA (CN)	17,634	2.16	+8	1
7	INDIA (IN)	3,572	0.44	+5	4
8	PHILIPPINES (PH)	3,476	0.43	+8	1
9	THAILAND (TH)	2,212	0.27	+7	2
10	AUSTRALIA (AU)	1,783	0.22	+10	-1
11	HONG KONG (HK)	1,710	0.21	+8	1
12	MALAYSIA (MY)	1,663	0.20	+8	1
13	RUSSIAN (RU)	1,576	0.19	+4	5
14	SINGAPORE (SG)	1,542	0.19	+8	1
15	KOREA (KR)	1,238	0.15	+9	0
16	NEW ZEALAND (NZ)	1,055	0.13	+12	-3
17	INDONESIA (ID)	773	0.09	+8	1
18	ROMANIA (RO)	550	0.07	+2	7
19	GERMANY (DE)	455	0.06	+1	8
20	ITALY (IT)	436	0.05	+1	8
21	PAKISTAN (PK)	378	0.05	+5	4
22	VIETNAM (VN)	371	0.05	+7	2
23	BRUNEI (BN)	368	0.04	+8	1
24	NEW CALEDONIA (NC)	368	0.04	+11	-2
25	BRAZIL (BR)	347	0.04	-3	12
26	POLAND (PL)	345	0.04	+1	8
27	ARGENTINA (AR)	311	0.04	-3	12
28	HUNGARY (HU)	261	0.03	+1	8
29	NEPAL (NP)	214	0.03	+6	3
30	BULGARIA (BG)	197	0.02	+2	7

Table A.3 Rank, name, associated threats, infection and aliases of Top-10 malware in 2010

Top	Name	Threats	Infection	Aliases
1	PE_VIRUT.AV	Exploit, download additional malicious programs, load into the memory, compromise privacy and slash computer security. A network-aware worm that attempts to replicate across the existing network(s)	Downloaded from the Internet, dropped by other malware, MS04-012: DCOM RPC Overflow exploit - replication across TCP 135/139/445/593.	W32.Rahack.H (SAV), Virus.Win32.Virut.av (KAV), Win32/Virut.AC (MSE), Net-Worm.Win32.Allaple (IKU), Win32/Virut.B (AOS)
2	BKDR_VANBOT.RG	Spreads via networks and infect other computers as Net-Worms do. Do not spread automatically (as Net-Worms do), but only upon a special command from the malicious user that controls them.	Dropped by other malware or as a file downloaded unknowingly by users when visiting malicious sites.	W32.IRCBot (SAV), Backdoor.Win32.VanBot.bdt (KAV), Backdoor.Win32.EggDrop.bmg (SKPF)

3	WORM_AUTORUN.CZU	Do not infect files, but may carry one or more payloads, such as computer security compromise and information theft. Modifies system settings to automatically start. Users may need to terminate worms before they can be deleted.	Propagates and spreads across networks using one or several of different transmission vectors like email, IRC, network shares, instant messengers (IM), and peer-to-peer (P2P) networks.	P2P-Worm.Win32.Palevo.brx (KAV), W32.SillyFDC (SAV), Worm/Agent.W.45 (AAV)
4	WORM_RBOT.SMA	Do not infect files, but may carry one or more payloads, such as computer security compromise and information theft. Modifies System settings to automatically start. Users may need to terminate worms before they can be deleted.	It may be dropped by other malware, propagates and spreads across networks using one or several of different transmission vectors like email, IRC, network shares, instant messengers (IM), and peer-to-peer (P2P) networks.	Win32.HLLW.MyBot (DWAV), BackDoor-DZP (McAfee), Backdoor.Win32.Rbot.adz(KAV), Trojan.Win32.Ircbrute (SKPF), Win32/Rbot.gen (MSE), W32.IRCBot.Gen (SAV)

5	TROJ_BUZUS.BEZ	Carries payloads or other malicious actions that can range from the mildly annoying to the irreparably destructive. Modifies system settings to automatically start. Restoring affected systems may require procedures other than scanning with an antivirus program.	It may be dropped by other malware or as a file downloaded unknowingly by users when visiting malicious sites. It drops copies of itself into the affected system:%System Root%\WIN\DOWS\LAX.exe	Win32/Vbinder.gen!GL(MSE), Generic.dx!dqv(McAfee), Trojan Horse (SAV), Trojan.Win32.VB.umo(KAV), Virtool.Win32.Vbinject.1(SKPF), Trojan horse VBCrypt.CZL(AVG)
6	WORM_KOLABC.ET	Perform DoS attacks against other computers. Replicates across networks by exploiting weakly restricted shares (common for Randex family of worms). It is a network worm and backdoor for the Windows platform. It allows a malicious user remote access to an infected computer via IRC.	It may be dropped by other malware or as a file downloaded unknowingly by users when visiting malicious sites. It drops the following copies of itself into the affected system: %Windows%\Fonts\unwise_.exe	Win32/MS08067.gen!A (MSE), W32/Kolab (McAfee), W32.Spybot.Worm (SAV), Net-Worm.Win32.Kolabc.hki (KAV), BehavesLike.Win32.Malware.eah (SKPF), Win32.Worm.Kolabc.V (FSAV)

7	BKDR_RBOT.ASA	Allows malicious users to remotely manipulate affected systems. It modifies system settings to automatically start. Users may need to terminate worms before they can be deleted.	It is installed either inadvertently by unsuspecting users or intentionally by malicious users (using net-Worms as tool).	Backdoor.Win32.Rbot.rax (KAV), W32/Sdbot.worm.gen.x (McAfee), W32.Spybot.Worm (SAV), W32/Trojan5.DCW (exact) (FP)
8	BKDR_NEPOE.CW	Modifies system settings to automatically start. It runs in the background and allows remote access to the system, giving the attacker full control.	To replicate across vulnerable networks. MS04-012: DCOM RPC Overflow exploit - replication across TCP 135/139/445/593 MS04-011: LSASS Overflow exploit - replication across TCP 445	Backdoor.IRCBot!sd5 (PCTAV), W32.IRCBot (SAV), Generic.dx (McAfee)
9	WORM_KOLAB.EA	A malicious or bot that drops the copies of itself into the affected system: %System Root%\RECYCLER\S-1-5-21-0243556031-888888379-781863308-1455\fjidg.exe. It downloads/requests other files from Internet.	This worm arrives on a system as a file dropped by other malware or as a file downloaded unknowingly by users when visiting malicious sites.	Win32/Lethic.H(MSE), Generic.ge(McAfee), Trojan.Gen (SAV), Trojan.Win32.Buzus.ctfx(KAV), Trojan.Win32.Buzus.bzaz (SKPF), Worm:W32/Palevo.gen!J(FSAV)

10	WORM_KOLAB.CV	Computer will become increasingly slow, email account may start to act strange, Emails may be sent to unfamiliar people, computer setting may suddenly change, feel like somebody is watching, just like a ghost on the computer.	Infects through subnet and cross network. Spreads itself through a variety of methods including P2P networks. Dropped by other malware or as a file downloaded unknowingly by users when visiting malicious sites.	Win32/DelfInject.gen!BD (MSE), BackDoor-DOQ.gen.z (McAfee), W32.Spybot.Worm (SAV), Net-Worm.Win32.Kolab.ffo (KAV), NetWorm.Win32.Kolab.ffo (SKPF)

APPENDIX B

STATISTICS AND DETAILS OF TOP MALWARE AND TOP COUNTRIES IN 2012

This appendix gives a brief description of the statistics and details of Top-20 most downloaded malware and countries in 2012 IIJ MITF dataset. In addision, some details of Conficker.B and Conficker.C as aliases of the two major clusters of Top-20 malware clustering are summarized.

Table B.1 presents the statistics of Top-20 malware downloads in 2012, which contains number of downloads (#Downloads), download percentage (%Download), and number of unique hashes (SHA1) of each unique malware name. It can be observed that the most frequently downloaded Top-20 malware accounts for 17,926,169 of total 32,070,143 downloads, representing 55% of the entire dataset.

Table B.1 Statistics of Top-20 malware in Top-20 source countries downloads of 2012 IIJ MITF dataset

Top	Malware Name, w	#Downloads, $\sum_d l_w(d)$	%Downloads	Unique Hashes
1	Trojan.Dropper-18535	3,738,829	23.4	2,540
2	Worm.Kido-20	1,766,252	11.0	514
3	Worm.Kido-102	1,451,967	9.1	646
4	Worm.Kido-99	1,169,602	7.3	437
5	Trojan.Agent-71049	961,731	6.0	270
6	Worm.Agent-194	868,378	5.4	331
7	Worm.Kido-367	642,253	4.0	173
8	Worm.Kido-182	617,509	3.9	2
9	Trojan.Agent-71068	557,115	3.5	132
10	Worm.Kido-24	498,357	3.1	217
11	Worm.Kido-119	497,163	3.1	167
12	Worm.Kido-223	492,787	3.1	188
13	Worm.Kido-25	481,447	3.0	2
14	Worm.Kido-128	425,643	2.7	102
15	Worm.Kido-175	339,416	2.1	1
16	Worm.Kido-85	328,762	2.1	219
17	Worm.Kido-51	322,627	2.0	97
18	Worm.Downadup-113	305,763	1.9	273
19	Trojan.Agent-71228	277,149	1.7	166
20	Worm.Kido-295	245,161	1.5	144

On the other hand, Table B.2 provides the statistics of Top-20 source countries including country (c), their number downloads (#Downloads), download percentage (%Download), time zone, and time difference with respect to Japan (p). The number downloads of Top-20 countries

amounts to 83% of Top-20 malware total downloads. Please note that to identify the source country of a given IP address, GeoIP Databases [62], a commercial geographical IP (GeoIP) service that provides information including country code, city name and so on has been used.

Table B.2 Top-20 source countries of Top-20 malware in 2012 IIJ MITF dataset

Top	Country (Code), c	#Downloads, $\sum_d l_c(d)$	%Downloads	Time Zone (UTC/GMT)	Time Diff. wrt Japan, p
1	RUSSIA (RU)	3,224,066	20.2	+4	5
2	TAIWAN (TW)	2,541,603	15.9	+8	1
3	USA (US)	1,996,188	12.5	-5	14
4	BRAZIL (BR)	1,468,410	9.2	-3	12
5	ROMANIA (RO)	939,753	5.9	+2	7
6	HUNGARY (HU)	596,561	3.7	+1	8
7	BULGARIA (BG)	586,321	3.7	+2	7
8	JAPAN (JP)	503,315	3.1	+9	0
9	POLAND (PL)	469,245	2.9	+1	8
10	KOREA (KR)	463,310	2.9	+9	0
11	ITALY (IT)	451,485	2.8	+1	8
12	ARGENTINA (AR)	434,514	2.7	-3	12
13	GERMANY (DE)	377,545	2.4	+1	8
14	CHINA (CN)	331,364	2.1	+8	1
15	UKRAINE (UA)	308,955	1.9	+2	7
16	ISRAEL (IL)	293,313	1.8	+2	7
17	INDIA (IN)	292,117	1.8	+5	4
18	CANADA (CA)	265,543	1.7	-5	14
19	TURKEY (TR)	257,814	1.6	+2	7
20	FRANCE (FR)	186,489	1.2	+1	8

Table B.3 sommarizez some details of Conficker.B and Conficker.C in 2012 including their detection date, threat behavior, infection vectors, and update propagation.

Table B.3 Summary of Conficker.B & Conficker.C in 2012

Conficker	Detection date	Threat behavior	Infection vectors	Update propagation
B	2008-12-29	Conficker.B tries to copy itself in the Windows system folder as a hidden DLL file using a random name. If it fails, it can then try to copy itself with the same parameters in the folders %ProgramFiles% \Internet Explorer and %ProgramFiles% \Movie Maker. It creates the following registry entry to ensure that its dropped copy is run every time Windows starts.	(1)NetBIOS: Exploits MS08-067 vulnerability in Server service, and Dictionary attack on ADMIN$ shares. (2)Removable media: Creates DLL-based AutoRun trojan on attached removable drives	(1)HTTP pull: Downloads daily from any of 250 pseudorandom domains over 8 TLDs. (2)NetBIOS push: Patches MS08-067 to open reinfection backdoor in Server service.

C	2009-02-20	Conficker.C is a worm that infects other computers across a network by exploiting a vulnerability in the Windows Server service (svchost.exe). If the vulnerability is successfully exploited, it could allow remote code execution when file sharing is enabled. It may also spread via removable drives and weak administrator passwords. It disables several important system services and security products.	(1)NetBIOS: Exploits MS08-067 vulnerability in Server service, and Dictionary attack on ADMIN$ shares. (2)Removable media: Creates DLL-based AutoRun trojan on attached removable drives.	(1)HTTP pull: Downloads daily from 500 of 50,000 pseudorandom domains over 8 TLDs per day. (2)NetBIOS push: Patches MS08-067 to open reinfection backdoor in Server service, and Creates named pipe to receive URL from remote host, then downloads from URL.

APPENDIX C

SUMMARY OF MALWARE AND COUNTRY HIERARCHICAL CLUSTERING IN 2010

This appendix provides a summary of malware and country hierarchical clustering in 2010 with a variety of feature (i.e., temporal, spatio/malware, and temporal+spatio/malware), dissimilarity (i.e., correlation and cosine), and linkage (i.e., single, average, and complete) options.

C.1 Summary of Malware Clustering in 2010

Malware hierarchical clustering in 2010 with a variety of dissimilarity (i.e., correlation and cosine), linkage (i.e., single, average, and complete), and feature such as temporal (weekly + hourly) only, spatio (country) only, and temporal + spatio are illustrated in Table C.1, Table C.2, and Table C.3, respectively.

Table C.1 Summary of hierarchical malware clustering in 2010 with Temporal-only feature, dissimilarity and linkage options.

Linkage	Dissimilarity	Temporal (Weekly+Hourly)	
		Correlation	Cosine
Single	$0.3 < x \leq 0.4$	[5,9,19,27] [21,24] [15,25]	[1-13,15,17-25,27-30]
		[2,30,20,18,1,4,22,10,	-
		3,6,7,8,11,12,13,23,29]	
	$0.4 < x \leq 0.5$	[5,9,19,27] [21,24]	[1-30]
		[1,2,3,4,6,7,8,10,11,12,13,15,	-
		17,18,20,22,23,25,26,29,30	
Average	$0.3 < x \leq 0.4$	[5,9,19,27] [15,25] [1,4,20,22]	[5,9,19,27] [21,24] [15,25]
		[21,24] [11,13] [2,30] [7,18]	[11,13] [1,2,4,18,20,22,30]
		[3,6,8,10,12,23,29]	[3,6,7,8,10,12,23,29]
	$0.4 < x \leq 0.5$	[5,9,19,27] [15,25] [1,4,20,22]	[5,9,19,27] [21,24] [15,25,28]
		[21,24] [11,13] [2,30,7,18]	[1,2,3,4,6,7,8,10,11,12,
		[3,6,8,10,12,23,29]	13,18,20,22,23,29,30]
Complete	$0.3 < x \leq 0.4$	[5,9,19,27] [15,25] [4,22]	[21,24] [5,9,19,27] [15,25]
		[21,24] [11,13] [1,18] [2,30]	[1,4,18,20,22] [2,7,30] [11,13]
		[3,6,8,10,12,23,29]	[3,6,8,10,12,23,29]
	$0.4 < x \leq 0.5$	[5,9,19,27] [15,25] [4,20,22]	[21,24] [5,9,19,27] [15,25]
		[21,24] [11,13] [1,18] [2,7,30]	[1,4,18,20,22,28] [2,7,30] [11,13]
		[3,6,8,10,12,23,29]	[3,6,8,10,12,23,29]

Table C.2 Summary of hierarchical malware clustering in 2010 with Spatial-only feature, dissimilarity and linkage options.

Linkage	Dissimilarity	Spatial (Country)	
		Correlation	Cosine
Single	$0.3 < x \leq 0.4$	[1-20 & 22-30]	[All except 21]
		-	-
	$0.4 < x \leq 0.5$	[1-20 & 22-30]	[All except 21]
		-	-
Average	$0.3 < x \leq 0.4$	[2,3,5,9,12,19,23,27,29] [11,30]	[2,3,5,9,12,19,23,27,29] [11,30]
		[6,7,17] [1,4,8,10,13,14,15,	[6,7,17] [1,4,8,10,13,14,15,
		16,18,20,22,24,25,26,28]	16,18,20,22,24,25,26,28]
	$0.4 < x \leq 0.5$	[2,3,5,9,12,19,23,27,29] [11,30]	[2,3,5,9,12,19,23,27,29]
		[6,7,17] [1,4,8,10,13,14,15,	[6,7,11,17,30] [1,4,8,10,13,14,15,
		16,18,20,22,24,25,26,28]	16,18,20,22,24,25,26,28]
Complete	$0.3 < x \leq 0.4$	[2,3,5,9,12,19,23,27,29] [11,30]	[2,3,5,9,12,19,23,27,29] [11,30]
		[6,7,17] [14,18,26] [1,4,8,10,	[6,7,17] [14,18,26] [1,4,8,10,
		13,15,16,20,22,24,25,28]	13,15,16,20,22,24,25,28]
	$0.4 < x \leq 0.5$	[2,3,5,9,12,19,23,27,29] [11,30]	[2,3,5,9,12,19,23,27,29] [11,30]
		[6,7,17] [1,4,8,10,13,14,15,	[6,7,17] [1,4,8,10,13,14,15,
		16,18,20,22,24,25,26,28]	16,18,20,22,24,25,26,28]

Table C.3 Summary of hierarchical malware clustering in 2010 with Temporal+Spatial feature, dissimilarity and linkage options.

Linkage	Dissimilarity	Temporal+Spatial	
		Correlation	Cosine
Single	$0.3 < x \leq 0.4$	[All except 11,14,21]	[All except 14,21]
		-	-
	$0.4 < x \leq 0.5$	[All except 14,21]	[All except 21]
		-	-
Average	$0.3 < x \leq 0.4$	[2,5,9,12,19,23,27,29]	[2,5,9,12,19,23,27,29]
		[6,7,17] [26,30][1,4,8,10,13,	[3,8,10] [6,7,17] [26,30]
		16,18,20,22,24,25,28]	[1,4,13,16,18,20,22,24,25,28]
	$0.4 < x \leq 0.5$	[3,2,5,9,12,19,23,27,29]	[2,5,9,12,19,23,27,29]
		[6,7,17] [26,30] [1,4,8,10,13,	[6,7,17] [26,30] [1,3,4,8,10,13,15,
		15,16,18,20,22,24,25,28]	16,18,20,22,24,25,28]
Complete	$0.3 < x \leq 0.4$	[15,25,28] [1,4,16,18,20,22,24]	[5,9,19] [2,12,23,27,29] [6,7,17]
		[5,9,19] [2,12,23,27,29]	[26,30] [3,8,10,13] [15,25,28]
		[6,7,17] [26,30] [8,10] [3,13]	[1,4,14,16,18,20,22]
	$0.4 < x \leq 0.5$	[15,25,28] [1,4,16,18,20,22,24]	[2,5,9,12,19,23,27,29] [6,7,17]
		[5,9,19] [2,12,23,27,29]	[26,30] [3,8,10,13] [15,25,28]
		[6,7,17] [26,30] [3,8,10,13]	[1,4,14,16,18,20,22]

C.2 Summary of Country Clustering in 2010

Country hierarchical clustering in 2010 with a variety of dissimilarity (i.e., correlation and cosine), linkage (i.e., single, average, and complete), and feature such as temporal (weekly + hourly) only, malware only, and temporal + malware are illustrated in Table C.4, Table C.5, and Table C.6, respectively.

Table C.4 Summary of hierarchical country clustering in 2010 with Temporal-only feature, dissimilarity and linkage options.

Linkage	Dissimilarity	Temporal (Weekly+Hourly)	
		Correlation	Cosine
Single	$0.3 < x \leq 0.4$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[All except NC,NP,UA,US]
		[AU,BN,CA,CN,HK,	-
		IN,JP,KR,MY,PH,SG,TW]	
	$0.4 < x \leq 0.5$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[All]
		[AU,BN,CA,CN,HK,ID,IN,JP,KR,	-
		MY,NZ,PH,PK,SG,TH,TW,UA,VN]	
Average	$0.3 < x \leq 0.4$	[BN,CA] [CN,HK,IN,JP,KR,PH,TW]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AU,BN,CA,CN,HK,ID,IN,JP,
	$0.4 < x \leq 0.5$	[TH,UA] [BN,CA] [AU,SG]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[CN,HK,IN,JP,KR,MY,PH,TW]	[AU,BN,CA,CN,HK,ID,IN,JP,
		[AR,BG,BR,DE,HU,IT,PL,RO,RU]	KR,MY,NZ,PH,PK,SG,TH,TW,VN]
Complete	$0.3 < x \leq 0.4$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[BN,CA] [IN,JP,KR]	[BN,CA,ID] [AU,CN,HK,IN,JP,
		[CN,HK,PH,TW]	KR,MY,NZ,PH,PK,SG,TH,TW]
	$0.4 < x \leq 0.5$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[CA,BN] [TH,UA] [IN,JP,KR]	[NC,US] [AU,BN,CA,CN,HK,ID,IN,
		[AU,SG] [CN,HK,MY,NZ,PH,TW]	JP,KR,MY,NZ,PH,PK,SG,TH,TW,VN]

Table C.5 Summary of hierarchical country clustering in 2010 with Malware-only feature, dissimilarity and linkage options.

Linkage	Dissimilarity	Malware	
		Correlation	Cosine
Single	$0.3 < x \leq 0.4$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[AU,CA,CN,HK,ID,IN,JP,KR,	[AU,BN,CA,CN,HK,ID,IN,JP,
		MY,NZ,PH,PK,SG,TH,TW,VN]	KR,MY,NZ,PH,SG,TH,TW,SG,VN]
	$0.4 < x \leq 0.5$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[AU,CA,CN,HK,ID,IN,JP,KR,	[AU,BN,CA,CN,HK,ID,IN,JP,
		MY,NZ,PH,PK,SG,TH,TW,VN]	KR,MY,NZ,PH,SG,TH,TW,SG,VN]
Average	$0.3 < x \leq 0.4$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[JP,TH] [HK,KR,MY,NZ,PH,TW]	[JP,TH] [HK,KR,MY,NZ,PH,TW]
		[AU,CN,ID,IN,PK,SG,VN]	[AU,CN,ID,IN,PK,SG,VN]
	$0.4 < x \leq 0.5$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[JP,TH] [HK,KR,MY,NZ,PH,TW]	[JP,TH] [HK,KR,MY,NZ,PH,TW]
		[AU,CN,ID,IN,PK,SG,VN]	[AU,CN,ID,IN,PK,SG,VN]
Complete	$0.3 < x \leq 0.4$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[JP,TH] [HK,KR,MY,NZ,PH,TW]	[JP,TH] [HK,KR,MY,NZ,PH,TW]
		[AU,CN,PK,SG] [ID,IN,VN]	[AU,PK,SG] [CN,ID,IN,VN]
	$0.4 < x \leq 0.5$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[JP,TH] [HK,KR,MY,NZ,PH,TW]	[JP,TH] [HK,KR,MY,NZ,PH,TW]
		[AU,CN,ID,IN,PK,SG,VN]	[AU,CN,ID,IN,PK,SG,VN]

Table C.6 Summary of hierarchical country clustering in 2010 with Temporal+Malware feature, dissimilarity and linkage options.

Linkage	Dissimilarity	Temporal+Malware	
		Correlation	Cosine
Single	$0.3 < x \leq 0.4$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[AU,CN,HK,ID,IN,JP,KR,	[AU,BN,CA,CN,HK,ID,IN,JP,
		MY,NZ,PH,PK,SG,TH,TW,VN]	KR,MY,NZ,PH,PK,SG,TH,TW,VN]
	$0.4 < x \leq 0.5$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[All]
		[AU,BN,CA,CN,HK,ID,IN,JP,	[AU,BN,CA,CN,HK,ID,IN,JP,
		KR,MY,NZ,PH,PK,SG,TH,TW,VN]	KR,MY,NZ,PH,PK,SG,TH,TW,VN]
Average	$0.3 < x \leq 0.4$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[AU,CN,PK,SG] [ID,IN] [JP,TH]	[JP,TH] [HK,KR,MY,NZ,PH,TW]
		[HK,KR,MY,NZ,PH,TW]	[AU,CA,CN,ID,IN,PK,SG,VN]
	$0.4 < x \leq 0.5$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[JP,TH] [HK,KR,MY,NZ,PH,TW]	[AU,CA,CN,HK,ID,IN,JP,KR,
		[AU,CN,ID,IN,PK,SG,VN]	MY,NZ,PH,PK,SG,TH,TW,VN]
Complete	$0.3 < x \leq 0.4$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[JP,TH] [HK,KR,MY,NZ,PH,TW]	[JP,TH] [HK,KR,MY,NZ,PH,TW]
		[AU,CN,PK,SG] [ID,IN]	[AU,PK,SG] [CA,CN,ID,IN,VN]
	$0.4 < x \leq 0.5$	[AR,BG,BR,DE,HU,IT,PL,RO,RU]	[AR,BG,BR,DE,HU,IT,PL,RO,RU]
		[JP,TH] [HK,KR,MY,NZ,PH,TW]	[JP,TH] [HK,KR,MY,NZ,PH,TW]
		[CA,VN] [AU,CN,ID,IN,PK,SG]	[AU,CA,CN,ID,IN,PK,SG,VN]

BIOGRAPHY

Personal Information

Name	Khamphao Sisaat
Sex	Male
Nationality	Lao
Date of Birth	November 20, 1977 at Sayaboury Province, Laos
Office Address	Department of Computer Engineering and Information Technology, Faculty of Engineering, National University of Laos Lao-Thai Road, Sisattanak District, P.O.Box 4242, Vientiane, Laos. Tel: (+856-21) 350960

Education

*1996-1999, Higher Diploma of Eng. in Electrical Engineering (National University of Laos, Laos).

*1999-2001, Bachelor of Eng. in Computer Engineering (King Mongkut's Institute of Technology Ladkrabang, Thailand).

*2004-2006, Master of Eng. in Information Technology (Nara Institute of Science and Technology, Japan).

Research Interests

Network security, Intrusion Detection/Prevention System (IDS/IPS), trace-back technologies, network forensics, and other countermeasures against Denial of Service (DoS) and Distributed DoS (DDoS) attacks, as well as countermeasures against malicious software attacks.

Work Experience

In 2001, I joined the National University of Laos as a Lecturer and teached following subjects:

System Administration (Windows/Linux System), Information Security, Wireless Technologies, Computer Network Design, LAN and WAN (Cisco CCNA Discovery - CCNA D1 and CCNA D2).

Other Certificate

CISCO CCNA Discovery (D1-D4) and CCNA Security Instructor courses in Laos in 2010 and 2011, respectively.

LIST OF PUBLICATIONS

Some parts of this work are published in the following articles.

Domestic Conference Proceedings

1. **Khamphao Sisaat**, Hiroaki Kikuchi, Surin Kittitornkun, Chaxiong Yukonhiatou, Masato Terada, and Hiroshi Ishii, "Time Zone Analysis on IIJ Network Traffic for Malicious Botnet Activities", IEICE Technical Report, ICSS2013-52, Vol. 113 No. 135, JULY 2013, pp. 373-380.

International Conference Proceedings

1. Naoki Hiroguchi, **Khamphao Sisaat**, Hiroaki Kikuchi, and Surin Kittitornkun, "Geographical Visualization of Malware Download for Anomaly Detection", 2012 Seventh Asia Joint Conference on Information Security, 9-10 Aug. 2012, pp. 74-78.
2. Chaxiong Yukonhiatou, Surin Kittitornkun, Hiroaki Kikuchi, **Khamphao Sisaat**, Masato Terada and Hiroshi Ishii, "Temporal Behavior Analysis of Malware/Bot Downloads Using Top-10 Processing", The 2013 International Computer Science and Engineering Conference (ICSEC 2013), Nakorn Pathom, THAILAND, September 4 - 6, 2013.
3. Chaxiong Yukonhiatou, Surin Kittitornkun, Hiroaki Kikuchi, **Khamphao Sisaat**, Masato Terada and Hiroshi Ishii, "Clustering Top-10 Malware/Bots based on Download Behavior", The 5th International Conference on Information Technology and Electrical Engineering (ICITEE 2013), THE SAHID RICH JOGJA HOTEL,Yogyakarta, Indonesia, Oct 7, 2013 - Oct 8, 2013.
4. Chaxiong Yukonhiatou, Surin Kittitornkun, Hiroaki Kikuchi, **Khamphao Sisaat**, Masato Terada and Hiroshi Ishii, "Temporal Behaviors of Top-10 Malware Download in 2010-2012", The 2014 International Electrical Engineering Congress (iEECON 2014), March 19-21, 2014, Pattaya City, Thailand.

International Journal Papers

1. **Khamphao Sisaat**, Hiroaki Kikuchi, Shunji Matsuo, Masato Terada, Masashi Fujiwara, and Surin Kittitornkun, "Time Zone Correlation Analysis of Malware/Bot Downloads", IEICE TRANS. COMMUN., VOL.E96-B, NO.7, JULY 2013, pp. 1753-1763.

2. **Khamphao Sisaat**, Surin Kittitornkun, Hiroaki Kikuchi, Chaxiong Yukonhiatou, Masato Terada, and Hiroshi Ishii, "A Spatio-Temporal malware and country clustering algorithm: 2012 IIJ MITF case study", International Journal of Information Security, JULY 2016, pp. 1-15.