

A Large-Scale Empirical Study of Conficker

Seungwon Shin, *Student Member, IEEE*, Guofei Gu, *Member, IEEE*,
Narasimha Reddy, *Fellow, IEEE*, and Christopher P. Lee, *Member, IEEE*

Abstract—Conficker [34] is the most recent widespread, well-known worm/bot. According to several reports [21], [36], it has infected about 7 million to 15 million hosts and the victims are still increasing even now. In this paper, we analyze Conficker infections at a large scale, about 25 millions victims, and study various interesting aspects about this state-of-the-art malware. By analyzing Conficker, we intend to understand current and new trends in malware propagation, which could be very helpful in predicting future malware trends and providing insights for future malware defense. We observe that Conficker has some very different victim distribution patterns compared to many previous generation worms/botnets, suggesting that new malware spreading models and defense strategies are likely needed. We measure the potential power of Conficker to estimate its effects on the networks/hosts when it performs malicious operations. Furthermore, we intend to determine how well a reputation-based blacklisting approach can perform when faced with new malware threats such as Conficker. We cross-check several DNS blacklists and IP/AS reputation data from Dshield [9] and FIRE [10], and our evaluation shows that unlike a previous study [25] which shows that a blacklist-based approach can detect most bots, these reputation-based approaches did relatively poorly for Conficker. This raises a question of how we can improve and complement existing reputation-based techniques to prepare for future malware defense? Based on this, we look into some insights for defenders. We show that neighborhood watch is a surprisingly effective approach in the case of Conficker. This suggests that security alert sharing/correlation (particularly among neighborhood networks) could be a promising approach and play a more important role for future malware defense.

Index Terms—Botnet, Conficker, Botnet measurement, and Botnet defense.

I. INTRODUCTION

Conficker worm [34] first appeared in November 2008 and rapidly spread in the world within a short period. It exploits a NetBIOS vulnerability in various Windows operating systems and utilizes many new, advanced techniques such as a domain generation algorithm, self-defense mechanisms, updating via Web and P2P, and efficient local propagation. As a result, it has infected millions of victims in the world and the number is still increasing even now [21], [36].

It is clear that the complex nature of Conficker makes it one of the state-of-the-art botnets, and therefore the analysis of Conficker is very important in order to defend against it. A full understanding of Conficker can also help us comprehend current and future malware trends. Existing research of Conficker analysis mainly falls into two categories. The first focuses on analyzing the Conficker binary and its behavior,

revealing its malicious tricks such as the domain generation algorithm [30], [38]. In this direction, SRI researchers [30] and the HoneyNet project [38] already provided excellent reports that analyzed Conficker in great detail. The second research category mainly focuses on analyzing the network telescope data [5] or DNS sinkhole data [17] to reveal the propagation pattern and victim distribution characteristics of Conficker on the Internet. There are very few studies in this direction, which is probably because it is very hard to obtain large scale real-world data of victims and the amount of data should be large enough to cover victims' global behavior. CAIDA [5] and Team Cymru [17] provided some initial reports which contain some very basic statistics on the scanning pattern and propagation information of Conficker. However, for a worm/bot that has infected so many victims and has so much potential to damage the Internet, it deserves a much deeper study. Such study is necessary because by analyzing this state-of-the-art botnet, we can gain more knowledge of current malware, e.g., how it differs from previous generation malware and whether such differences represent future trends or not. These deeper investigations could also provide new insights in developing new detection and defense mechanisms for current and future malware.

In this paper, we attempt to provide a deeper empirical measurement study of Conficker. We have collected a large-scale data set which contains almost 25 million Conficker victims with the help of *Shadowserver.org* (details on data collection are discussed in Section III). We believe such scale is large enough to uncover Conficker's global patterns. We provide an extensive measurement of various distribution patterns of Conficker victims. Furthermore, we use a comparison- and cross-check-based methodology in our measurement study. We study the similarities and differences between Conficker and several other publicly reported worms/botnets. Then we analyze how these differences may affect existing reputation-based detection approaches. We also investigate possible aspects that may be useful for Conficker and future malware defense.

In short, this paper makes the following contributions:

- We provide a large-scale empirical study of almost 25 million Conficker victims. By analyzing this data, we reveal many interesting aspects that were previously unknown and show that Conficker victims exhibit a very different distribution pattern from many previously reported botnets or worms. This difference could be a new trend or some ignored facts that are potentially important for future malware defense. Detailed information is in Section IV.
- We evaluate the effectiveness of existing reputation-based approaches for detecting emerging malware threats.

A preliminary version of this paper appeared in [28]

Seungwon Shin, Guofei Gu, and Narasimha Reddy are with Texas A&M University, College Station, TX 77840. E-mail: seungwon.shin@neo.tamu.edu, guofei@cse.tamu.edu, reddy@ece.tamu.edu. Christopher P. Lee is with ISC/SIE and his email is chrisl@isc.org.

They are considered as promising in defending against unknown malware compared to traditional signature-based approaches [3]. Through cross-checking several DNS blacklists and reputation data from Dshield [9] and FIRE [10], our evaluation shows that these reputation-based approaches are not effective for Conficker defense. In addition, we investigate whether there are differences/similarities between victims detected by them and victims not detected by them. Our study suggests that these reputation-based approaches need to be significantly improved and complemented by other techniques. Detailed information is in Section V.

- We measure the potential power of Conficker to predict how much network traffic they can produce or how much information they can steal. For example, we discover that Conficker can generate nearly 2 Tbps network traffic even when several reputation-based detection systems filter some of them. Moreover, we reveal that it is possible for a botmaster of Conficker to run huge amount of malicious sites with minimal detection.
- We study the Conficker data and find that neighborhood watch is surprisingly effective to infer new victims. This could suggest that alert sharing/correlation (among distributed collaborators, particularly neighborhood networks) could be an effective and promising technique to defend against future emerging threats and it needs more attention for such research. Detailed information is in Section VII.

II. RELATED WORK

Conficker binary analysis. Porras et al. from SRI International provided a very extensive study of the Conficker binary analysis [30]. They analyzed several variants of Conficker and revealed how Conficker propagates, how it infects others, how it evades anti-virus tools and how it updates itself. This provided very detailed and valuable information of Conficker behavior. The HoneyNet project [38] also provides a detailed analysis of Conficker binary. These studies also provide scanning tools for detecting Conficker victims in the network.

Conficker data analysis. With the use of the telescope data, researchers from CAIDA provided a simple analysis on Conficker propagation [5]. The Telescope data mainly contains scanning traffic from Conficker victims, which reveals Conficker victim location and timing information to display how Conficker emerges and spreads on the Internet. However, such data is not complete due to the size limit of (passive) monitoring networks. Recently, researchers started to use the DNS sinkholing technique [17] to collect much more accurate Conficker victim data. A report from Team Cymru [17] analyzed the behavior of Conficker victims and provided some general distribution and propagation information. However, there is still a lack of some deep analysis of Conficker victims such as how different the victims are from previous malware. This paper is a first attempt to provide an empirical deep study of Conficker victims, reveal how they are distributed differently from previous generation malware, and how this affects current reputation-based defense mechanisms. In addition, we

want to understand if there are some effective techniques for early detection of future variations of Conficker.

III. DATA COLLECTION

An interesting feature of Conficker is the resilient updating mechanism. To avoid detection, it automatically generates new domain names (of updating servers) [30], [38] and connects to those domain names to download an updated version of itself.¹ This function greatly supports Conficker to increase the survivability and resilience. However, once the domain generation algorithm was cracked by researchers, it also provides a way to sinkhole and track the victims. By registering new domain names that will be used by Conficker victims on controlled servers, defenders can collect visits from hosts infected by Conficker. This approach is widely known as C&C sinkholing and has been successfully adopted by researchers that study Conficker [17]. Since all Conficker infected hosts (regardless of versions) need to visit C&C domains frequently (an intrinsic feature of all botnets), the sinkholing approach is likely to collect most of Conficker infected victims regardless of their locations or individual behaviors, as long as the infected machines are alive and have network connections.

With the aid of *Shadowserver.org*, we have collected the Conficker sinkhole data² captured from January 1, 2010 to January 8, 2010. During this period, we observe 24,912,492 unique IP addresses of Conficker victims. Like many other measurement/analysis work, our empirical study has some limitations and they are mainly caused by not perfectly collected data. For example, our collection of Conficker victims may not cover all infected victims. In addition, we note that the accurate counting of malware victims is not an easy task because of the existence of DHCP (which may inflate actual numbers), NAT (which may deflate actual numbers), and many other issues [39], [32]. For example, Stone-Gross et al. [32] pointed out that there is a slight difference between the number of IP addresses and the number of real infected hosts. This is the limitation of almost all existing worm/botnet measurement studies. We do not intend to solve this problem in this paper. We simply report our observations from our collected data. Although the number may not be exact, with such a large scale it at least provides an estimation of overall characteristics and statistics of Conficker.

To obtain more interesting results, we survey previous work [20], [19], [27], [25], [39], [40], [31], [4] about the behavior of nefarious worms and bots/botnets³. They are used to compare with our Conficker result and to help us track whether infection trends have changed. Based on the information they provide, we select seven measurement studies, which are summarized in Table I. Of these, three are well-known network worms [20], [19], [27] and six are botnets [25], [39], [40], [31], [33], [26]. Fortunately, we could get the information of victims infected by Megad [26] or Srizbi bots [33] through [4]. Note that some

¹We provide a brief description on Conficker's domain generation mechanism in Appendix.

²Detailed information of the Conficker sinkhole is explained in Appendix.

³At this time, we have also tried to compare our results with the previous work of analyzing Conficker victims such as [5]. However we could not compare the results, because we were not able to get their data.

studies of botnets do not specify botnet names in their work [25], [39], [40], but they show the result of malicious nodes that send spam emails. Since most spam emails are delivered by botnets [25], we can reasonably assume that their studies represent the behavior of some bots or malware.

IV. WHO IS WORKING FOR CONFICKER?

In this section, we provide a basic but important network-level examination, which demonstrates fundamental characteristics of Conficker victims. We review how Conficker victims are distributed over the IP address space and ASes. Also, we investigate the bandwidth of Conficker victims and domain names that Conficker victims belong to. Finally, we survey portions of countries where Conficker victims heavily exist. Some of them are already provided by other studies [5], [17], but our work is more than just providing basic measurement results. To comprehend the radical alteration of malware, we compare Conficker victims' network-level characteristics with those of previous well-known bots or worms.

A. Distribution Over Networks

We plotted each victim's IP address to determine how Conficker victims are distributed over the IP address space and found that they are not uniformly distributed in the whole IP address space; instead the distribution is highly biased, mostly concentrated in some specific ranges.

Result 1. (Distribution over the IP address space) *Most of hosts infected by Conficker are concentrated in several specific IP address ranges.*

Figure 1 depicts the number of victims of three botnets - Conficker, MegaD, and Srizbi - over the IP address space. At first, we focus on the result of Conficker and we will discuss the results of others. The presence of several rising regions, which represent densely infected areas by Conficker, reveals that the victims are not uniformly distributed. Since the IP address ranges within these regions could be regarded as more vulnerable, we inspected three notable rising regions of Conficker in detail. They are in the range of (77.* - 96.*), (109.* - 125.*), and (186.* - 222.*) and they cover around 87% of all victims. In particular, the rapidly increasing region, which is in the range of (109.* - 125.*), includes 9,303,423 infected hosts and accounts for 37.34% of the total number of Conficker victims. To get a more detailed view, we narrowed down the scope from the ranges to more specific networks. In that region, we found that 123.* and 124.* networks are the main contributors. They comprise 1,701,438 infected hosts and account for 6.83% of all victims. We analyzed further and discovered that there are 40,278 Conficker victims in the 123.19.* network, which is around 61.9% of all possible IP addresses in that /16 subnet. Similar characteristics were observed in nearby networks such as the 123.22.* and the

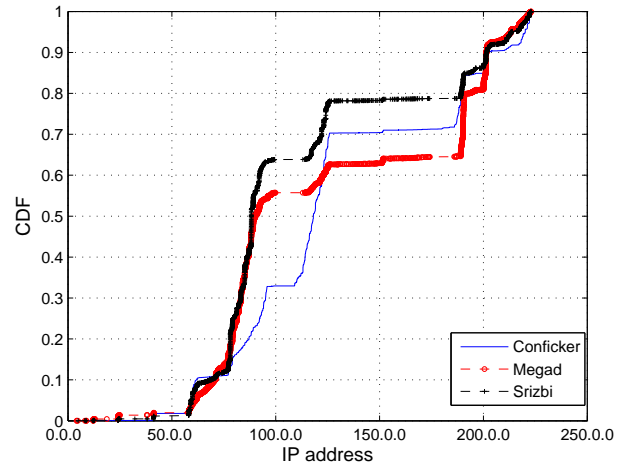


Fig. 1. Distribution of infected hosts over IP address; Conficker, MegaD, and Srizbi.

123.23.*⁴

Result 1.1. (Distribution over IP address space - Comparison) *Some portions of IP address ranges were already affected by the previous botnets, but some ranges such as 109.* - 125.* are more highly affected by Conficker.*

Figure 1 also shows the results of two recent botnets - MegaD and Srizbi. As displayed in Figure 1, their infected regions are very similar to those of Conficker. (77.* - 96.*), (109.* - 125.*), and (186.* - 222.*) are especially highly infected. However, their rising degree, which represents the infection rate, is different from that of Conficker. While Conficker shows high increase in the region of (109.* - 125.*), both MegaD and Srizbi show the rapid increase in the region of (77.* - 96.*).

We also compare this distribution of Conficker victims with that of previous botnets such as Waledac [31] and Botnet1 [25]. Interestingly, the comparison results are similar to the above results. While the ranges of (77.* - 96.*) and (186.* - 222.*) are also known as major locations of the Waledac botnet and Botnet1, they have no significant number of victims in the region of (109.* - 125.*).

We tried to understand why the range of (109.* - 125.*) has more victims in the case of Conficker. After investigating the data in this range, we concluded that the reason is most likely a change of infection trend, and we will elaborate on this in

Result 2.1.

Since it is nearly impossible to monitor the *entire* Internet, it is more efficient to focus on specific (suspicious) networks that are more likely to contain commands directed by a botmaster. The IP address ranges within wide spikes, which

⁴Since the 123.* network is in Class A network, it seems that there is no meaning in splitting it into subnetworks. However, people commonly split Class A networks into several /16 subnets to manage them efficiently. As in the case of 123.* network, we found that it is divided and assigned to several network providers. The 123.19.* network is one of them and it is assigned to VietNam Post and Telecom Cooperation and its inetnum is 123.19.0.0 - 123.19.255.255.

<i>Malware [Work]</i>	<i>Type</i>	<i>Data Source</i>	<i>Data Collection Time</i>
Botnet1 [25]	Botnet	Sinkhole server	Aug. 2004 ~ Jan. 2006
Botnet2 [39]	Botnet	Hotmail	Jun. 2006 ~ Sep. 2006
Botnet3 [40]	Botnet	Spamhaus	Nov. 2006 ~ Jun. 2007
Waledac [31]	Botnet	Infiltration into Waledac	Aug. 2008 ~ Sep. 2009
Srizbi [33]	Botnet	Sinkhole server	Aug. 2010
MegaD [26]	Botnet	Sinkhole server	Aug. 2010
CodeRed [20]	Worm	Measurement	Jul. 2001 ~ Oct.2001
Slammer [19]	Worm	Measurement	Jan. 2003
Witty [27]	Worm	Measurement	Mar. 2004

TABLE I
DATA SOURCE OF PREVIOUS WORMS/BOTS FOR COMPARISON.

are shown in Figure 1, can be good candidates that need to be focused.

Insight from Result 1 and 1.1 (Monitoring Networks more efficiently) *It is impossible to monitor all the IP addresses on the Internet, but we can monitor a limited number of specific ranges to efficiently detect commands and attacks in infected networks. Even though the ranges may be different for each botnet, there are still some common parts and they are good candidate ranges to monitor.*

Representing identities of Conficker-infected hosts by IP address is often preferable in a way that it is precise and elaborate. However, the number of the infected IP addresses is so large that this makes it hard to grasp the global view of Conficker victims. Hence, we use the *Autonomous System (AS)*, which is a useful method for clustering hosts on the Internet for easier management and has been applied in previous measurement work, to group the hosts infected by Conficker.

Result 2. (Distribution over ASes) *Of all infected hosts, the top two ASes account for 28.37% of all victims and top 20 ASes cover 52.54% of all victims. In particular, most of the top rated ASes are located in Asia.*

Conficker victims are concentrated in a few ASes and most of the top infected ASes are located in Asia. As shown in Table II, around 30% of infected hosts belong to one of only two ASes and more than 50% of infected hosts belong to one of the (top) 20 ASes. Most highly infected ASes are mainly distributed in Asia, particularly in China. This result also suggests that an approach to detect malicious hosts based on ASes would be practical.

Result 2.1. (Distribution over ASes - Comparison) *Even though the top two ASes were also sources of previous botnets, most of other top rated ASes are newly emerged in the Conficker case.*

By comparing the result of the distribution over ASes with that of other bots, we find that even if there are common ASes between recent bots (Conficker, MegaD, and Srizbi) and previous bots (Botnet1, Botnet2, and Botnet3), there is a

<i>ASN</i>	<i># Host</i>	<i>AS Name</i>	<i>Country</i>
4134	2825403	CHINA-BACKBONE	China
4837	1435411	CHINA169-BACKBONE	China
7738	385672	TELECOMUNICAOES	Brazil
3462	280957	HINET	Taiwan
45899	273577	VPNT-AS-VN	Vietnam
27699	260848	TELECOMUNICAOES	Brazil
9829	248444	BSNL-NIB	India
8167	237465	TELESC	Brazil
3269	231020	ASN-IBSNAZ	Italia
9121	207849	TTNET	Turkey
9394	195088	TELEFONICA	China
4812	182015	CRNET	China
4788	180876	CHINANET-SH-AP	Malaysia
8402	141130	TMNET-AS-AP	Russia
8151	138567	CORBINA-AS	Mexico
17974	137991	UNINET	Indonesia
4808	137672	TELKOMNET-AS2-AP	China
3352	135276	CHINA169-BJ	China
8708	128228	TELEFONICA-DATA-ESPANA	Romania
3320	126520	RDSNET	Germany

TABLE II
CONFICKER VICTIMS IN THE TOP 20 ASes.

significant difference in the locations of infected ASes. Some previous studies [25], [39], [40] investigated which ASes are the major sources of the botnets that deliver spam emails⁵. We compare their findings with our result and denote it in Table III. In [25], the authors analyzed data collected in 2004 - 2006 and pointed out that most of the bots are located in North America (particularly in USA), while in [39] and [40] in which data was collected in 2006 - 2007, it was emphasized that bots spread widely over the world. However, in the case of Conficker and two other recent bots (MegaD and Srizbi), ASes in the USA are not major resources of bots any more. Instead, most highly infected ASes are located in Asia and South America.

From this result, we conclude that the trend of major locations of bot infected hosts is still changing; (i) *mainly located in North America*, (ii) *widely spread over the World*, (iii) *popular in Asia and South America*. This trend guides us to observe Asia and South America more closely than North America, which used to be the major source of spam email

⁵In [40], they only present the top five of ASes, and that is why we could not compare the whole list.

Conficker		Botnet1 [25]		Botnet2 [39]		Botnet3 [40]		MegaD [26]		Srizbi [33]	
ASN	Country	ASN	Country	ASN	Country	ASN	Country	ASN	Country	ASN	Country
4134	China	766	Korea	4134	China	4766	Korea	3352	Spain	9121	Turkey
4837	China	4134	China	4837	China	19262	USA	3269	Italy	17552	Thailand
7738	Brazil	1239	USA	4776	Australia	3215	France	6739	Spain	4134	China
3462	Taiwan	4837	China	27699	Brazil	4837	China	9121	Turkey	5617	Poland
45899	Vietnam	9318	Japan	3352	Spain	4134	China	6147	Peru	9829	India
27699	Brazil	32311	USA	5617	Poland	no info.	no info.	19262	USA	4837	China
9829	India	5617	Poland	19262	USA	no info.	no info.	4134	China	7738	Brazil
8167	Brazil	6478	USA	3462	Taiwan	no info.	no info.	7738	Brazil	4766	Australia
3269	Italy	19262	USA	3269	Italy	no info.	no info.	7418	Chile	24560	India
9121	Turkey	8075	USA	9121	Turkey	no info.	no info.	22927	Argentina	27699	Brazil

TABLE III
TOP 10 ASes HOSTING CONFICKER AND SPAMMING BOTNETS.

when we built blacklists to prevent spam at the time. It is important that the trend of major sources of bots is changing. Also, we find that four ASes in Conficker are never seen in the previous results. Two of them are in Asia (Vietnam and India) and another two are in South America (Brazil).

Insight from Result 2 and 2.1. (Change of Infection Trend) *North America used to be the main contributors of botnets, but now Asia and South America contribute more. This means that the locations of the main sources of botnets are changing and we may chase this trend (e.g., new malware spreading models and defense strategies are probably needed).*

B. Distribution Over Domain Names

In this section, we inspect the domain names of each victim using DNS reverse lookup.⁶ A domain name indicates a group in which a host belongs and it can be a good way to reveal the host itself because domain names are expressed in easy and comprehensible words.

Result 3. (Distribution over Domain Name) *The .br, .net and .cn domains cover around 24.42% of Conficker victims. Interestingly, one of the third level domains covers around 7% of infected hosts, which means it contains more than 1,700,000 victims.*

As shown in Table IV, only a few domains account for about 20% of hosts infected by Conficker. This does not solely apply to top level domains but to all second level domains and third level domains as well. In the case of top and second level domain names, their scope is quite broad and it is hard to find any big advantage when compared to IP address range or AS number. However, for third level domain names, it is possible to focus on small sets of victims. It is useful to monitor victims because the top third level domain includes numerous Conficker victims. In particular, we find that domain *163data.com.cn* accounts for 6.88% of infected

⁶In our DNS reverse lookups, about 49% of victims did not return valid results and therefore we labeled them as “Unknown”, shown in Table IV. Since previous studies also showed similar rates of “unknown” domains, we leave them in the table.

hosts. Also, more than 99% of victims in *163data.com.cn* include the word *dynamic* in their fourth level domain names. From this, we can guess that they are using dynamic IP addresses, as their names imply. This result is similar to [39] which uncovers dynamic IP addresses as a main source of most spam emails.

Result 3.1. (Distribution over Domain Name - Comparison) *The .net domain is still prevalent, but new domains such as .br, .cn, and .ru have recently emerged as heavy resources of botnets. The .com and .edu domains used to be the major sources of worms, but now they seem to cast off the yoke of malicious domains.*

Comparing the domain result with previous well-known worms, we found that a few domains that were not previously seen in Conficker. Also, we found that *.com* and *.edu* domains, which used to be nefarious domains, are now relatively clean. Unfortunately, because they do not provide the result of second level and third level domain distributions, we could only compare top level domains. In the case of the previous worms, top contributors of infected domains are *.net*, *.com* and *.edu*. However, in the case of Conficker, things have changed. While the *.net* domain is still prevalent, there are newly emerged domains which are not shown in the previous work: *.cn*, *.ru*, *.in*, and *.mx*. All domains that are newly seen represent their countries and we call these ccTLDs (Country Code Top Level Domains). The report from Verisign [37] shows that the registration rate of above ccTLDs has increased a lot for the past three years (e.g., 10% of increasing in 2009). From the Verisign report, we find that some vulnerable domains such as *.cn*, *.ru*, *.br*, and *.ar* are registered much more than many other domains. This implies that the number of hosts in recently registered ccTLDs have increased a lot. Therefore we may monitor more closely whether they are infected by malware or not, since they may not be on any blacklists. The more interesting part is *.edu* and *.com* domains are no longer serious sources of malware. Of course, there are infected hosts which still belong to those domains, but its coverage is reduced to 1.21% in *.com* and 0.0096% in *.edu*. This result implies that the networks in *.com* and *.edu* domains are probably better managed and protected than before.

<i>Top Level</i>	<i>Percentage</i>	<i>Second Level</i>	<i>Percentage</i>	<i>Third Level</i>	<i>Percentage</i>
Unknown	48.81%	Unknown	48.81%	Unknown	48.81%
br	8.83%	com.cn	6.89%	163data.com.cn	6.88%
net	8.65%	net.br	4.61%	veloxzone.com.br	1.96%
cn	6.94%	com.br	4.20%	dynamic.hinet.net	1.86%
ru	5.01%	hinet.net	1.91%	telesp.net.br	1.69%
it	2.36%	telecomitalia.it	1.55%	retail.telecomitalia.it	1.46%
ar	1.54%	corbina.ru	0.99%	brasilecom.net.br	1.39%
in	1.35%	ny.adsl	0.93%	broadband.corbina.ru	0.99%
com	1.21%	com.mx	0.90%	kd.ny.adsl	0.93%
mx	1.16%	com.ar	0.84%	prod-infinitum.com.mx	0.85%

TABLE IV
TOP 10 DOMAIN NAMES HOSTING CONFICKER VICTIMS IN EACH LEVEL.

We also investigated the domain result of other recent botnets, and we discovered that *.net* and ccTLDs are prevalent, but *.com* is less serious than the previous worms in their results. This result confirms the above comparison result between Conficker and previous worms. The comparison result is summarized in Table V.

Result 3.2. (Distribution over Domain Name - Sensitive Domain Name) *There are Conficker victims in government networks and companies listed in Fortune 100, even though the number of infected hosts is small.*

Besides sending DDoS packets and spam emails, a botnet can steal sensitive information from victims [15]. If hosts infected by a bot belong to critical networks such as government and military networks that contain sensitive information, a botmaster can steal important information from them. Using our Conficker data, we investigated how many victims are affiliated with government or military networks and we found 714 such victims. Surprisingly, victims in government networks are not limited to a few countries, instead they are spread around 70 countries including U.S.A., Parkistan, India and China. Also, we investigated how many victims are in well-known companies. To do this, we used the *Fortune 100 Company List* [11] and we found 2,847 such hosts. Conficker victims still exist within several reputable companies such as HP and IBM.

Insight from Result 3, 3.1 and 3.2. (Watch out for new and sensitive Domains!) *It is nearly impossible to monitor all domain names. However, we have observed that recently registered ccTLDs are more vulnerable and more easily infected by Conficker. Hence, it is necessary to closely monitor those recently registered domains. In addition, even though the number of victims is not large, a botmaster of Conficker can steal sensitive information from government and top rated company networks.*

C. Distribution over Bandwidth

Besides IP address, AS and domain names, bandwidth gives us information that shows us what kinds of networks Conficker victims belong to. It also helps to predict the power of the botnet. For instance, if we know there are one million

Conficker victims in the world and most Conficker victims are in networks with bandwidth less than 1 Kbps, we can deduce that it could generate 1 Gbps traffic in the best case. To measure the bandwidth, we use *Tmetric* [35] which sends ICMP packets to the target network and provides a measured bandwidth result. *Tmetric* sends several packets, whose sizes are different to each other, to a target network and measure their response times. Basically, a response time depends on the bandwidth of the target network, thus we can estimate the bandwidth of a network based on the measured response time. Here, to estimate the bandwidth of a subnet, we have sent five packets, whose sizes are different from each other, to the subnet and estimated the average bandwidth value from their results. Since *Tmetric* needs to contact the target network to estimate the bandwidth, we cannot get the bandwidth result without live target networks and hosts. It takes quite a long time to contact each host and measure the bandwidth, so we only contact one host in each distinct subnetwork (/24) where Conficker victims exist. We reasonably assume that hosts in the same subnetwork (/24) have the same bandwidth. In addition, we have contacted each network 5 times and measured the average value of estimated bandwidth results.

Result 4. (Bandwidth Distribution) *About 99% of Conficker victims have bandwidth less than 1 Mbps and this means that most of them are ADSL or Modem/Dialup users.*

We find that most victims are using Modem/Dialup or ADSL networks. As shown in Figure 2 (a), about 90% of Conficker victims are in the network whose bandwidth is less than 200 Kbps and around 99% of victims are residing in the network whose bandwidth is less than 1 Mbps. This result is similar to [14] and [39] which denote most bots are using ADSL or Dialup networks. When we conducted this measurement, we found interesting patterns between the bandwidth of a subnet and the number of infected hosts in the subnet.

Result 4.1. (Bandwidth Distribution - relation with the numbers of victims) *The networks that have low bandwidth are likely to have more Conficker victims than those with high bandwidth.*

We suspect that there is a relationship between the

Conficker		CodeRed		Slammer		Witty		MegaD		Srizbi	
Top level	%	Top level	%	Top level	%	Top level	%	Top level	%	Top level	%
Unknown	48.81%	Unknown	47.22%	Unknown	59.49%	net	33%	Unknown	25.87%	Unknown	37.93%
br	8.83%	net	18.79%	net	14.37%	com	20%	net	19.44%	net	8.65%
net	8.65%	com	14.41%	com	10.75%	Unknown	15%	com	10.28%	tr	8.19%
cn	6.94%	edu	2.37%	edu	2.79%	fr	3%	br	7.51%	ru	6.12%
ru	5.01%	tw	1.99%	tw	1.29%	ca	2%	it	4.30%	br	5.45%
it	2.36%	jp	1.33%	au	0.71%	jp	2%	ar	3.42%	pl	3.98%
ar	1.54%	ca	1.11%	ca	0.71%	au	2%	co	2.85%	com	3.20%
in	1.35%	it	0.86%	jp	0.65%	edu	1%	cl	2.48%	th	3.19%
com	1.21%	fr	0.75%	br	0.57%	nl	1%	pl	2.20%	in	2.38%
mx	1.16%	nl	0.73%	uk	0.57%	ar	1%	es	1.98%	it	1.59%

TABLE V
TOP 10 DOMAIN NAMES HOSTING CONFICKER, CODERED, SLAMMER, WITTY, MEGAD, AND SRIZBI.

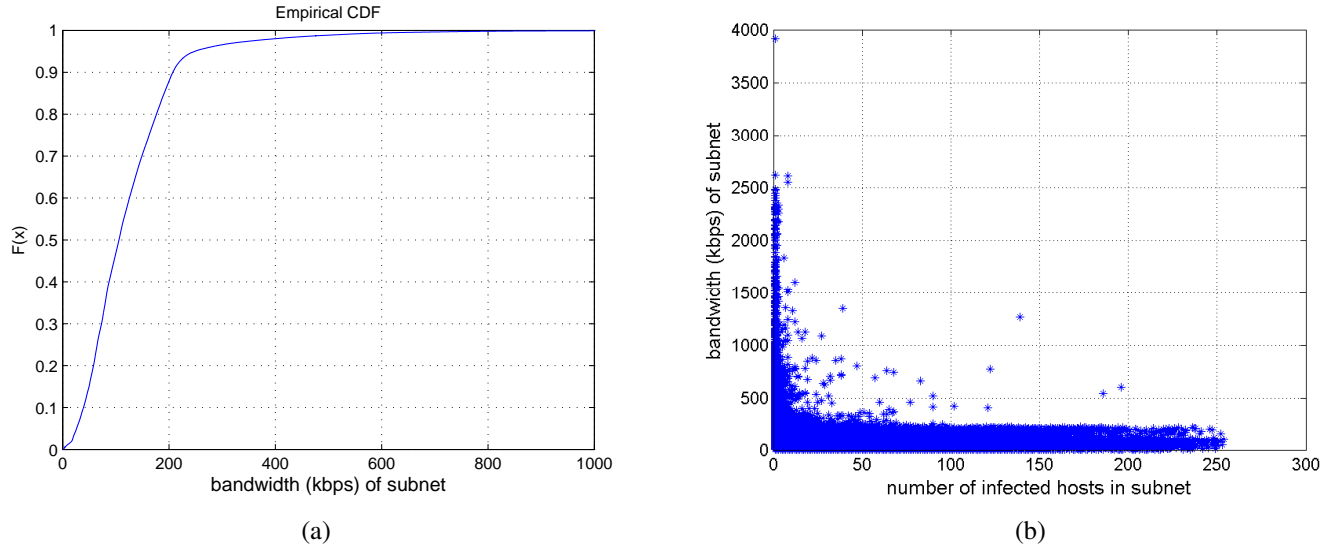


Fig. 2. Bandwidth measurement of Conficker victims.

bandwidth of a network and the number of infected hosts of the network. As shown in Figure 2 (b), the bandwidth of the subnet is inversely related to the number of infected hosts in the subnet. We think that this pattern is related to the manageability of each network. A network with high bandwidth indicates consuming high setup cost and it also means the network is that worthy. And we could infer that such worthy network is under reasonably good maintenance.

Insight from Result 4 and 4.1. (Examine ADSL or Modem/Dialup networks) *Hosts with ADSL or Modem/Dialup connections are still very vulnerable.*

D. Distribution over Geographic Location

Result 5. (Geographic Location) *34.47% of infected hosts are located in China, which is larger than the total number of Conficker victims from the next top eight countries.*

As shown in Table VI on the distribution over countries, the top ten countries include over 70% of Conficker victims, China ranks number one by a large margin. Conficker victims are distributed over most of the world including Asia, Europe, and South America, but interestingly, only 1.1% of victims

are located in North America. This result is somewhat different from previous infection patterns.

Result 5.1. (Geographic Location - Comparison) *In previous worms and botnets, most the infected hosts were located in North America - especially in USA, but in recent botnets such as Conficker, most victims are located in the Asian region - especially in China.*

We compare the country distribution with that of other worms and bots to determine whether it is different or similar and we find that the location of heavy malware contributors is changing. Even though we could not get the exact country distribution from the previous work [25] [39], we are able to estimate which country had more victims based on their distribution over ASes. From Table VI and III, we observe that worms prevalent several years ago were mainly located in North America. In previous botnets, [39] and [40] show that victims are mainly located in both Asia and North America, but [25] and [31] denote that most victims are located in North America. However, contrast to the results of previous work, we find that Conficker victims are mainly located in Asia and not in North America, where only 1.1%

Conficker		Waledac		CodeRed		Slammer		Witty		MegaD		Srizbi	
Code	%	Code	%	Code	%	Code	%	Code	%	Code	%	Code	%
CN	34.47%	US	17.34%	US	43.91%	US	42.87%	US	26.28%	US	11.45%	TR	18.66%
BR	9.43%	U.K	7.76%	KR	10.57%	KR	11.82%	U.K	7.27%	ES	11.04%	RU	9.41%
RU	7.39%	FR	7.04%	CN	5.05 %	Unknown	6.96%	CA	3.46 %	BR	8.33%	KR	6.69%
IN	4.45%	ES	5.90%	TW	4.21%	CN	6.29%	CN	3.36%	IT	5.43%	IN	5.85%
IT	3.56%	IN	5.50%	CA	3.47%	TW	3.98%	FR	2.94%	U.K	5.24%	BR	5.76%
VT	2.81%	no info.	no info.	U.K.	3.32%	CA	2.88%	JP	2.17%	CO	4.88%	CN	5.66%
TW	2.59%	no info.	no info.	GE	3.28%	AU	2.38%	AU	1.83%	CN	4.32%	US	4.26%
GE	2.03%	no info.	no info.	AU	2.39%	U.K.	2.02%	GE	1.82%	KR	3.88%	PL	4.19%
AR	2.00%	no info.	no info.	JP	2.31%	JP	1.72%	NL	1.36%	AR	3.74%	TH	3.47%
ID	1.85%	no info.	no info.	NL	2.16%	NL	1.53%	KR	1.21%	CL	3.65%	GE	2.53%

TABLE VI

TOP 10 COUNTRIES WHERE CONFICKER, WALEDAC, CODERED, SLAMMER, MEGAD, AND SRIZBI ARE LOCATED. (CODE DENOTES *two-letter country code*, AND THE MEANING OF EACH COUNTRY CODE IS; CN = CHINA, BR = BRAZIL, RU = RUSSIA, IN = INDIA, IT = ITALY, VT = VIETNAM, TW = TAIWAN, GE = GERMANY, AR = ARGENTINA, ID = INDONESIA, U.K = UNITED KINGDOM, FR = FRANCE, ES = SPAIN, KR = KOREA, CA = CANADA, AU = AUSTRALIA, JP = JAPAN, NL = NETHERLANDS, TR = TURKEY, CO = COLUMBIA, CL = CHILE, PL = POLAND, TH =THAILAND

of victims are located. Also, this pattern is shown in other recent Srizbi botnet. Most victims of the Srizbi botnet are located in Asian regions. In the case of the MegaD botnet, although USA is on the top rank, the percentage of USA is less than previous worms and the percentage of Asian and South America regions is higher than previous worms. Thus, we may think that MegaD also shows similar characteristics. Therefore, changing monitoring focus from North America to Asia seems reasonable.

Insight from Result 5 and 5.1. (From North America to Asia - Confirmed) *We clearly observe that the hosts infected by Conficker are mainly located in Asia and not in North America, as also shown in Result 2 and 2.1.*

V. HOW WELL DO REPUTATION-BASED DETECTION SYSTEMS DETECT CONFICKER?

In this section, we examine how well current reputation-based detection systems detect Conficker. Reputation-based approaches (e.g., DNS blacklists) are widely considered as promising to detect malicious hosts and networks [25], [3]. To investigate their effectiveness, we will test DNS blacklists [3] and other reputation-based detection systems such as Dshield [9] and FIRE [10] to check if they could successfully detect emerging threats such as Conficker.

A. DNS Blacklists

We have investigated several well-known blacklists such as DNSBL [7], SORBS [29], SpamHaus [2], and SpamCop [1] to see how many victims of Conficker are on their blacklists. We tested all 24,912,492 infected hosts and we found out that only 4,281,069 hosts are on blacklists which is only 17.18% of all victims.

Result 6. (DNS Blacklist) *DNS blacklists only cover a small portion of Conficker victims. More specifically, only 17.18% of Conficker victims are found on any of four DNS blacklists.*

Our investigation result is quite different from the previous work [25] which shows about 80% of bot infected hosts are already on some blacklists and we believe that the disparity is caused by the difference of distribution of infected hosts. As we mentioned in Section IV-A and IV-B, the distribution of Conficker victims (over IP address space, ASes, Domain names and Countries) is different from the previous work, and this makes it hard to build effective blacklists for detecting emerging malicious hosts/networks, because blacklists highly depend on the reputation of hosts and networks obtained from their previous records (and currently heavily rely on spam activity records).

Insight from Result 6. (Unfortunately, blacklists can not help us all the time) *Only less than 20% of victims are on DNS blacklists, which means that we need better ways to detect future emerging malware.*

B. Dshield and FIRE

Since most DNS blacklists are mainly to detect hosts or ASes sending spam, they may not detect other malicious behaviors (potentially) performed by (emerging) infected hosts. Some other reputation-based detection systems are also proposed to complement DNS blacklists, and we need to investigate their performance of detection. There are several studies that try to detect network scanning attacks or web-based attacks and Dshield [9] and FIRE [10] are good examples of them. Dshield provides information to detect hosts or ASes sending suspicious network scanning/attacking packets, and FIRE [10] lists malicious ASes which frequently host rogue networks by measuring their reputation. We plan to inspect how many Conficker victims are notified by Dshield and FIRE⁷.

⁷It is worth noting that the main purpose of FIRE is to find networks/hosts launching malicious services instead of directly detecting bot-infected hosts. We still perform the experiment because it is well-known that botnets are heavily used platforms to launch malicious services [15].

Result 7. (Dshield) Only 0.33% of victims of Conficker are found on the list of malicious IP addresses reported by DShield, and most of the top ASes infected by Conficker are not on the malicious AS list of Dshield.

Checking Conficker victims against the list provided by Dshield [8], we found that only a small portion of hosts and ASes are on the list. We investigated 588,797 IP addresses presented by Dshield, and they denoted world-wide attackers/scanners that were detected by all kinds of IDSs and reported to DShield. Since one of the infection vectors in Conficker is random IP scanning [22], we expect a large portion of Conficker victims to show up in Dshield. However, we only find 82,856 hosts from the list. This shows that these Conficker victim hosts are probably easy targets of many previous malware. However, DShield is still not good at catching major portions of new emerging malware such as Conficker. Similarly, we examined the malicious AS list provided by Dshield and we only observed 83 Conficker infected ASes out of 10,584 ASes given by Dshield. Only one of them (AS4812) is a serious contributor of Conficker (ranked 12th among infected ASes) but the rest are not as critical as AS4812. Most of them cover less than 0.02% of Conficker victims.

Result 8. (FIRE) Most highly infected ASes by Conficker are not reported by FIRE.

We compared our infection list of ASes with the results provided by FIRE as well and we want to know whether FIRE is helpful in detecting Conficker victims. Although FIRE denotes AS4134 as the 8th most malicious AS in its list, most of other heavily infected ASes by Conficker are not shown in the top 500 malicious ASes of FIRE. Some of the main contributing ASes to Conficker have never shown up on FIRE’s list.

Insight from Result 7 and 8. (New and complementary detection approaches are needed) DNS blacklists, Dshield and FIRE detect only a small portion of Conficker victims. This means that these reputation-based approaches are not the perfect solution. We need to improve them significantly and complement them with other approaches.

When we tested Dshield and FIRE, we expected that they could complement DNS blacklists, but the result is not very positive. This implies that these reputation-based systems alone are far from enough to protect the Internet from emerging threats. We believe that new detection systems (e.g., those based on anomalous behaviors of malware) are badly needed to complement existing reputation-based approaches to defend against future malware threats.

C. Blacklisted and Non-blacklisted victims

As we observe in this section, only a small number of victims are detected by the current reputation-based detection systems. From this observation, we raise following questions;

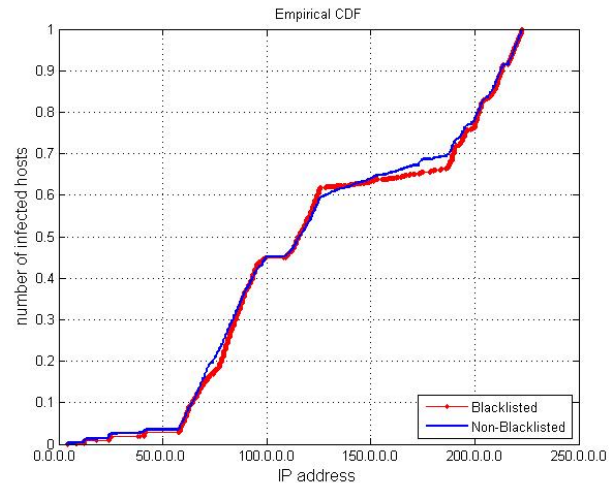


Fig. 3. Distribution of blacklisted/non-blacklisted victims over IP address.

ASN (BL)	%	ASN (NBL)	%
AS45899	8.35%	AS4134	21.87%
AS9829	7.87%	AS4837	11.12%
AS4134	4.25%	AS7738	2.25%
AS27699	4.07%	AS3462	2.13%
AS7738	3.97%	AS3269	1.80%
AS8167	3.35%	AS9394	1.50%
AS4837	2.58%	AS4812	1.42%
AS17974	2.23%	AS4788	1.41%
AS17557	2.17%	AS9121	1.28%
AS24560	1.97%	AS8167	1.20%

TABLE VII
TOP 10 ASes WHICH HAVE BLACKLISTED AND NON-BLACKLISTED CONFICKER VICTIMS (BL IS FOR BLACKLISTED AND NBL FOR NON-BLACKLISTED).

(i) who (i.e. Conficker victims) are detected by the current reputation-based detection systems and who are not? and (ii) how are they different? We believe that answers to the questions can give us some hints to design more effective reputation-based detection systems.

To understand who are listed and who are not, we show the distribution of blacklisted and non-blacklisted victims over IP address spaces and it is shown in Figure 3. As presented in Figure 3, the distributions of blacklisted and non-blacklisted victims are very similar to each other and both are mainly distributed in (77.* - 96.*), (109.* - 125.*), and (186.* - 222.*). This result implies that even though each individual victim of blacklisted is different from that of non-blacklisted, they are both located in similar IP address spaces.

We investigate AS and Country distributions of blacklisted and non-blacklisted victims to obtain more knowledge of their distributions.

The top 10 ASes which include blacklisted and non-blacklisted victims are shown in Table VII. Interestingly, some top 10 ASes containing blacklisted victims also have many non-blacklisted victims. Moreover, these common ASes cover more than 35% of non-blacklisted victims.

In addition, we present how both blacklisted and non-

Country (BL)	%	Country (NBL)	%
BR	15.85%	CN	40.23%
IN	15.43%	BR	8.00%
VN	11.18%	RU	7.17%
RU	8.42%	IT	4.01%
CN	8.22%	TW	2.99%
ID	2.63%	DE	2.36%
UA	2.47%	IN	2.11%
AR	2.43%	AR	1.89%
PK	2.41%	KR	1.82%
SA	2.34%	ID	1.71%

TABLE VIII
TOP 10 COUNTRIES WHICH HAVE BLACKLISTED AND NON-BLACKLISTED
CONFICKER VICTIMS (BL IS FOR BLACKLISTED AND NBL FOR
NON-BLACKLISTED).

blacklisted victims spread over the world and then we show the top 10 countries with both in Table VIII. As represented in Table VIII, we observe that blacklisted victims and non-blacklisted victims spread over similar countries.

Result 9. (Blacklisted and Non-blacklisted victims) *The distributions of victims of blacklisted and non-blacklisted are very similar to each other.*

Then, what do these observations imply? We believe that they indicate that we might need another viewpoint for reputation-based detection systems toward victims. Current reputation-based detection systems focus on individual victim hosts (i.e. fine-grained view). However, we consider that the clustering (or grouping) of victims (coarse-grained view) can help us to complement the reputation-based detection systems based on each individual host, since both blacklisted and non-blacklisted victims are located close to each other. There are also some recent studies which propose spam detection systems based on coarse-grained view of victims [23], [24] and we believe we confirm that their approaches are helpful to complement fine-grained reputation-based detection systems.

Insight from Result 9. (Coarse-grained view of victims) *Coarse-grained view could complement fine-grained reputation-based detection systems based on each individual victim.*

VI. HOW POWERFUL IS CONFICKER BOTNET

A botmaster will use infected victims to perform malicious operations. Although it has not been previously reported whether Conficker really fulfills attacks or not, it is possible that a botmaster of Conficker commands victims to carry out some malicious actions. Thus, it may be interesting to understand how Conficker victims affect networks or hosts when they perform attacks.

The power of botnets can be defined by each attack method they provide. For example, if bots generate malicious traffic such as DDoS packets, the performance of this attack will be decided by the amount of traffic they can generate. And in this case, the performance highly depends on the number of

infected hosts since the more victims there are, the more traffic they can produce.

We define three different power metrics for Conficker. First, we define *power of massive attack* to represent how much network traffic Conficker can create. We believe this metric enables us to understand the effects of Conficker on the network when they perform DDoS-like attacks or send spam emails massively. Second, we define *power of information stealing attack* to show how much important information can be stolen by Conficker. Finally, we define *power of launching malicious sites* to reveal how many malicious sites a botmaster could possibly run for her profit.

A. Power of Massive Attack

To comprehend massive attacks produced by bots, we should measure network traffic produced by infected hosts. We assume that a host (or hosts) is connected to a network and the network is managed by subnets such as /24 and /16. In addition, we will treat each subnet as a source of network traffic. We define the following parameters:

- N_i , the number of infected hosts (Conficker victims) in the i_{th} /24
- B_i , the bandwidth of end hosts (Conficker victims) in the i_{th} /24 subnet
- I_i , 1 if there is any infected host(s) in i_{th} /24 subnet, 0 otherwise

We measure possible network traffic from each infected subnet considering two extreme cases. The first case is that a single host in the subnet can consume all the allowed bandwidth of the backbone router connected to the subnet. That is, the generated traffic from this infected subnet i is just B_i . The second case is that the bandwidth of the backbone router connected to the subnet is significantly larger than the bandwidth available at end hosts. In this case, suppose there are N_i infected hosts in the subnet i and each generates B_i Mbps traffic, then the infected subnet generates $N_i B_i$ Mbps maximum traffic. We can easily infer that the actual traffic could be generated by infected hosts inside the subnet ranges between these two bounds. More formally, the *power of massive attack* is defined as follows (assume that Conficker victims are across n /24 subnets):

$$\sum_{i=1}^n I_i B_i \leq \text{power of massive attack} \leq \sum_{i=1}^n N_i B_i$$

When we measured this metric, we found 1,339,698 /24 subnets in our data set. We calculated the number of infected hosts and the bandwidth of each /24 subnet. To represent those values more clearly, we denote the $N_i B_i$ and $I_i B_i$ values of infected /24 subnets in Figure 4.

As presented in Figure 4, in the case of $N_i B_i$, 90% of /24 subnets have bandwidth with lower than about 4.5 Mbps and $I_i B_i$ values show that most of them are lower than 0.5 Mbps. To measure the maximum bound of the *power of massive attack*, we sum up all total bandwidth of Conficker and find that the total value is 2.067 Tbps. In addition, we measure the minimum bound and it is 160.37 Gbps. Thus, the value of *power of massive attack* will be determined between 160.37

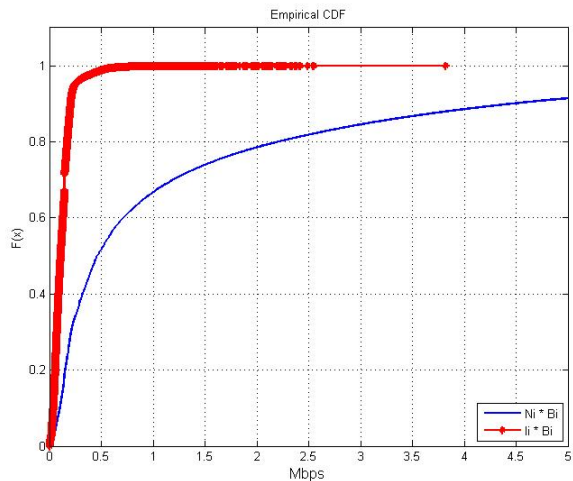


Fig. 4. Distribution of $N_i B_i$ and $I_i B_i$ values of each /24 subnet.

Gbps and 2.067 Tbps. We believe that the value is enough (even the minimum bound value) to consume most resources of the target server, if the Conficker victims are used to perform a DDoS attack heading to a certain target server (e.g., one of popular web servers).

Result 10. (Massive Attack) *If a botmaster of Conficker performs DDoS attack, she may produce enough network traffic to deny the services of popular web servers.*

In the previous metric, we assume that all bots could generate malicious traffic without any problem. However, it may not be true, since some network security solutions such as reputation-based detection systems may detect malicious traffic and filter it. It motivates us to modify the previous power metric to consider detection and filtering.

To do this, we add another parameter of *detection rate* to the previous metric. The *detection rate* is determined for each /24 subnet and calculated by dividing the number of hosts, which are infected and enlisted in blacklists, in a /24 subnet by the number of all infected hosts in the /24 subnet. Since this rate means how current reputation-based detection system can filter malicious traffic, we will use $1 - \text{detection rate}$ to measure the percent of traffic not filtered. This idea can be formalized as follows.

- P_i , the number of infected hosts which are in blacklists⁸ of the i_{th} /24 subnet
- Q_i , the number of all infected hosts of the i_{th} /24 subnet
- α_i , *detection rate* of i_{th} /24 subnet = $\frac{P_i}{Q_i}$

Finally, we can get the modified metric by multiplying the previous metric of *power of massive attack* with $(1 - \alpha_i)$, thus we will have a new *minimum bound of traffic* and *maximum bound of traffic*.

We present the distribution of $(1 - \alpha_i)N_i B_i$ and $(1 - \alpha_i)I_i B_i$ values in Figure 5. Since some traffic is detected and filtered,

⁸When we measure the *detection rate*, we use four different DNS blacklists mentioned in Section V.

the distribution shows that the amount of overall traffic is smaller than the previous metric. Finally, *maximum bound of traffic* is 1.7835 Tbps and *minimum bound of traffic* is 149.3 Gbps. These ranges are lower than the previous values (around 15% lower). However the ranges still point out that even the minimum bound of traffic is enough to consume most of the network resources of certain target networks. This implies that even if there were perfect traffic filtering system based on reputation, a botmaster still can launch a severe attack against the networks or hosts.

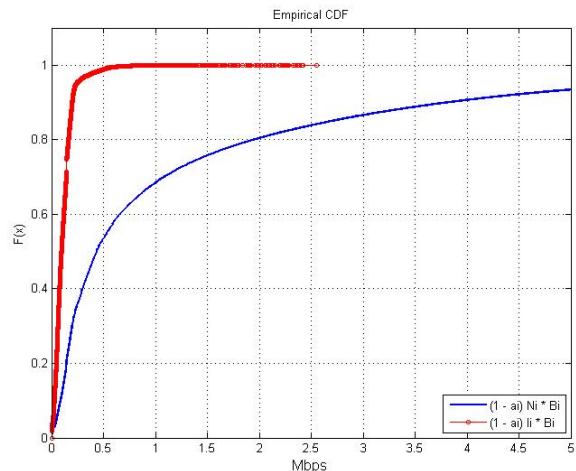


Fig. 5. Distribution of $(1 - \alpha_i)N_i B_i$ and $(1 - \alpha_i)I_i B_i$ values of each /24 subnet.

Result 11. (Massive Attack - with reputation-based filtering) *Although some traffic can be filtered by reputation-based detection, the Conficker botmaster still can generate at least 150 Gbps (up to 1.68 Tbps) network traffic.*

B. Power of Information Stealing Attack

Besides producing malicious network traffic, bots also can steal sensitive information from infected hosts. At this point, the performance of a botnet is measured in a different way. Previously, we consider both the number of infected hosts and their bandwidth. However in this case, we do not need to consider the bandwidth any more, since a botmaster can steal sensitive information from infected hosts through a few network messages and this operation does not require heavy traffic. Thus, we can focus only on the number of infected hosts.

However, considering only the number of infected hosts is not enough. Every host has its own sensitive information, but is the importance of the information same on each host? There may be hosts with more or less important information. Then, how do we define the importance of the information? We think that if victims belong to some companies or organizations, which are more likely to have critical information, we consider them as more important hosts. Some cautious readers will notice that this intuition is very similar to the *Sensitive Domain Name* of **Result**

3.2. Thus, we use the same notion in this measurement. We regard that the victims have more sensitive information, if they belong to companies in *Fortune 100 lists* [11] or have domains of *gov* or *mil*. Thus, the case of stealing information from each type of domain can be summarized like following.

- A_1 , the number of hosts which belong to companies in *Fortune 100 list*
- A_2 , the number of hosts which have *gov* or *mil* domain
- A_3 , the number of hosts which belong to neither A_1 nor A_2

When a botmaster issues a command to random bots in order to steal information, the probability p_i , which she steals information from each type, is simply described as the following equation.

$$p_i = \frac{A_i}{\sum_{j=1}^3 A_j}, \text{ where } i \in \{1, 2, 3\}$$

In the case of Conficker, we found 2,847 victims of A_1 , 714 victims of A_2 and 24,908,931 victims related to A_3 . Thus, the value of p_1 is 0.00014, p_2 is 0.00003, and p_3 is 0.999857. The probability of randomly stealing information (from the sensitive domains) seems to be really low. However if a botmaster has knowledge of the sensitive domains, it is easy for her to steal important information by sending a command to targeted domains.

Result 12. (Information Stealing Attack) *Conficker has a good foothold to steal important information from sensitive companies or organizations.*

C. Power of Launching Malicious Services

The previous metrics mainly consider each bot as a client. However sometimes a botmaster lets a bot be a server to launch a malicious service such as a phishing site or a malware downloading site [15]. Thus, we also need to estimate the power of hosting malicious sites.

The parameters to denote this metric are similar to the previous ones. However we should consider another important factor, the *reputation* of a bot-infected host. When a bot launches a malicious site, it may not always succeed, because some reputation-based detection systems or other protection schemes can block users from visiting the malicious site. At this time, those systems mainly use reputation of a host to determine whether the host is malicious or not. It implies that the *reputation* is an important factor to estimate how a botmaster can successfully runs malicious sites.

To measure the *reputation* of a host, we borrow knowledge from the reputation-based detection systems. Similar to the way of measuring *detection rate*, we investigate how many bots are already listed in the blacklists. When we determine the *reputation* of a Conficker victim, we do not only investigate the host level, but also consider neighbors' information, i.e., even though a host is not in any blacklists, if it is surrounded by several malicious hosts, it can be also considered as suspicious.

Some recent reputation-based detection systems use neighbor's information to detect hosts launching malicious services and they show promising detection results [10]. Thus, we believe that our consideration of neighbor's information is reasonable in this case.

In addition, this idea is similar to coarse-grained view (i.e. aggregation or grouping) of Conficker victims that we mentioned in section V. Thus, we aggregate victims within the same /24 network to measure the reputation⁹. To represent the *reputation* of each /24 subnet, we define a term of β_i for each i_{th} /24 subnet and it can be represented as follows:

$$\beta_i = \begin{cases} 1 & \text{if there is no victim enlisted in the blacklists} \\ 0 & \text{if there is any victim enlisted in the blacklists} \end{cases}$$

Besides the *reputation*, we should also consider the probability of hosts being behind NAT (network address translation). Multiple hosts behind NAT may not provide network services because they do not have unique IP addresses which are accessible from outside. To consider this probability, we define another parameter of ρ :

$$\rho_i = \begin{cases} 0 & \text{if multiple (more than one) hosts share NAT} \\ 1 & \text{otherwise} \end{cases}$$

We can estimate the total number of victims that may run malicious sites successfully by applying β and ρ values to each /24 subnet, and it can be formalized as the following equation.

$$\text{Power of launching malicious services} = \sum_{i=1}^n \beta_i \rho_i N_i$$

In the case of Conficker, we found 345,609 /24 subnets over 1,339,698 have "1" for the *reputation* value and it is 25.79% of all subnets. While we can estimate *reputation* of a subnet easily, it is hard to know whether multiple hosts behind NAT or not. According to a recent study [18], around 34% - 45% of DSL networks are using one IP address to serve multiple hosts behind. Since most of Conficker infected subnets are DSL networks, we might apply this ratio to our estimation. Since we want to roughly estimate the number of infected hosts which can possibly launch malicious services, we conservatively select NAT rate of 45%. Thus, we randomly select 45% of subnets from the found subnets (i.e., 345,609 subnets) and remove them from our consideration. Finally, we have 190,085 subnets and 7,575,111 victims are in those /24 subnets, which are 30.4% of all infected victims. This means that a botmaster of Conficker may be able to launch more than 7 million malicious web sites with minimal detection.

Result 13. (Hosting Malicious Sites) *It is possible for a botmaster of Conficker to host a huge amount of malicious sites with minimum detection.*

Insight from Result 10, 11, 12 and 13. (High-performance botnet) *If a botmaster of Conficker performs malicious actions, she may ruin most networks or hosts easily with her huge amount of bots.*

⁹We can apply different levels of aggregation easily by changing the method of grouping.

VII. CAN NEIGHBORHOOD WATCH HELP?

Conficker still uses network scanning to infect other hosts on the Internet as previous worms and bots did, and it also adopts several advanced skills to infect hosts efficiently. The spreading techniques of Conficker can be classified into two categories [6], [22]; (i) *infecting random hosts* and (ii) *infecting nearby hosts*. Conficker has a function of scanning randomly selected IP addresses. Although this will help Conficker to spread globally, it is not probably very efficient these days because most networks are protected by firewalls or Network Intrusion Detection/Prevention Systems. To propagate more efficiently, Conficker adopts several interesting techniques to infect hosts nearby: (1) an ability to infect other hosts in the same subnet, (2) an ability to infect hosts in the nearby subnets [6], and (3) an ability to infect portable storage devices.

The diverse infection techniques of Conficker lead us to ask this question: “Which vector is more effective to infect hosts?”. Some previous studies suggested that second approach - (ii) *infecting nearby hosts* - is probably more dominant in the Conficker case [22], [16]. We think that this seems reasonable, because even though most networks are protected well from outside threats, they are still open to internal attacks. However, they do not show concrete evidence to support it.

To determine whether this hypothesis is correct, we constructed a test. Prior to explaining our test, we declare that we will use /24 subnet as a basic unit in our test, since Conficker scans neighbor /24 subnets to find new victims. And we make the following definition to simplify the test. We define two terms: (i) “camp” is the group of /24 subnets whose /16 subnet is the same and locations are close together, and (ii) each /24 subnet is a “neighbor” of nearby /24 subnets in the same camp. Sometimes, even if two /24 subnets are in the same /16 subnet, their physical locations could be far from each other. However, since our concept of “camp” is each /24 subnet with both nearby IP address and physical location, we should consider its location as well. The example case of “camp” and “neighbors” is shown in Figure 6. In this case, we assume that /24 subnets could be “neighbors” in the same “camp” if they are located within 100km and share the same /16 (i.e., 10.11.*.*) subnet prefix.

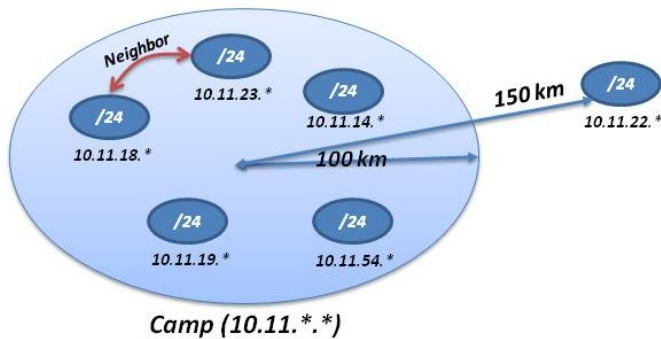


Fig. 6. Example camp and neighbor of 10.11/16 networks (assuming distance threshold is 100km).

Based on the above definition, we establish a hypothesis as follows. *Of the two infection vectors of Conficker, suppose the second infection vector plays a dominant role, the infection*

pattern¹⁰ of a /24 subnet will be similar to that of its “neighbors” in the same “camp”¹¹. In other words, the hosts in nearby networks of infected host are more likely to be selected [6] as future victims than randomly chosen hosts.

To evaluate the hypothesis, we have tested the following scenarios. First, we divide hosts into /24 subnets and assign each /24 subnet into a “camp” based on our definition. Second, we investigate the infection pattern of each /24 subnet to see whether the infection pattern of each /24 subnet is similar to its “neighbors”. We use *Variance-Mean Ratio (VMR)* [12] for a numerical expression. In this test, we measure the mean and variance value of the numbers of infected hosts of each /24 subnet in each “camp”, and calculate *VMR* for each “camp”. If the value of *VMR* is less than one, distribution of the data set shows under-dispersion with mean value in the center, which means that infection patterns of /24 subnets in the “camp” are very similar to each other.

Result 14. (Neighborhood) *Most /24 subnets show similar infection patterns (numbers of infected hosts) with their “neighbors”. The closer they are located with each other, the more similar in their infection patterns.*

We measured the *VMR* value of each “camp” and we found that more than 70% of “camps” denoted that their /24 subnet members are similar to each other. From this result, we reasonably infer that the dominant infection vector of Conficker is to infect nearby hosts. The test result is shown in Table IX. When we did this test, we got three types of “camps” based on its geographical information. For instance, if we set the distance metric for the “camp” as 100km which means that all /24 subnets in the “camp” have the same /16 subnet and they are within 100km of each other, we found 85,246 “camps” from our data and we discovered 62,121 “camps” whose /24 subnet members are similar to each other. We observed that more than 67% of “camps” showed that their /24 subnet members are similar to each other. The closer their locations are, the clearer this pattern is shown. This result tells us that Conficker is more likely to select nearby hosts than randomly chosen hosts and this means Conficker victims are mainly infected by neighbor networks/hosts. We deduce from this result that infection from the inside could be more harmful than the threats from the outside. Usually, most enterprise networks and ISPs protect their internal hosts using firewalls and IPS/IDS from external attacks, but there are very few approaches to protect hosts from internal threats.

Result 14.1 (Victim inference based on neighborhood information) *We could infer unknown victims by sharing and correlating neighbor alert information, even if we only know*

¹⁰We use the number of infected hosts of /24 subnet as a feature to represent an infection pattern.

¹¹On the contrary, if the first infection vector (random scanning) plays a more important role, then the victim infection pattern across neighbor networks in the same camp will be likely more different (and note that vulnerable hosts on the Internet is highly unevenly distributed as shown in previous studies [13]).

<i>Within Distance</i>	<i># of all "camps"</i>	<i># of "camps" whose /24 subnet members are similar to each other</i>
$\approx 100\text{km}$	85,246	62,121 (72.87%)
$\approx 200\text{km}$	65,748	44,633 (67.88%)
$\approx 300\text{km}$	54,415	36,495 (67.06%)

TABLE IX
THE NUMBER OF ALL "CAMPS" AND "CAMPS" WHOSE MEMBERS ARE SIMILAR TO EACH OTHER.

small sets of families and its neighbors.

Based on previous results, we propose an approach of inferring (or early warning) emerging (unknown) infected /24 subnets using neighborhood information and we will show that the approach can infer unknown infected /24 subnets with more than 90% of accuracy. From the above test, we find that Conficker victims share their infection patterns with their neighbors, and this finding gives us an intuition that collecting and sharing neighborhood information would be helpful to infer unknown victims or provide early warnings. To validate this intuition, we have tested the simple scenario of "We only have small portions of information of benign and malicious hosts, but we can gather neighborhood information. Then, how many unknown malicious hosts can we infer based on neighborhood information?".

As a method of considering neighborhood information, we use the K-Nearest Neighbor (KNN) classification algorithm, because it is a very popular approach that classifies unknown examples using the most similar "neighbors" in the known examples. When we apply the KNN algorithm to our data, we need the following preparations.

- **define classes:** *in this test, we define two classes: benign (normal /24 subnet) and malicious (/24 subnet which has Conficker victims)*
- **collect data:** *we use our Conficker data for malicious data, and we collected the same number of benign /24 subnets as malicious /24 subnets.*¹²
- **divide data:** *we randomly select 20% of data from both data sets for training samples and other 80% of data is used for testing.*

When all the preparation is completed, we apply the KNN algorithm, which checks neighbors iteratively, to our data. To efficiently run the KNN algorithm without high overhead, we simply choose a common setting of 3 for K. In addition, we use IP address location information to calculate the distance. We know the longitude and latitude of IP address, thus we can estimate the distance of two different IP addresses by measuring the differences between their longitude and latitude. After applying KNN to the data, we found that it can infer unknown infected /24 subnets with a high accuracy. As shown in Table X, even if we only know a small part of Conficker data (20%), we can infer other infected /24 subnets more than 90% accuracy with reasonable True Positive

¹²As a result, we have 1,300,000 malicious /24 subnets (infected by Conficker), and 1,300,000 benign /24 subnets (NOT infected by Conficker or other malware).

(TP) and False Positive (FP)¹³ rates. This inference result implies that if we share neighbor information, we could infer unknown victims or provide early warnings more efficiently.

<i>Inference Accuracy</i>	<i>TP rate</i>	<i>FP rate</i>
91.59%	91.65%	8.5%

TABLE X
ACCURACY, TP AND FP RATE OF THE INFERENCE APPROACH BASED ON NEIGHBORHOOD INFORMATION.

Insight from Result 14 and 14.1. (Neighborhood watch) *We observe that a large portion of victims could be infected by nearby victims and find that it is very important to share threat information with neighborhood networks. And this insight implies that further research is needed for developing new inference or early-warning approaches based on cooperated/shared (alert) information (and probably in an efficient privacy-preserving way).*

VIII. CONCLUSION

In this paper, we have studied a large-scale Conficker infection data to discover (i) their distribution over networks, ASes and etc, (ii) difference from previous bots/worms (iii) the effectiveness of current reputation-based malware detection/warning systems, and (iv) some insight to help detect future malware.

Our analysis of Conficker victims and cross-comparison results allowed us to obtain profound insights of Conficker victims. They also guide us to understand the trends of malware infections and to find interesting ideas that can aid the design of future malware detecting systems. We revealed that current reputation-based malware detecting systems that depend on previously known information are not enough to detect most Conficker victims. This result suggests that different kinds of (complementary) detection systems such as an anomaly-based detection system are needed. We measure the performance of Conficker and we believe that it provides more clear understanding the real effects of Conficker on the network or host. We provide a basis that proves the hypothesis of "A Conficker bot is more likely to infect nearby hosts than randomly chosen hosts" and we believe that it calls for more research of detection systems which are based on watching/sharing/correlating neighborhood information.

¹³TP denotes the rates that the algorithm classifies real malicious networks correctly, and FP denotes the rates that the algorithm classifies benign networks as malicious.

ACKNOWLEDGMENT

We would like to thank Sven Dietrich for his comments and feedback to improve the paper. This material is based upon work supported in part by the Office of Naval Research under Grant no. N00014-09-1-0776, the National Science Foundation under Grant CNS-0954096, and the Texas Higher Education Coordinating Board under NHARP Grant no. 01909. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research, the National Science Foundation, and the Texas Higher Education Coordinating Board.

REFERENCES

- [1] SPAMCOP.NET. <http://www.spamcop.net/>.
- [2] The SPAMHAUS Project. <http://www.spamhaus.org/>.
- [3] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a Dynamic Reputation System for DNS. In *Proceedings of USENIX Security of Symposium*, Aug. 2010.
- [4] BotLab. BotLab: A Study in Spam. <http://www.botlab.org/>.
- [5] CAIDA. Conficker/Conflicker/Downadup as seen from the UCSD Network Telescope. <http://www.caida.org/research/security/ms08-067/conficker.xml>.
- [6] E. Chien. Downadup: Attempts at Smart Network Scanning. <http://www.symantec.com/connect/blogs/downadup-attempts-smart-network-scanning>.
- [7] DNSBL. invaluable DNSBL (an anti-spam blacklist). <http://dnsbl.invaluable.com/>.
- [8] DShield. All suspicious Source IPs in DSIELD. http://www.dshield.org/feeds/daily_sources.
- [9] Dshield. Cooperative Network Security Community. <http://www.dshield.org/>.
- [10] FIRE. Finding Rogue Networks. <http://maliciousnetworks.org/>.
- [11] Fortune. Fortune 100 companies. <http://money.cnn.com/magazines/fortune/>.
- [12] U. G. and C. I. *Oxford Dictionary of Statistics (2nd edition)*. Oxford University Press, 2006.
- [13] G. Gu, Z. Chen, P. Porras, and W. Lee. Misleading and defeating importance-scanning malware propagation. In *Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks (SecureComm'07)*, September 2007.
- [14] T. Holz, C. Gorecki, and F. Freiling. Detection and Mitigation of Fast-Flux Service Networks. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*, Feb. 2008.
- [15] N. Ianelli and A. Hackworth. Botnets as a Vehicle for Online Crime. 2005.
- [16] S. Krishnan and Y. Kim. Passive identification of Conficker nodes on the Internet. In *University of Minnesota - Technical Document*, 2009.
- [17] J. Kristoff. Experiences with Conficker C Sinkhole Operation and Analysis. In *Proceedings of Australian Computer Emergency Response Team Conference*, May 2009.
- [18] G. Maier, F. Schneider, and A. Feldmann. Nat usage in residential broadband networks. In *Proceedings of the 12th international conference on Passive and active measurement (PAM'11)*, 2011.
- [19] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. In *Proceedings of IEEE Security and Privacy*, May 2003.
- [20] D. Moore, C. Shannon, and K. Calff. Code-red: a case study on the spread and victims of an internet worm. In *Proceedings of ACM SIGCOMM Workshop on Internet Measurement*, Nov. 2002.
- [21] B. N. Online. Clock ticking on worm code. <http://news.bbc.co.uk/2/hi/technology/7832652.stm>.
- [22] P. Porras, H. Saidi, and V. Yegneswaran. A Foray into Conficker's Logic and Rendezvous Points. In *Proceedings of USENIX LEET*, Apr. 2009.
- [23] Z. Qian, Z. Mao, Y. Xie, and F. Yu. On Network-level Clusters for Spam Detection. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*, Feb. 2010.
- [24] Z. Qian, Z. Mao, Y. Xie, and F. Yu. Spam Mitigation using Spatio-Temporal Reputations from Blacklist History. In *Proceedings of Annual Computer Security Applications Conference (ACSAC)*, Dec. 2010.
- [25] A. Ramachandran and N. Feamster. Understanding the Network-Level Behavior of Spammers. In *Proceedings of ACM SIGCOMM*, Sep. 2006.
- [26] SecureWorks. Ozdok/Mega-D Trojan Analysis. <http://www.secureworks.com/research/threats/ozdok/?threat=ozdok>.
- [27] C. Shannon and D. Moore. The Spread of the Witty Worm. In *Proceedings of IEEE Security and Privacy*, May 2004.
- [28] S. Shin and G. Gu. "Conficker and Beyond: A Large-Scale Empirical Study". In *Proceedings of 2010 Annual Computer Security Applications Conference (ACSAC)*, Dec. 2010.
- [29] SORBS. Fighting spam by finding and listing Exploitable Servers. <http://www.au.sorbs.net/>.
- [30] SRI-International. An analysis of Conficker C. <http://mtc.sri.com/Conficker/addendumC/>.
- [31] B. Stock, M. E. Jan Goebel, F. C. Freiling, and T. Holz. Walowdac Analysis of a Peer-to-Peer Botnet. In *Proceedings of European Conference on Computer Network Defense (EC2ND)*, Nov. 2009.
- [32] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *Proceedings of ACM CCS*, Nov. 2009.
- [33] Symantec. Trojan.Srizbi. http://www.symantec.com/security_response/writeup.jsp?docid=2007-062007-0946-99.
- [34] M. S. Techcenter. Conficker worm. <http://technet.microsoft.com/en-us/security/dd452420.aspx>.
- [35] Tmetric. Bandwidth Measurement Tool. <http://mbacarella.blogspot.com/projects/tmetric/>.
- [36] UPI. Virus strikes 15 million PCs. http://www.upi.com/Top_News/2009/01/26/Virus-strikes-15-million-PCs/UP1-19421232924206/.
- [37] Verisign. The Domain Name Industry Brief. <http://www.verisign.com/domain-name-services/domain-information-center/domain-name-resources/domain-name-report-sept09.pdf>.
- [38] D. Watson. Know Your Enemy: Containing Conficker. <http://www.honeynet.org/papers/conficker>.
- [39] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldzmid, and T. Wobber. How Dynamic are IP Addresses? In *Proceedings of ACM SIGCOMM*, Aug. 2007.
- [40] Y. Xie, F. Yu, K. Achan, R. Panigraphy, G. Hulte, and I. Osipkov. Spamming Botnets: Signatures and Characteristics. In *Proceedings of ACM SIGCOMM*, Aug. 2008.

APPENDIX A

CONFICKER DOMAIN GENERATION

While our paper here focuses on analyzing the infected machines or bots, we give a brief description of the domain generation of Conficker botnet for completeness. Conficker uses domain fluxing and IP fluxing techniques to evade detection and containment. Conficker botnet has evolved over time and comes in different versions. Conficker uses random names for the C&C servers and employs multiple IP addresses for the C&C servers. Conficker.A bots [22] generate 250 domains every three hours, using the current date and time as the seed. All the bots generate the same domain names every day and the C&C server can be hosted at one of these domain names. In order to make it harder for a security vendor to pre-register the domain names, Conficker.C [30] increased the number of randomly generated domain names per bot to 50K.

APPENDIX B

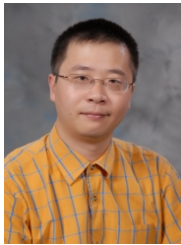
CONFICKER SINKHOLE SERVER

Conficker sinkhole servers are operated by several companies and organizations (e.g., Georgia Tech, AOL, F-Secure, ShadowServer, Support Intelligence, SIE, and Symantec, later known as the Conficker Working Group) and have been started in February 2009. The operators have purchased domains used by the Conficker bots and redirected to their servers. In preparation of Conficker.C, which would generate 50K domains each day over 116 TLDs, ICANN worked with

them to dub Conficker Cabal, and the TLD operators block, monitor, and/or redirect domains to the conficker sinkholes. The sinkholes run web servers and collect web logs. Each sinkhole uploads its logs to a central repository used for remediation reporting and for academic research.



Seungwon Shin is a Ph.D student in the Department of Electrical and Computer Engineering at Texas A&M University working with Dr. Guofei Gu in SUCCESS Lab. He received his B.S. and M.S. degrees in Electrical Engineering from KAIST in Korea. His research interests include analyzing and detecting botnet, and protecting cloud computing environments from threats.



Guofei Gu is an assistant professor in the Department of Computer Science & Engineering at Texas A&M University (TAMU). Before coming to Texas A&M, he received his Ph.D. degree in Computer Science from the College of Computing, Georgia Institute of Technology. His research interests are in network and system security, such as malware analysis/detection/defense, intrusion/anomaly detection, web and social networking security. Dr. Gu is a recipient of 2010 NSF CAREER award and a co-recipient of 2010 IEEE Symposium on Security and

Privacy (Oakland'10) best student paper award. He is currently directing the SUCCESS (Secure Communication and Computer Systems) Lab at TAMU.



Narasimha Reddy received a B.Tech. degree in Electronics and Electrical Communications Engineering from the Indian Institute of Technology, Kharagpur, India, in August 1985, and the M.S. and Ph.D degrees in Computer Engineering from the University of Illinois at Urbana-Champaign in May 1987 and August 1990 respectively. Reddy is currently a J.W. Runyon Professor in the department of Electrical and Computer Engineering at Texas A & M University. Reddy's research interests are in Computer Networks, Storage Systems, Multimedia

systems, and Computer Architecture. During 1990-1995, he was a Research Staff Member at IBM Almaden Research Center in San Jose. Reddy holds five patents and was awarded a technical accomplishment award while at IBM. He has received an NSF Career Award in 1996. He was a faculty fellow of the College of Engineering at Texas A&M during 1999-2000. His honors include an outstanding professor award by the IEEE student branch at Texas A&M during 1997-1998, an outstanding faculty award by the department of Electrical and Computer Engineering during 2003-2004, a Distinguished Achievement award for teaching from the former students association of Texas A&M University and a citation "for one of the most influential papers from the 1st ACM Multimedia conference". Reddy is a Fellow of IEEE Computer Society and is a member of ACM.



Christopher P. Lee When not playing peekaboo with his young daughter, Dr. Chris Lee researches Information Assurance and Cyber Crime at George Mason University (GMU) and the Internet Systems Consortium (ISC). Chris' current work focuses on sinkholing botnets, monitoring DNS traffic for anomaly detection, and using offensive network tactics for defense of critical network assets. Dr. Lee earned his Ph.D. from the Georgia Institute of Technology in Electrical and Computer Engineering with a thesis on modeling botnets (Rubot), with a keen focus on peer to peer command structures. Much of this work was inspired by monitoring projects of Nugache, Mayday, Storm, and Conficker botnets. In the future, Chris desires to focus on measuring the practical effectiveness of various security practices and evaluating methods of sharing security indicators across security, industry, and governmental parties.