Collaborative Detection of Fast Flux Phishing Domains

Chenfeng Vincent Zhou, Christopher Leckie and Shanika Karunasekera
Department of Computer Science and Software Engineering, The University of Melbourne, Australia
Email: {cvzhou, caleckie, shanika}@csse.unimelb.edu.au

***Abstract*— Phishing is a significant security threat to users of Internet services. Nowadays, phishing has become more resilient to detection and trace-back with the invention of Fast Flux (FF) service networks. We propose two approaches to correlate evidence from multiple DNS servers and multiple suspect FF domains. Real-world experiments show that our correlation approaches speed-up FF domain detection, based on an analytical model that we propose to quantify the number of DNS queries needed to confirm a FF domain. We also show how our correlation scheme can be implemented on a large scale by using a decentralized publish-subscribe correlation model called LarSID, which is more scalable than a fully centralized architecture.**

***Index Terms*— phishing, fast flux service networks, collaborative intrusion detection, round-robin DNS**

I. Introduction

Phishing is a form of social engineering attack, which exploits human vulnerabilities rather than software vulnerabilities. In order to initiate phishing scams, the phishers (the operator of phishing sites) send out a large number of fraudulent emails that include a link to the website under their control. These emails normally spoof a reputable company, and encourage a quick reply so that users' information can be collected before the phishing site is taken down [5]. The potential victim then connects to a spoofed website by clicking on the link provided by the spam email. An accurate imitation of the legitimate organization's website is presented to ensure that the victim enters their personal details. The compromised details can then be used by the phisher for financial gain. Gartner has estimated the cost of identity theft increased from $2bn to $3.2bn in 2007 in the USA alone [7], [8].

In order to address this security challenge, significant resources have been invested in anti-phishing research. A range of anti-phishing tools have been proposed [2], [3], [6], as well as more fundamental research [5], [12], [13], [17]. One of the most common approaches to the problem has been phishing site *take-down* [18], *i.e.*, removing a fraudulent website. A key part of any take-down procedure is the problem of *trace-back* of hosting systems for phishing websites, *i.e.*, finding the underlying computers that host the site. Traditional phishing hosts can be traced back relatively quickly based on their public DNS name or directly by their IP address if it is embedded within the original spam email. Further action can then be taken against the identified phishing host.

In order to protect their criminal assets, the operators of phishing sites invented a better architecture called *Fast Flux (FF)* service networks to hide the hosting machine of phishing websites from trace-back. In FF networks, a large number of proxy hosts are used to relay requests to the back-end server, called a *mothership*, that actively hosts the phishing site (called the *FF domain*). Hundreds or even thousands of compromised computers can be used as the *front-end proxies*. The DNS infrastructure is then used to map the phishing domain name to different front-end proxies. This multi-layer architecture makes it extremely difficult to trace-back the hosting machine, which is hiding behind many front-end proxies.

Recent research has identified a method for detecting possible FF domains by searching for domains that are associated with many IP addresses and use short time-to-live (TTL) values in their DNS query results [9], [28]. There are two potential limitations of this approach. (1) We can expect FF domain operators to send fewer IP addresses from a single query, so that the above detection method can be evaded. (2) It may require multiple queries to confirm whether a suspect domain is actually a FF domain, which increases the time required for FF domain detection. In order to address these limitations, our aim in this paper is to reduce the time required to detect FF domains by correlating evidence from multiple sources.

In this paper, we present several approaches to correlating evidence about FF domains. We first begin by characterising the behavior of FF domains based on evidence collected from multiple points around the Internet on real FF domains. Based on an analysis of the behavior of the FF domains, we present an analytical model of the number of DNS queries needed to confirm a FF domain, and we use this analytical model to motivate our approach to correlation. We then present two approaches to correlating FF evidence to speed up detection: (1) correlate IP addresses using queries from multiple DNS servers; (2) correlate results of queries from multiple possible FF domains. In both cases, we empirically demonstrate the potential speed up in detection using these correlation schemes. Finally, we consider the problem of how to correlate evidence on a large scale across the Internet.

The rest of this paper is organized as follows. In Section II, we describe the background of phishing

An extended version of this paper can be found in our technical report "Collaborative Detection of Fast Flux Phishing Domains," by C. V. Zhou, C. Leckie, S. Karunasekera, which can be located at http://www.cs.mu.oz.au/∼cvzhou/pub/jnw_full.pdf.

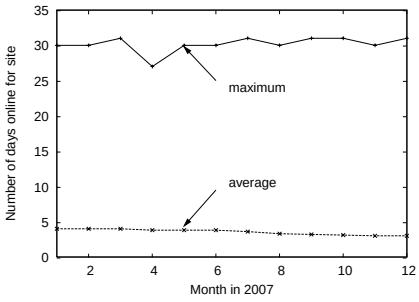


Figure 1. Lifetime for phishing sites observed in 2007

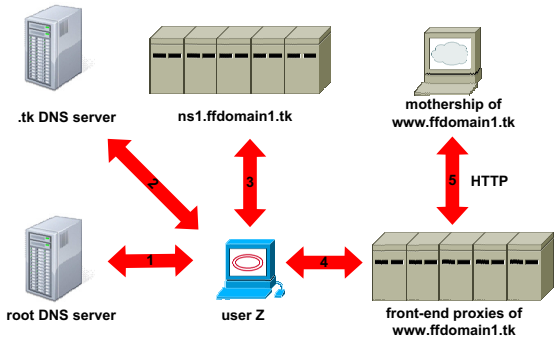


Figure 2. A real-world example of a fast flux service network

domains, and highlight the detection challenges of FF domains. We then propose a theoretical model to analyze the problem of FF domain detection in Section III. We propose two correlation schemes for FF domain detection based on multiple DNS servers and multiple FF domains, and present evaluation results based on a real-world experiment in Section IV. In Section V, we introduce a decentralized platform to support the proposed correlation schemes. We review the related work in Section VI, and conclude the paper in Section VII.

II. Background on Phishing Domains

In this section, we first describe recent trends in phishing attacks and their corresponding impact on Internet security. We next explain the technical details of a new technique, Fast Flux (FF) service networks, which are used to host phishing domains (note that we refer to domains that are hosted by FF networks as *FF domains* in the rest of this paper). We then revisit several preliminary efforts to detect FF domains in the literature, and finally highlight the open research issues in FF domain detection that motivate our work in the following sections.

A. Trends and Impact of Phishing

Since phishing made its debut in the mid 1990's by targeting America Online, it has become a major threat to online services. According to the data collected by the APWG (Anti-Phishing Work Group) [1] in 2007, there were more than 20,000 unique phishing domains reported every month in 2007 except February. With increased efforts in anti-phishing, the average online time for phishing domains has decreased from 4 days to 3 days, as shown in Figure 1. However, the longest online time has remained steady at 30 days, due to the evolution of phishing techniques, such as FF service networks [23].

B. Fast Flux Service Networks

A Fast Flux (FF) service network is a term coined in the anti-spam community to describe a decentralized *botnet* (*i.e.*, a network of compromised computer systems) with constantly changing public DNS records, which enables their controller (the attacker) to hide and sustain malicious websites [23]. The method of changing IP addresses dynamically is borrowed from a technique called *round-robin* DNS, which is used by legitimate companies for load balancing. A round-robin DNS works on a rotating basis so that in response to a DNS request, the least recently used IP address is selected from a list. In a FF network, compromised machines are used merely as frontline proxies, so that the attackers are safely hidden behind a veil of constantly shifting IP addresses.

From the attacker's perspective, the FF architecture with multiple proxies helps to protect their criminal assets. Multiple proxies can be used for load balancing during the phishing attack. Moreover, thousands of front-end proxies can confuse any attempts to trace the phishing server, and make it difficult to shut-down the front-end proxies.

In general, the computer systems in FF service networks can be divided into two layers based on their functionality. The *front-end proxies* are a large pool of compromised computer systems that serve as the first layer of a FF network. They are normally used for blind proxy redirection. At first, their IP addresses are assigned to the same fully qualified domain name. These addresses are swapped frequently in a round-robin manner by using very short TTL values for each given DNS resource record. Then the corresponding request and data to the advertised domain name will be funelled to and from the second layer - the *mothership* that actually delivers content back to the misled user who requests it.

Figure 2 describes the layered architecture of a real-world FF network that hosts the phishing domain *www.ffdomain1.tk* (note that the phishing domain names appearing in this paper have been anonymized to preserve privacy), and also illustrates the DNS query process for this FF domain. As shown in Figure 2, after user Z clicks on the link of *www.ffdomain1.tk* that is embedded inside a spam message and points to a domain owned by the phisher, the DNS lookup process can be analyzed as follows:

1) User Z queries the DNS root nameserver (usually through their browser) for the top-level domain *.tk* and receives an answer.
2) User Z asks the *.tk* nameserver for the domain *ffdomain1.tk*, and is referred to a nameserver *ns1.ffdomain1.tk*, which is a DNS server under the control of the attacker or an illegal DNS server from a country where the DNS servers are loosely monitored. In this example, the IP address of the nameserver keeps changing frequently, which is a double-flux architecture based on the definition

TABLE I.
A SAMPLE DNS ENTRY FOR FAST FLUX DOMAIN *www.ffdomain1.tk*

Domain Name	TTL	Class	Types	Type-specific-data
ffdomain1.tk.	600	IN	A	006.49.240.42
ffdomain1.tk.	600	IN	A	011.181.95.149
ffdomain1.tk.	600	IN	A	015.125.239.184
ffdomain1.tk.	600	IN	A	019.128.83.145
ffdomain1.tk.	600	IN	A	027.88.94.3
ffdomain1.tk.	6462	IN	NS	*ns1.ffdomain1.tk.*
ns1.ffdomain1.tk.	583	IN	A	007.157.114.207
ns1.ffdomain1.tk.	583	IN	A	016.65.34.228
ns1.ffdomain1.tk.	583	IN	A	019.234.199.93
ns1.ffdomain1.tk.	583	IN	A	020.234.127.147
ns1.ffdomain1.tk.	583	IN	A	023.138.52.7

in [23]. If it is a single-flux architecture, the IP address for the nameserver remains unchanged for a comparatively longer time period, such as 24 hours.

3) User Z queries the authoritative name server *ns1.ffdomain1.tk* for the actual IP address of *www.ffdomain1.tk* and receives an IP address from the pool of FF front-end proxies, which is normally the IP address of a compromised home computer.
4) User Z then initiates direct communication with the IP address that is assigned to *www.ffdomain1.tk*, *e.g.*, 027.88.94.3 (note that the first eight bits of any IP address appearing in this paper have been anonymized to preserve privacy). This IP address is usually frequently changed.
5) All the HTTP requests from user Z to the front-end proxy (027.88.94.3) are actually redirected to the mothership that will respond to the victim.

A snapshot of the corresponding DNS lookup results are shown in Table I. This shows 5 DNS A records that were returned by a DNS query for the domain *ffdomain1.tk*. Note that the TTL field of these A records is set to 600 seconds (10 minutes). This means that the front-end proxies returned for this FF domain will change every 10 minutes in a round-robin manner. In the bottom of Table I, we see that multiple hosts are used as the nameserver *ns1.ffdomain1.tk*, which change after 583 seconds.

In traditional phishing scams, the resolvable main IP address of the phishing domain can normally be located, hence the corresponding connection can be blocked. However, the distributed, constantly changing infrastructure of a FF domain makes it impractical to trace-back the real hosting machine (the FF mothership) and shut down its operations. If we try to shut down the IP address that the phishing domain is currently resolved to, there are thousands of other candidate IP addresses to be resolved in the phishing domain by the DNS server. In this double-flux service network, even the IP addresses of the DNS nameservers keep changing.

C. Previous Work on Fast Flux Domain Detection

In order to address the challenges of FF domain detection, there are several approaches that have been proposed [9], [28]. These approaches focus on detecting the front-end proxies in FF networks, using distinct features, such as short TTL values and multiple A records from the DNS query results of a FF domain. In [28], we proposed a multi-layer FF domain trace-back approach. A suspect FF domain is tested using DNS queries based on the number of unique A records returned, the TTL value, and the similarity of the domain name to a well-known domain.

Holz *et al.* [9] proposed a similar approach to FF network detection and measurement, during the same period as our research in [28] was being conducted. In order to distinguish between a FF domain and a legitimate domain, a FF score is calculated based on the number of unique A records, the number of NS records and the number of unique autonomous systems (ASs) from the results of DNS queries for the suspect domain.

D. Open Research Issues

Several open research problems are raised by this existing research into the detection of FF domains. First, as described in the previous subsection, both detection approaches [28] and [9] are based on the principle that the results of a DNS query to a FF domain will contain many unique IP addresses. In order to evade this approach to detection, we may expect attackers to become more stealthy by limiting the number of FF front-end proxies returned for a single DNS query. This raises the question of how we detect a FF domain based if a smaller number of IP addresses are returned with each query?

Second, it is important to detect FF domains as quickly as possible in order to minimize the damage they can cause. We have observed that the IP addresses returned in response to a DNS query to a FF domain appear to be drawn at random from a pool of available IP addresses. If DNS queries from different networks return different IP addresses, then by combining the list of IP addresses observed from different locations in the Internet, is it possible to confirm that a domain is a FF domain more quickly? If so, can we quantify the benefit of combining the results of DNS queries from multiple networks?

Third, another trend we have observed is that different phishing domains are operating on the same FF network infrastructure. Thus, several FF phishing domains can use the same IP address as a proxy. Consequently, can we correlate IP addresses across different suspect phishing domains in order to speed-up detection?

III. ANALYTICAL MODEL FOR IDENTIFYING FAST FLUX DOMAINS

In this section, we first propose a theoretical approach to model the time required to confirm that a given domain is actually a FF domain, by analyzing the relationship between the number of DNS queries and the number of unique IP addresses returned. We then extend this analytical model to quantify the time that can be saved by combining DNS queries from different networks.

Consider the case of a FF domain containing H front-end proxies, where we need to observe a minimum of h

unique IP addresses belonging to a domain before it is considered to be a FF domain. Each DNS query samples the set of hosts, and returns an IP address that is drawn at random from the set of H possible hosts. Assuming that samples are independent, in some cases a query will return a new IP address, while in other cases it will return an IP address that we have already seen. Let $X_{h,H}$ be a random variable that denotes the number of queries issued before we observe the minimum number h of unique IP addresses (out of a possible set of all H IP addresses that belong to the domain) required to consider the domain to be a FF domain. The aim of our analysis is to determine the expected number of queries $E(X_{h,H})$ in order to observe h unique hosts from a set of H hosts.

Based on our real-time monitoring of several FF domains, we can divide the process of identifying the IP addresses that belong to a FF domain into two phases. In the *discovery* phase, we are sampling from a large, static pool of IP addresses used in FF networks, while in the *stable* phase new front-end proxies are gradually being added to the initial pool of IP addresses. For example, Figure 3 plots the number of unique IP addresses identified by 480 DNS queries for the FF domain *www.ffdomain1.tk* in August 2008. The DNS lookup was conducted every 10 minutes, and there were 14 IP addresses returned by the FF domain *www.ffdomain1.tk* for each query. As shown in Figure 3, the results from the first 20 queries form the discovery phase, where there were 14 out of approximately 115 unique IP addresses returned each time in a random manner, as the front-end proxies for the FF domain *www.ffdomain1.tk*. The remaining lookup results form the stable phase, where new IP addresses are added gradually to the FF network, *i.e.*, 14 IP addresses are selected from an increasing number of IP addresses each time. Given that in the discovery phase we can identify a set of h hosts with the fewest queries, we model the discovery phase as a random sampling problem.

We model the discovery phase as an example of the *coupon collector problem* [16], which finds the expected number of samples needed (with replacement) from a population of H objects in order to sample each object at least once. It can be shown [16] that if $Y_{i,H}$ denotes the number of samples made in order to go from having seen $i-1$ objects to i objects, then

$$E(Y_{i,H}) = \frac{H}{H-i+1}, and$$

$$\begin{aligned} E(X_{H,H}) &= E(Y_{1,H}) + E(Y_{2,H}) + \cdots + E(Y_{H,H}) \\ &= H\sum_{i=1}^{H}\frac{1}{i}. \end{aligned}$$

In our case, we wish to observe h out of H hosts, so

$$\begin{aligned} E(X_{h,H}) &= E(Y_{1,H}) + E(Y_{2,H}) + \cdots + E(Y_{h,H}) \\ &\cong H[lnH - ln(H-h)] \\ &= O(Hln\frac{H}{H-h}). \end{aligned}$$

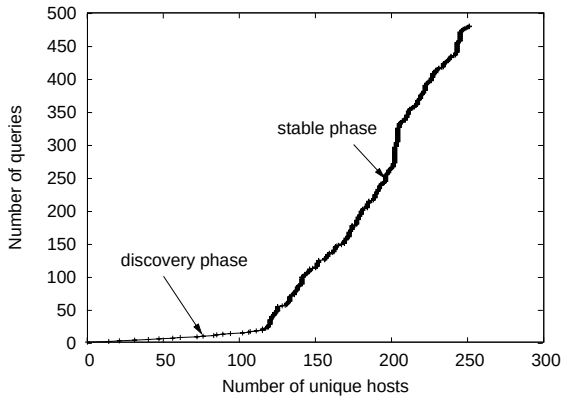


Figure 3. Number of DNS queries needed to find a given number of unique IP addresses for FF domain *www.ffdomain1.tk*

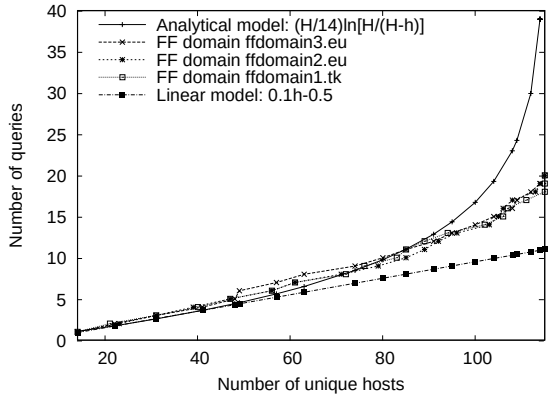


Figure 4. Approximation of number of DNS queries for FF domain detection

If the observation is based on a single DNS server, *e.g.*, local DNS queries are generated for a suspect domain at an average rate of ρ queries per unit time, then the expected time to detect a sufficient number of unique IP addresses to consider the domain a FF domain is

$$T = O(\frac{H}{\rho}ln\frac{H}{H-h}).$$

To validate this analytical model, we have shown in Figure 4 our proposed model against the actual number of queries required for three different FF domains. We observed that our model ($\frac{H}{\rho}ln\frac{H}{H-h}$, where $\rho = 14$ and $H = 115$) provides an upper bound on the number of queries required. As shown in Figure 4, the queries needed for these real life FF domains falls between a linear model and our analytical model, which indicates that the discovery phase is a mix of round-robin (linear model) and random sampling.

As proposed in [28], we have the potential to reduce the time required to detect at least h unique IP addresses by combining query results from different DNS servers. Namely, if we have m DNS servers, and each are queried at an average rate ρ, then by correlating their results, the expected time T required for detection becomes

$$T = O(\frac{H}{\rho m}ln\frac{H}{H-h}),$$

i.e., the theoretical speed-up in detection time is proportional to the number of DNS servers that participate in

monitoring the suspect domain. In the following sections we consider different approaches for correlating evidence from multiple sources about FF domains.

IV. Correlating Evidence of Fast Flux Phishing Domains

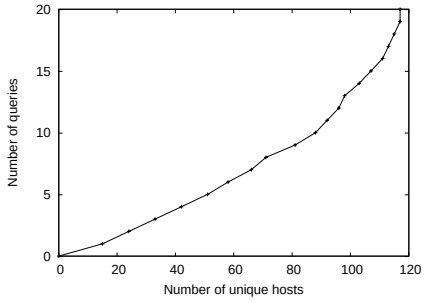
The speed of FF domain detection is limited by the frequency with which a single site will see new IP addresses being mapped to the suspect domain.

In order to address the limitations of using a single point of detection, we propose two correlation approaches for FF domains detection, in terms of correlating results (1) from different DNS servers; (2) from different suspect FF domains. In each case, we consider empirical evidence from actual FF phishing domains to validate our approaches.

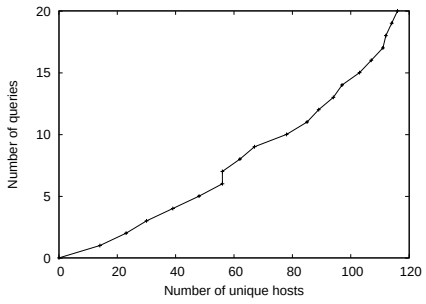
A. Correlating Evidence from Multiple DNS Servers

The motivations for this approach are as follows:

- If different DNS servers see different results for the same DNS query, then there is a benefit in combining evidence from multiple DNS servers.
- As described in Section III, there is potentially a linear speed-up in detection time when we increase the number of DNS servers that participate in monitoring the suspect domain, based on our analytical model.



(a) Query results from a DNS server in Brazil



(b) Query results from a DNS server in Switzerland

Figure 5. DNS queries of FF domain *www.ffdomain3.eu*

We focus our study on the discovery phase of the FF front-end proxy deployment.

Figure 5 plots sample DNS query results of a FF domain *www.ffdomain3.eu* that we collected simultaneously from two different DNS servers: one in Brazil and another in Switzerland. Query results from both DNS servers have a similar trend in the number of unique IP addresses identified in the discovery phase. As shown in Figure 5(a) the DNS server in Brazil reported that 14 unique IP addresses were assigned to FF domain *www.ffdomain3.eu* in the first query, and 117 unique IP addresses after 20 queries. Similar behavior can be observed from the DNS server in Switzerland.

TABLE II.
SAMPLE DNS QUERY RESULTS FROM A DNS SERVER IN BRAZIL

;; ANSWER SECTION:
ffdomain3.eu. 600 IN A 001.181.31.106
ffdomain3.eu. 600 IN A 002.8.35.209
ffdomain3.eu. 600 IN A 003.184.35.240
ffdomain3.eu. 600 IN A 004.94.99.152
ffdomain3.eu. 600 IN A 007.137.128.99
ffdomain3.eu. 600 IN A 010.178.217.58
ffdomain3.eu. 600 IN A 012.69.170.118
ffdomain3.eu. 600 IN A 012.98.71.156
ffdomain3.eu. 600 IN A 013.147.2.253
ffdomain3.eu. 600 IN A 017.102.47.94
ffdomain3.eu. 600 IN A 018.100.69.190
ffdomain3.eu. 600 IN A 020.130.9.214
ffdomain3.eu. 600 IN A 024.46.80.193
ffdomain3.eu. 600 IN A 025.131.109.236

TABLE III.
SAMPLE DNS QUERY RESULTS FROM A DNS SERVER IN SWITZERLAND

;; ANSWER SECTION:
ffdomain3.eu. 600 IN A 001.181.31.106
ffdomain3.eu. 600 IN A 001.208.14.142
ffdomain3.eu. 600 IN A 001.210.33.238
ffdomain3.eu. 600 IN A 001.76.83.55
ffdomain3.eu. 600 IN A 002.158.225.101
ffdomain3.eu. 600 IN A 003.184.35.240
ffdomain3.eu. 600 IN A 010.178.217.58
ffdomain3.eu. 600 IN A 011.191.248.113
ffdomain3.eu. 600 IN A 012.183.220.208
ffdomain3.eu. 600 IN A 012.69.170.118
ffdomain3.eu. 600 IN A 012.98.71.156
ffdomain3.eu. 600 IN A 015.109.227.159
ffdomain3.eu. 600 IN A 017.102.47.94
ffdomain3.eu. 600 IN A 028.183.196.29

Figure 5 shows a scenario where each DNS server works independently to detect the FF domain. If we can combine the results from both DNS servers, intuitively we could gather more evidence of the FF domain, and hence potentially speed up detection. However, the reduction in the detection time depends on the extent to which queries from different servers are independent or uncorrelated. Tables II and III show the results of a sample DNS query for the FF domain *www.ffdomain3.eu* generated simultaneously on two different sites. In this example, 6 IP addresses were reported by both sites.

Motivated by this example, we formalize the correlation scheme as follows. We consider a set of DNS servers $\mathcal{D} = \{d_i | i = 1, 2, \dots, m\}$ which come from different network domains or different ISPs. Each DNS server d_i issues queries at an average rate ρ for a suspect FF domain f, and correlates their results to gather evidence of potentially suspicious FF domains. The DNS servers collaborate to share the evidence they have collected.

The evidence that is shared can potentially take many forms. We propose a correlation scheme based on the DNS A records (*i.e.*, a set of suspicious IP addresses that are potentially being compromised) that are associated with the FF domain f. We refer to the set of suspicious IP addresses that have been observed by DNS server d_i as domain f's *proxy list (PL), i.e.*,

$$PL_i = \{s_{ij} \in IP | j = 1, 2, \dots, n_i\},$$

where s_{ij} is the j^{th} IP address that has been associated with domain f as seen by DNS server d_i.

Consider the case where all participating DNS servers correlate their proxy lists for domain f after each query. We denote the resulting proxy list of domain f after t queries (or being correlated t times) as the *combined*

proxy list (CPL), i.e.,

$$CPL_t = CPL_{t-1} \cup PL_1 \cup PL_2 \cup \ldots \cup PL_m.$$

Hence, the number of unique IP addresses identified after t queries is $h_t = |CPL_t|$.

In order to evaluate the effect of correlating evidence from multiple DNS servers, we continuously monitored several actual FF domains, which used the Danmec/Asprox SQL injection attack tool [22] for two weeks in August 2008, from seven different sites on PlanetLab [20] across three continents. In particular, we generated DNS queries every 10 minutes for monitored FF domains on seven DNS servers simultaneously. The DNS servers were selected to provide geographical diversity, and are from Brazil, United States, Russia, Singapore, Australia, Switzerland and Netherlands.

TABLE IV.
SAMPLE DNS QUERY RESULTS OF FF DOMAIN *www.ffdomain1.tk*

;; ANSWER SECTION:
ffdomain1.tk. 600 IN A 002.57.70.209
ffdomain1.tk. 600 IN A 004.209.243.172
ffdomain1.tk. 600 IN A 005.209.75.40
ffdomain1.tk. 600 IN A 007.223.178.4
ffdomain1.tk. 600 IN A 009.80.11.108
ffdomain1.tk. 600 IN A 011.181.105.112
ffdomain1.tk. 600 IN A 012.183.220.208
ffdomain1.tk. 600 IN A 012.251.254.179
ffdomain1.tk. 600 IN A 012.69.170.118
ffdomain1.tk. 600 IN A 015.108.209.42
ffdomain1.tk. 600 IN A 015.109.227.159
ffdomain1.tk. 600 IN A 017.102.44.173
ffdomain1.tk. 600 IN A 019.235.222.87
ffdomain1.tk. 600 IN A 026.255.101.5

TABLE V.
SAMPLE DNS QUERY RESULTS OF FF DOMAIN *www.ffdomain2.eu*

;; ANSWER SECTION:
ffdomain2.eu. 600 IN A 002.57.70.209
ffdomain2.eu. 600 IN A 003.184.35.240
ffdomain2.eu. 600 IN A 004.209.243.172
ffdomain2.eu. 600 IN A 005.209.75.40
ffdomain2.eu. 600 IN A 007.223.178.4
ffdomain2.eu. 600 IN A 008.240.173.146
ffdomain2.eu. 600 IN A 009.80.11.108
ffdomain2.eu. 600 IN A 011.8.98.176
ffdomain2.eu. 600 IN A 012.251.254.179
ffdomain2.eu. 600 IN A 015.108.209.42
ffdomain2.eu. 600 IN A 015.109.227.159
ffdomain2.eu. 600 IN A 017.102.44.173
ffdomain2.eu. 600 IN A 019.235.222.87
ffdomain2.eu. 600 IN A 026.255.101.5

Figure 6 shows the correlation results in the discovery phase of a FF domain across seven different DNS servers. Figure 6 plots both the individual results PL_i and the correlation results CPL for the FF domain *www.ffdomain2.eu* in terms of the number of unique IP addresses that can be identified for a given number of queries. As shown in Figure 6, the individual DNS servers require a similar number of DNS queries in order to detect a given number of unique IP addresses from the FF domain. In comparison, the correlated results are consistently able to identify a given number of unique IP addresses using fewer DNS queries, *e.g.*, we can detect 100 unique IP addresses using only 9 queries based on the correlated results, compared with up to 17 queries in the worst case for the individual servers. Similar results were achieved for other FF domains in both individual and correlation cases. Note that the speed-up in detection is sub-linear, rather than linear as discussed in Section III for the ideal case, since the query results from each DNS server are not fully independent.

B. Correlating Evidence from Multiple FF Domains

During our large-scale monitoring of FF domains, we made an interesting observation about the use of FF service networks. We observed that multiple FF domains use the *same* pool of IP addresses in their FF service network. As shown in Figure 7, two different FF domains observed from a single DNS server have almost identical behavior in terms of the number of unique IP addresses identified during 480 queries. Further testing highlighted that there was a significant overlap in the IP addresses used. As shown in Tables IV and V, 11 out of 14 IP addresses for a single query are identical between these two domains. The example shows that both FF domains are using the same group of IP addresses as their FF proxies. This indicates that either both domains are controlled by the same attacker, or the same FF service network is being "rented out" to host multiple FF domains. To the best of our knowledge, this use of a common FF service network for multiple domains has not been reported in the literature.

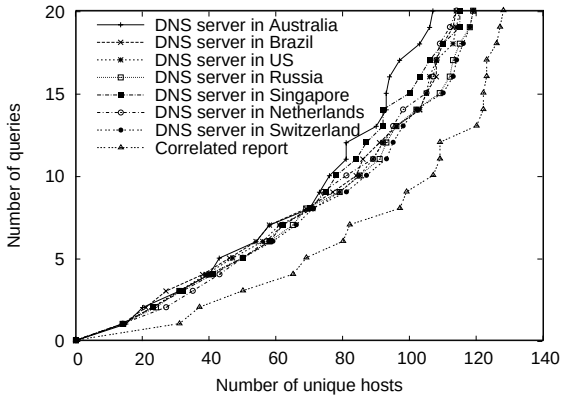
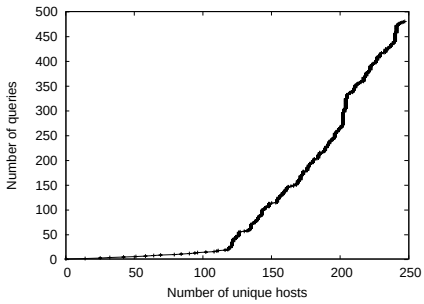
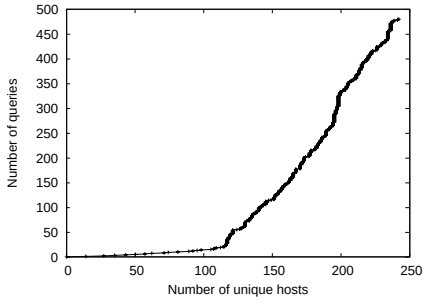


Figure 6. Effect of DNS query correlation from seven DNS servers



(a) Query results of FF domain *www.ffdomain1.tk*



(b) Query results of FF domain *www.ffdomain2.eu*

Figure 7. DNS queries of two different FF domains from a single DNS server

We found that this characteristic of the same IP addresses being used in multiple FF domains can also be observed in data that was collected in June 2008 by the

University Mannheim, Germany [10].

Motivated by these observations, we propose a model of FF domain detection using a single DNS server as follows. If we see a potentially suspicious domain, but we lack sufficient evidence for confirmation, then if some of the IP addresses have already appeared in other FF domain names, we can speed up the confirmation process.

We formalize our correlation scheme as follows. We consider a set of FF domains $\mathcal{F} = \{f_i | i = 1, 2, \dots, m\}$ are monitored by a single DNS server. Each FF domain f_i is queried at an average rate ρ, and we correlate their results, in order to gather evidence of potentially suspicious FF proxies. In this case, we propose a correlation scheme based on the DNS A records (*i.e.*, a set of suspicious IP addresses that are potentially being compromised) that are associated with the FF domain f. We refer to the set of suspicious IP addresses that have been observed by the DNS server d_i as domain f_i's *proxy list (PL), i.e.*,

$$PL_i = \{s_{ij} \in IP | j = 1, 2, \dots, n_i\},$$

where s_{ij} is the j^{th} IP address that has been associated with domain f_i.

We correlate the proxy lists of all potential FF domains after each query. We denote the resulting proxy list of all monitored suspect domains $\mathcal{F}$ after t queries (or being correlated t times) as the *combined proxy list (CPL), i.e.*,

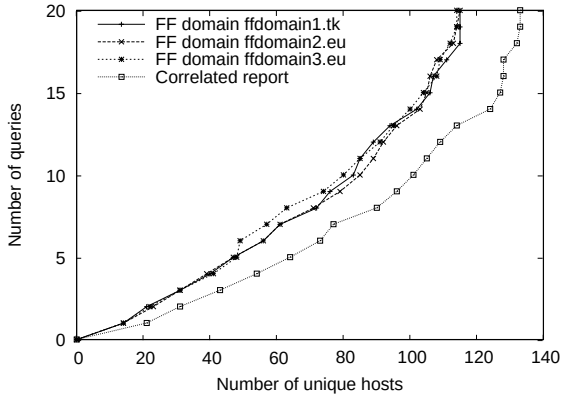
$$CPL_t = CPL_{t-1} \cup PL_1 \cup PL_2 \cup \dots \cup PL_m.$$

Hence, the number of unique IP addresses identified after t queries is $h_t = |CPL_t|$.

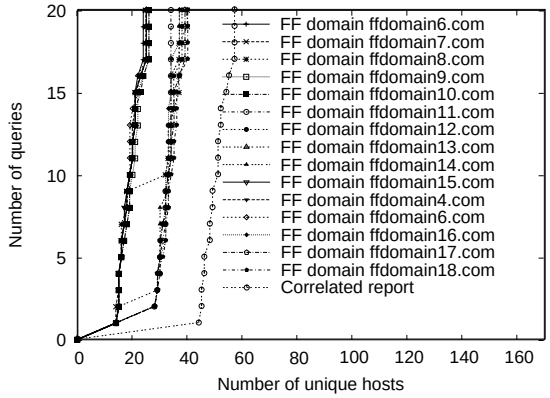
The key difference between this FF domain based correlation scheme and the DNS based correlation scheme is that this scheme correlates the IP address for multiple different FF domains from a single DNS server, while the DNS based correlation scheme correlates the IP address for a single FF domain from multiple DNS servers.

In order to evaluate the effect of correlating evidence from multiple FF domains, we plot the correlation results of DNS queries of three actual FF domains collected on PlanetLab in Figure 8(a), and the correlation results of DNS queries of seventeen FF domains collected by [10] in Figure 8(b). Both figures only plot the first 20 queries, *i.e.*, the discovery phase of the FF proxy deployment.

Figure 8(a) plots both the individual results and correlation results for three different FF domains in terms of the number of queries required to identify a given number of unique IP addresses. As shown in Figure 8(a), the individual FF domains have similar detection performance. In contrast, the results from the correlation scheme consistently outperform the results from the individual FF domains, with approximately 30% fewer queries needed to identify the same number of hosts. Figure 8(b) plots the correlation results of seventeen different FF domains. In this case, the results from the correlated scheme can detect far more unique IP addresses than any single domain in isolation. Furthermore, any new domain that uses an IP address from the correlation results can immediately be treated as suspicious.



(a) Correlation results on PlanetLab



(b) Correlation results of data collected by [10]

Figure 8. Effect of DNS query correlation from different FF domains

V. Scalable Platform for Correlation of Evidence

Having proposed two correlation approaches to detect FF domains, we now consider how these approaches can be implemented in practice. Intuitively, an easy approach to collaborative detection is to use a centralized server to correlate all information. In this approach, each IDS monitors queries from its local DNS, then the query results are reported to a central server, which correlates all the reported query results. However, this centralized approach can introduce a central point of failure and poor scalability.

Therefore, rather than relying on a centralized correlation platform, we need to support distributed correlation in a scalable manner with little management overhead. Previously, we have developed a general platform for collaborative correlation, called LarSID [27]. In this section, we describe how the correlation problems in Section IV can be mapped onto the LarSID architecture.

A. Correlation Problems

We are addressing two correlation problems in this paper: (1) correlation from multiple DNSs, and (2) correlation from multiple suspect FF domains.

In the first correlation problem, each participating IDS queries its local DNS for suspect FF domains $\mathcal{F} = \{f_k | k = 1, 2, \dots, r\}$ simultaneously. After each query,

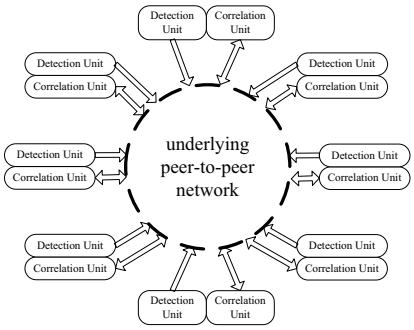


Figure 9. LarSID Architecture (*Detection Unit* and *Correlation Unit* are two separate processes in the participating IDS)

each participant shares their query results in the form of $\langle f_k, PL_i \rangle$, where PL_i is a list of IP addresses that are associated with f_k detected by participant IDS d_i, *i.e.*,

$$PL_i = \{s_{ij} \in IP | j = 1, 2, \dots, n_i\}.$$

The suspect FF domain name f_k is used as a key for correlation. All corresponding reported lists of IP addresses are aggregated to produce a global set of evidence statistics for the suspect domain.

In the second correlation problem, the source of data is the same as in the first correlation problem, *i.e.*, all the DNS query results reported by the participating IDSs. In this case, these results are shared in the form of $\langle s_k, FL_i \rangle$, where s_k is a suspicious IP address, *i.e.*,

$$\mathcal{S} = \{s_k \in IP | i = 1, 2, \dots, r\},$$

and FL_i is a list of suspect FF domains that are associated with s_k detected by participant IDS d_i, *i.e.*,

$$FL_i = \{f_{ij} | j = 1, 2, \dots, n_i\}.$$

The suspicious IP address s_k is used as a key for correlation. All corresponding reported lists of FF domains are aggregated to produce global evidence for s_k.

B. Distributed Correlation Algorithms

In order to support the distributed correlation, we use LarSID, a collaborative intrusion detection architecture proposed in our previous work [27], as shown in Figure 9. In LarSID, each participant IDS has two functional units, a *detection unit* that is responsible for collecting alerts locally; and a *correlation unit* that is a part of the distributed correlate-and-filter scheme. LarSID comprises a set of IDSs $\mathcal{D} = \{d_i | i = 1, 2, \dots, m\}$. Each detection system d_i queries its local DNS for suspect FF domains $\mathcal{F}$, in order to gather evidence of potentially suspicious activities. All of these intrusion detection systems are connected by the LarSID architecture and share information with each other. Periodically, each detection system shares evidence about either the suspect FF domains (first correlation problem) or the suspicious IP addresses (second correlation problem) seen on its local DNS.

In order to achieve collaboration between the participant IDSs, LarSID uses a peer-to-peer publish-subscribe mechanism for sharing evidence. Each participant d_i shares the contents of its query results via a publish-subscribe process, which takes place periodically after each query with a time interval Δ. We explain how each correlation problem (multiple DNS servers or suspected FF domains) can be mapped onto LarSID as follows.

- In the first correlation problem, at the end of each period Δ, each detection system $d_i \in \mathcal{D}$ subscribes to information about suspect FF domain $\langle f_k, PL_i \rangle$ detected from the local DNS. If this suspect domain f_k is confirmed as a FF domain by LarSID after correlating all related PLs, then d_i is notified about this domain f_k, as well as related information such as the number of unique IP addresses associated with it. The detection system d_i can then take further action based on the information received about the globally confirmed FF domain.
- In the second correlation problem, at the end of each period Δ, each detection system $d_i \in \mathcal{D}$ subscribes to information about suspicious IP address $\langle s_k, PL_i \rangle$ detected from the local DNS. If this suspicious IP address s_k is confirmed as a FF front-end proxy by LarSID after correlating all related FLs, then d_i is notified about this IP address s_k, as well as related information such as the number of FF domains associated with it. The detection system d_i can then take further action.

C. Performance Evaluation

Note that decentralized systems introduce additional communication overhead due to distributed message routing in comparison to a centralized approach. It is essential to quantify whether this increase in communication overhead outweighs any savings in computational load.

In order to measure the performance of the proposed distributed correlation schemes in LarSID, we have previously implemented a simple correlation scheme on LarSID [27]. In this implementation, we correlated suspicious evidence based on the *source IP address* only. This is representative of the proposed correlation scheme. We measured the time required for subscription and information correlation in comparison to a centralized collaborative intrusion detection system (CIDS), based on a deployment of LarSID on PlanetLab.

We measured the performance of our decentralized architecture in comparison to a centralized CIDS, which uses the same detection mechanism as the decentralized architecture. The experimental results of the performance of LarSID with the source IP based correlation scheme were reported in [27]. The traffic used was from a different attack scenarios, *i.e.*, the scanning behavior of sources during worm outbreaks. However, the underlying correlation task is the same as the proposed algorithm, and the results are indicative of the performance that can be expected for large-scale source address correlation. To summarize the results from [27], we found that the average subscription delay on the decentralized version of

LarSID was 16 to 10,000 times faster than the corresponding delay on the centralized LarSID, where the speed-up increased as the number of participants in LarSID increased. This is because when more participants join the collaborative framework, the increase in queuing delay on the centralized server is far higher than the increase in the routing delay on the decentralized CIDS.

VI. Related Work

Our study is related to two main areas of research: anti-phishing and collaborative intrusion detection. In this section, we provide a brief review of the relevant approaches and techniques in the literature.

A. Anti-phishing Approaches

Miyamoto *et al.* [17] proposed a filtering algorithm to defeat phishing attacks. They protect novice users from web phishing attacks by removing part of the content that traps users into entering their personal information. This approach is client-based webpage filtering, while our approach is phishing website trace-back.

Liu *et al.* [13] proposed an approach for server operators to automatically detect phishing webpages based on visual similarity. This approach focuses on detecting fraudulent sites by comparing the webpages, while our approach is to identify the phishing website hosts by correlating suspicious traffic patterns.

Due to their limited lifetime, phishing websites attempt to gain the trust of their users and convince them to act quickly. Drake *et al.* [5] discussed numerous tricks employed by phishing email scammers, in order to understand the psychological processes behind phishing attacks. There are a variety of anti-phishing tools that have been proposed to prevent users from disclosing their personal information, such as Cloudmark Anti-Fraud Toolbar [3], eBay Toolbar [6] and SpoofGuard [2]. However, based on a recent study conducted by Zhang *et al.* [26], the best tool among 10 popular anti-phishing tools tested can achieve more than 90% detection rate, but with 42% false positives. Furthermore, many users tend to ignore any warnings provided by anti-phishing tools [21], [24].

McGrath *et al.* [15] examined the operational aspects of phishing. ICANN [11] published an advisory which indicated FF is increasingly used to host phishing attacks. Moore *et al.* [18] examined the impact of phishing website take-down, and found the average life time of phishing domains was extended by using FF hosting. Holz *et al.* [9] proposed a similar work on FF network detection and measurement. They developed a metric for detecting FF domains based on three parameters that are similar to our detection characteristics. However, their approach and evaluation are based on a single point of observation. In contrast, our approach focuses on the collaboration between multiple observation points.

B. Collaborative Intrusion Detection

Locasto *et al.* [14] proposed a fully distributed CIDS based on a P2P architecture. Each participant uses an IDS to monitor its subnetwork or hosts. A tool called *Worminator* is run on the participating hosts at specified intervals to parse the alert output of the local IDS into the form of a *watchlist* (a list of suspicious IP addresses). Next, the encoded *watchlists* are distributed over a decentralized P2P-style network among the participants.

The DOMINO project [19], [25] is a distributed CIDS that aims to monitor Internet-scale outbreaks. The nodes in DOMINO are connected using a P2P protocol, and they participate in a periodic exchange of intrusion information. However, the DOMINO system has not been evaluated in a large-scale deployment.

Dash *et al.* [4] proposed a collaborative system of host-based IDSs, which use distributed probabilistic inference to detect slow network intrusions. A gossip protocol is used to communicate state between detection systems. A global view of the current security status of the monitored system is generated by analyzing local state information using a probabilistic detector model.

VII. Conclusion

In conclusion, FF domains are extremely difficult to detect in a timely and accurately manner, due to the use of a screen of proxies to shield the FF mothership. We present a theoretical model to analyze the FF detection problem by quantifying the number of DNS queries needed to retrieve a certain number of unique IP addresses. Our analytical model identifies an upper bound on the expected time required to detect a sufficient number of unique IP addresses to confirm a FF domain, and the theoretical speed-up for cooperation between multiple DNS severs. We then propose two approaches to correlating evidence from multiple DNS servers and from multiple suspect FF domains to speed-up FF domain detection. Our experimental results on real-world data show that a substantial speed-up in the number of queries needed for FF domain detection can be achieved. We finally discuss the implementation of our proposed correlation schemes in practice by using a decentralized correlation architecture called LarSID, as experimental results have shown that this decentralized correlation architecture is more scalable than a fully centralized architecture.

Acknowledgment

We thank the Laboratory of Dependable Distributed Systems in the University Mannheim, Germany for making their Fast Flux data available. This research was supported by the Australian Research Council.

References

[1] Anti-Phishing Working Group. [Online]. Available: http://www.antiphishing.org/

[2] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. Mitchell, "Client-side defense against web-based identity theft," in *Proceedings of Network and Distributed Systems Security (NDSS)*, 2004.

[3] Cloudmark Inc. [Online]. Available: http://www.cloudmark.com/desktop/download/

[4] D. Dash, B. Kveton, J. Agosta, E. Schooler, J. Chandrashekar, A. Bachrach, and A. Newman, "When gossip is good: Distributed probabilistic inference for detection of slow network intrusions," in *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI)*, 2006, pp. 1115–1122.

[5] C. Drake, J. Oliver, and E. Koontz, "Anatomy of a Phishing Email," *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[6] eBay Inc., "Using eBay Tool's Account Guard." [Online]. Available: http://pages.eBay.com/help/confidence/accountguard.html

[7] Gartner Inc., "Gartner Says Number of Phishing E-Mails Sent to U.S. Adults Nearly Doubles in Just Two Years, 2006 Press Release, 9 Nov 2006." [Online]. Available: http://www.gartner.com/it/page.jsp?id=498245

[8] Gartner Inc., "Gartner Survey Shows Phishing Attacks Escalated in 2007." [Online]. Available: http://www.gartner.com/it/page.jsp?id=565125

[9] T. Holz, C. Gorecki, K. Rieck, and F. Freiling, "Measuring and Detecting Fast-Flux Service Networks," in *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS08)*, 2008.

[10] T. Holz, "Fast Flux research data in the Laboratory of Dependable Distributed Systems in the University Mannheim, Germany," 2008. [Online]. Available: http://pi1.informatik.uni-mannheim.de/index.php?pagecontent=site/Research.menu/Fast-Flux.page

[11] ICANN Security and Stability Advisory Committee, "SAC advisory on fast flux hosting and DNS," January 2008. [Online]. Available: http://www.icann.org/committees/security/sac025.pdf

[12] M. Jakobsson and S. Myers, *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley-Interscience, 2006.

[13] W. Liu, G. Huang, X. Liu, M. Z, and X. Deng, "Detection of phishing webpages based on visual similarity," *International World Wide Web Conference*, pp. 1060–1061, 2005.

[14] M. Locasto, J. Parekh, A. Keromytis, and S. Stolfo, "Towards Collaborative Security and P2P Intrusion Detection," in *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security*, 2005, pp. 333–339.

[15] D. McGrath and M. Gupta, "Behind Phishing: An Examination of Phisher Modi Operandi," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[16] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms And Probabilistic Analysis*. Cambridge University Press, 2005.

[17] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi, "SPS: a simple filtering algorithm to thwart phishing attacks," in *Proceedings of Technologies for Advanced Heterogeneous Networks*, vol. 3837, 2005, pp. 195–209.

[18] T. Moore and R. Clayton, "Examining the impact of website take-down on phishing," *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*, pp. 1–13, 2007.

[19] V. Y. P. Barford, S. Jha, "Fusion and filtering in distributed intrusion detection systems," in *Proceedings of Annual Allerton Conference on Communication, Control and Computing*, September 2004.

[20] Princeton University, "PlanetLab Testbed." [Online]. Available: http://www.planet-lab.org

[21] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer, "The Emperor's New Security Indicators," in *SP'07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*, 2007, pp. 51–65.

[22] J. Stewart, "Danmec/Asprox SQL Injection Attack Tool Analysis," May 13, 2008. [Online]. Available: http://www.secureworks.com/research/threats/danmecasprox/

[23] The Honeynet Project & Research Alliance, "Know Your Enemy: Fast-Flux Service Networks," 2007. [Online]. Available: http://www.honeynet.org/papers/ff/

[24] M. Wu, R. Miller, and S. Garfinkel, "Do security toolbars actually prevent phishing attacks?" *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 601–610, 2006.

[25] V. Yegneswaran, P. Barford, and S. Jha, "Global Intrusion Detection in the DOMINO Overlay System," in *Proceedings of Network and Distributed Security Symposium (NDSS)*, 2004.

[26] Y. Zhang, S. Egelman, L. Cranor, and J. Hong, "Phinding Phish: Evaluating Anti-Phishing Tools," *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS 2007), San Diego, CA*, vol. 28, 2007.

[27] C. V. Zhou, S. Karunasekera, and C. Leckie, "Evaluation of a Decentralized Architecture for Large Scale Collaborative Intrusion Detection," in *Proceedings of the Tenth IFIP/IEEE International Symposium on Integrated Network Management (IM), Germany*, 2007, pp. 80–89.

[28] C. V. Zhou, C. Leckie, S. Karunasekera, and T. Peng, "A self-healing, self-protecting collaborative intrusion detection architecture to trace-back fast-flux phishing domains," in *Proceedings of the 2nd IEEE Workshop on Autonomic Communication and Network Management (ACNM 2008)*, April 2008.

Chenfeng Vincent Zhou received his Ph.D. in computer science from the University of Melbourne, Australia, in 2009. He is currently a Research Fellow at The University of Melbourne. He is a Certified Information Systems Security Professional (CISSP) since 2008. His research interests include computer security, network management and distributed systems.

Christopher Leckie is an Associate Professor in the Department of Computer Science and Software Engineering at the University of Melbourne, Australia. His research interests include using data mining and other artificial intelligence techniques for network intrusion detection and network management, as well as the management of sensor networks. Prior to joining the University of Melbourne, he was a Principal Engineer at Telstra Research Laboratories, where he conducted research and development into artificial intelligence techniques for various telecommunication applications.

Shanika Karunasekera received the B.Sc (Honours) degree in electronics and telecommunications engineering from the University of Moratuwa, Sri Lanka, in 1990 and the Ph.D. degree in electrical engineering from the University of Cambridge, UK, in 1995. From 1995 to 2002, she was a Software Engineer and a Distinguished Member of Technical Staff at Lucent Technologies, Bell Labs Innovations, USA. Since January 2003, she has been a Senior Lecturer at the Department of Computer Science and Software Engineering, University of Melbourne. Her current research interests are distributed computing, software engineering and peer-to-peer computing. Dr Karunasekera is a member of the ACM.