ABSTRACT

Title of Document: AN ANALYSIS OF NETWORK FLOW RECORDS FOR INFERRING WEB BROWSER REDIRECTION

Frank Shawn Hemingway
M.S. Electrical Engineering, 2013

Directed By: Dr. Michel Cukier
A. James Clark School of Engineering

Legitimate web browser redirection is often used to take users to web pages that have moved or to help users find the correct website when they have entered the web address incorrectly. Unfortunately, computer network attackers can use web browser redirection to manage malware-serving hosts and conceal their activity. An analysis of network flow records yields heuristics for flow size, flow duration, and inter-flow duration that indicate flows where web browser redirection is likely to have occurred. Results show that flows matching these redirection heuristics are indeed several times more likely to communicate with Internet hosts that have exhibited a history of malicious behavior. A network security administrator can thus filter large sets of network flow records to reveal flows most likely to contain web browser redirection. This capability reduces the sample space when looking for evidence of malicious activity targeting web browsers and contributes more generally to the expanding field of flow-based application recognition.

AN ANALYSIS OF NETWORK FLOW RECORDS FOR INFERRING WEB BROWSER REDIRECTION

By

Frank Shawn Hemingway

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2013

Advisory Committee:
Professor Michel Cukier, Chair
Professor T. Charles Clancy
Professor Tudor Dumitraş

Preface

I would like to step back to discuss some of my philosophical observations of the field of network intrusion detection in 2013. Some of these observations may be obvious—if not outright tautological—but I take a moment of pause nonetheless.

Intrusions happen because intruders intrude. This is *the* causal relationship, which should not be forgotten. Specific times and targets of computer attack can be influenced by external factors ranging from prominent geopolitical exigencies to novice school-aged hackers with too much time on their hands during the summer. Research methods and models may necessarily abstract away these uncontrollable exogenous factors, but we should not forget about them and then be surprised when our expectations and results aren't fully aligned. It is tantalizing for statisticians to analyze data, track trends, and even make predictions. Indeed, this thesis will introduce statistical distributions and models for inference—almost obligatory in the field of network intrusion detection. But like those of other intrusion detection efforts, these models are built upon prior events and assumptions that may not last long into the future. In short, such models work well…until they don't. Even with all external factors accounted for, intruder techniques and the underlying technology will change over time, and even the best prior models will need to be adapted or abandoned.

There's a paradox in detecting malicious activity. For example, the research in this thesis points to an association between web browser redirection and malicious activity. But in order to study these events and make this association, malicious

activity must first be recognized somehow. (Herein, we use a blacklist of IP addresses.) So, if we can already recognize the malicious activity enough to label it malicious or benign, then why don't we just use that technique for intrusion detection to begin with? After careful thought, the answer is scale and automation. Network security administrators want intrusion detection systems that will tune themselves, discover new attacks, and do it all without too many false positives. The work continues.

Information that's helpful to the defender can be helpful to the intruder. When we publish our work as computer security researchers, the information can be just as valuable to attackers as it is to defenders—if not more so. In cases of new, provable security solutions, this problem lessens, but such solutions may be further and fewer between. In the more frequent instances of intrusion discovery, where a new intrusion detection technique is developed and subsequently published, the intruders invariably take little time to adjust. In fact, computer security researchers often point out weaknesses and limitations in their own detection methodologies—even within the 'limitations' section of the original paper in which these discoveries are published. This knowledge of an intrusion detection method's limitations makes it easier for attackers to evade the method. In a similar way, it is not uncommon for system vulnerabilities to be disclosed before patches are available. Certainly attackers can and do misuse such information.

So the game of cat-and-mouse continues between defender and intruder, between publisher and the keyboard miscreant, both with fingers at the keyboard—

one typing works for publication, the other typing commands for intrusion. The works and the fingers continue. And both will press on.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

As computer systems remain vulnerable to attackers and malware, network security administrators and security-conscious users seek improved ways to detect and respond to threats. One such threat is malware that can be spread to users' computers through ordinary web browsing. Often, these browser-based incidents of malware infection involve web browser redirection. This thesis presents heuristics for inferring web browser redirection in network flow records and shows that such flows are more likely to be associated with malicious activity. The rest of this chapter will discuss the motivation for this research with respect to web-based malware, network flows, and redirects; it will discuss the contribution of this thesis; and it will discuss the outline of remaining chapters.

Motivation for Researching Web-based Malware

The motivation for researching the behavior of web-based malware includes that it is widespread and difficult to defend against. These negative aspects imply there is much need for research in detecting it, as a first step toward reducing its negative impact.

Web-based malware is so widespread that as much as 90% of malware is delivered via the web, according to a 2013 report by Palo Alto Networks [9][37]. The threat of web-based malware also persists across the continuum of attacker skills.

Novice attackers can set up malicious websites and lure victims through fraudulent email links; meanwhile, more sophisticated attackers may compromise legitimate websites to deliver malicious content.

Web-based malware is difficult to defend against because web browsing is the de facto way most users access the Internet. Furthermore, many Internet applications—ranging from email services and chat clients, to social networking applications and VPN clients—all employ web browser interfaces. This non-traditional use of many different application services on the same TCP port means that network administrators can no longer block application-specific ports as an effective way to block applications that have traditionally posed the greatest threat to security. To make matters worse, the development of complex web applications has encouraged more complex frameworks and program interactions on both the client-side and the server-side, opening the way for more software vulnerabilities to be exploited by attackers and further compounding the problem.

Motivation for Researching Network Flows

There are several motivations for researching network flow records. First, network flow records' storage and analysis requires much less space and processing power than would be required to store and analyze payload content. Second, since having network flow records does not require access to the underlying content, flow-based techniques can be used in environments where encryption is present. This benefit will be discussed in the context of previous work in Chapter 3. Finally, not

having access to the payload content allows network flow analysis to occur in environments that it otherwise wouldn't due to privacy concerns of users.

Motivation for Researching Redirects

Web browser redirection is a behavioral feature commonly associated with malware delivered by web browsing [19]. In some instances redirection techniques are even used by online criminal enterprises [28]. The reason for using one or more layers of redirection is that attackers or criminals receive benefits such as flexibility in directing victims to malicious websites and a means of avoiding detection [39] [38]. Studying the methodologies of attackers provides insight into how to best detect and prevent their actions. Ideally, a network security administrator would want to detect the malware, itself, but if this is not possible due to network-layer encryption or due to there being no known signature for the malware, then an alternative detection mechanism is to detect a closely linked event. In many cases, this closely linked event is a web browser redirection.

Contribution

This thesis describes a novel methodology for detecting web browser redirection in network flow records. The methodology builds on a foundation of work published by Hu, Knysz, and Shin [20], but this thesis provides more detailed investigation in some areas and a simpler approach in other areas. First, we validate the fundamental premise that web browser redirection is associated with malicious

activity. We conduct a novel experiment on the University of Maryland campus network with our own flow records to quantify the proportion of malicious activity associated with redirect flows. We label data as malicious using a simpler blacklist method that does not require DNS logs as is not limited to botnet detection as in [20]. We derive optimal threshold values from the heuristics and data presented in [20] but do not require sequential testing. The result is a faster determination of likely redirects with fewer computational resources by accepting the tradeoff of a higher false positive rate. We are more rigorous in specifying the directionality and delineation of flows that allows greater confidence in our experimental methods and in the reproducibility of our work. Finally, we thoroughly investigate parametric modeling of inter-flow duration time of web browser redirects. This novel analysis uncovers a better fitting probability distribution that more closely models the underlying phenomenon of web browser redirection.

Outline

The rest of this thesis is organized in the following way. Chapter 2 discusses background in NetFlow (the protocol used to collect and manage network flow records) and web browser redirection. Both topics are used extensively in the ensuing chapters. Chapter 3 discusses previous work. Chapter 4 presents the methodology for detecting web browser redirection in NetFlow. Chapter 5 presents the methodology for labeling flows as malicious or non-malicious. Chapter 6 explains

the data and results of our experiments. Chapter 7 provides an in-depth exploration of discovering the best parametric model for inter-flow duration of redirects.

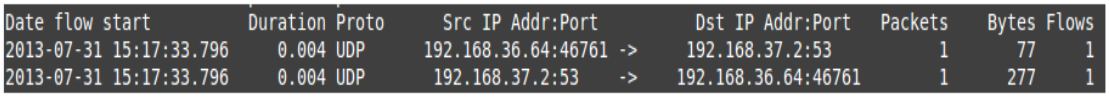
Chapter 2: Background and Definitions

The research in this thesis makes detailed use of network flow records (in NetFlow format) and the nuances of web browser redirection. This chapter explains these protocols and decomposes them into subcomponents to allow for analysis in ensuing chapters.

NetFlow

NetFlow is a Cisco-developed protocol for collecting and maintaining records about network connection events. Typically a network administrator enables NetFlow collection on the organization's gateway router, where a process monitors network connection activity. This NetFlow exporter sends the logs to another device, the NetFlow collector, which writes the logs to storage. The network administrator typically accesses the records through the NetFlow collector, which reads the logs back from storage. NetFlow records do not contain information that was actually transmitted; they only contain information about the connection, itself. Such information varies depending upon the version and settings of the particular flow record protocol in use [6][7]. Typical NetFlow fields include the following: date, time, duration, protocol, source IP address, destination IP address, source port, destination port, number of packets, number of flows, bytes per second, packets per second and TCP flags. The TCP flags field is the union of all TCP flags that were set

at any point in the connection and only applies if the connection was TCP. Fig. 1 shows a sample of two NetFlow records printed to the terminal screen in text format using the command-line tool nfdump.



Date flow start	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Packets	Bytes	Flows
2013-07-31 15:17:33.796	0.004	UDP	192.168.36.64:46761	->	192.168.37.2:53	1	77	1
2013-07-31 15:17:33.796	0.004	UDP	192.168.37.2:53	->	192.168.36.64:46761	1	277	1

Fig. 1. Two sample NetFlow records are printed to the terminal screen using nfdump.

NetFlow records are stored in a binary format to save space. Tools like nfdump can manipulate the binary files and print the contents to a terminal as shown above.

The definition of a flow is a unique 7-tuple composed of the following elements:

1) source IP address,
2) destination IP address,
3) source port,
4) destination port,
5) next protocol,
6) type-of-service field, and
7) inbound router interface.

The first five elements are commonly used for defining a socket in network programming or a conversation in a protocol analyzer. The last two elements deserve additional attention. The type-of-service field is deprecated and rarely used, but since it is one of the elements that define a flow, a change in its value will delineate the start of a new flow as far as the NetFlow exporter is concerned. Stated slightly

differently, the same source host traffic communicating with the same destination host over the same ports will be defined as a new flow each and every time one of the packets in that flow changes the value of the type-of-service field in the header of any IP datagram. The last element of the 7-tuple, the inbound interface of the NetFlow-exporting router is not part of the header of an IP datagram; instead, it is a value maintained by the capture process running on the exporter. This element adds ground truth to the directionality of flows, since it uses the hardware interface on the router and not a "soft" field in an IP header.

Redirect

The term "redirection" (or "redirect" for short) refers to an event where a user's web browser gets content from an initial server (Server 1) and then automatically accesses a second server (Server 2) based upon the content received from Server 1. Fig. 2 enumerates the four flows that occur in a typical redirection scenario. These flows are labeled as flow (a) through flow (d) and will be referred to in subsequent chapters. This thesis uses the term "redirect flow" to refer to a flow associated with a redirect.

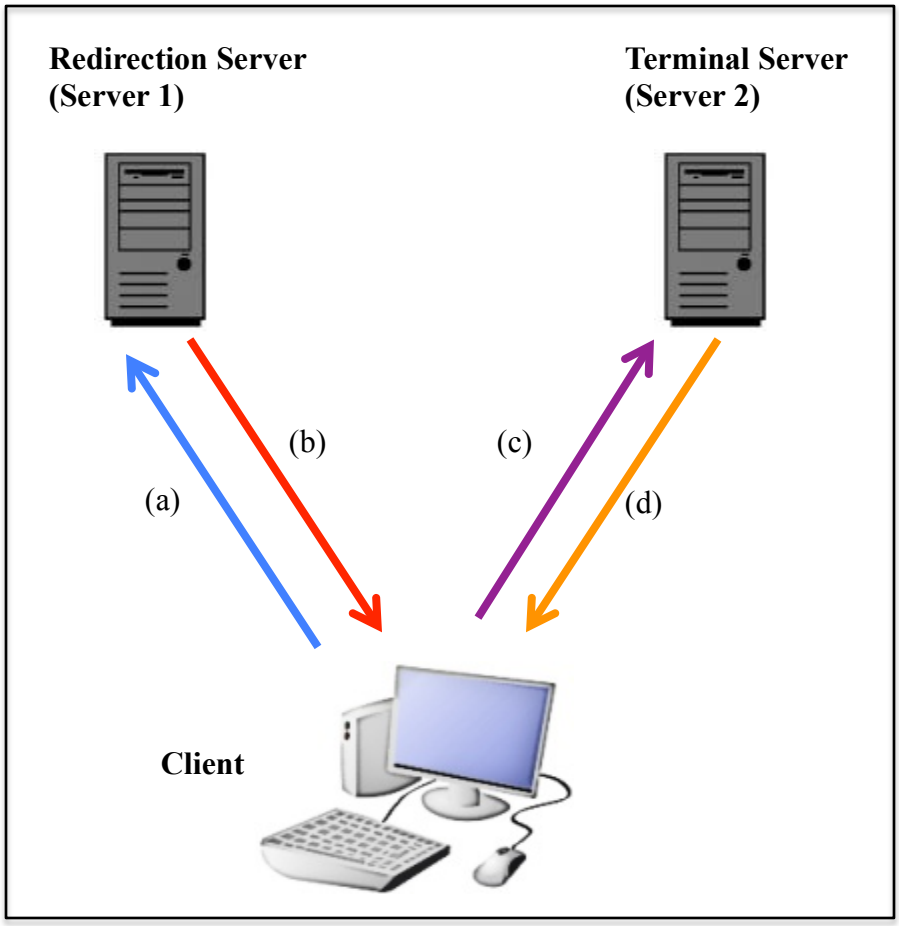


Fig. 2. The redirect sequence of four flows, labeled in order (a)-(d).

Unless otherwise specified, flows are unidirectional. That is, a flow contains all the packets from one host to another in a given direction; meanwhile, a separate flow contains the packets in the reverse direction. This definition is per the NetFlow and IPFIX specifications [6][7]. It is common for NetFlow analysis systems to assemble opposing flows between two hosts (i.e., where the source and destination IPs and ports are swapped), but it is useful to preserve uni-flow granularity—albeit while maintaining awareness that the opposite sides are bi-directionally related. We call the

aggregation of both associated flows a bi-directional flow or *bi-flow*. The packets contained in the four flows (a, b, c, d) and two bi-flows are described as follows:

a) Client to Server 1 (SYN, HTTP GET Request, … , [FIN, ACK]);

b) Server 1 to Client (SYN/ACK, HTTP 301/302 Redirect, [FIN, ACK]);

c) Client to Server 2 (SYN, HTTP GET Request, … , [FIN, ACK]); and

d) Server 2 to Client (SYN/ACK, HTTP 200 Response, …, [FIN, ACK]).

Flows (a) and (b) comprise the bi-flow between the Client and Server 1. Flows (c) and (d) comprise the bi-flow between the Client and Server 2. The packets associated with each flow are shown in parentheses, and the FIN/ACK segments are in brackets to indicate that they may or may not be included, depending upon the circumstances. Fig. 3 shows a detailed specification of an HTTP 301 web browser redirection with granularity of each packet in its corresponding flow. The two bidirectional flows (a-b) and (c-d) are drawn next to each other for compactness and to demonstrate their duality but do not represent their actual timing. In an implementation setting flows (c) and (d) will begin very shortly after flows (a) and (b) begin, and flows (a) and (b) may not terminate until much later—well after the redirect event has ended.

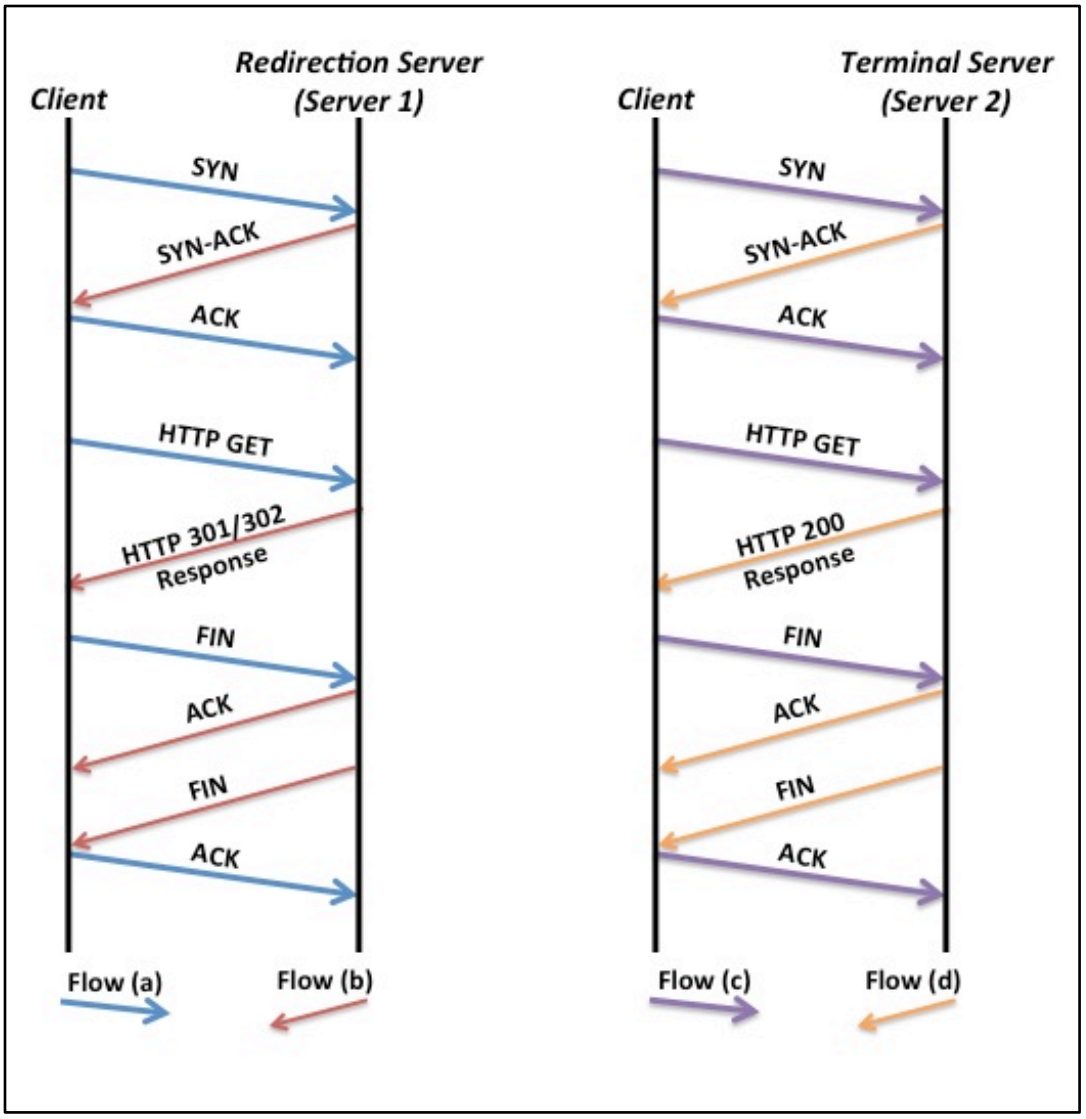


Fig. 3. Full redirect sequence of packet exchanges for all four flows and both bi-flows between the client and both servers.

A final note concerning terminology is the use of the term "normal" to distinguish non-redirect flows from redirect flows. When compared to redirection, the term "normal" refers to events, flows, URLs, or servers that are not associated with redirection. It a later context, when compared to malicious, "normal" refers to

events, flows, or URLs that are benign. The meaning should be clear from the context, but the terms “benign redirect” or “malicious non-redirect” will be specifically stated where necessary to avoid ambiguity.

Chapter 3: Related Work

The field of network intrusion detection is full of creative approaches for improving current and future capabilities. Since this thesis focuses on using NetFlow to detect web-based malware, we focus of some of the previous literature related to these topics. First we discuss how encryption of payload content has caused trouble for some intrusion detection methods, noting that NetFlow-based methods are not affected. Second, we discuss previous work that focuses on security concerns presented by web browser redirection. Third, we discuss work that uses NetFlow to detect botnets. Fourth, we take a closer look at the paper by Hu et al., explaining more of its intricacies. Last, we set the direction for the analysis and experimental methods of this thesis.

Intrusion Detection Difficulties with Encryption

Increased use of encryption in network traffic prevents traditional network sensors from evaluating payload contents. As a result, sensors relying solely on deep-packet inspection or payload signatures will fail to detect malicious activity. Since the use of encryption is likely to increase, much research focuses on the use of techniques to classify and analyze traffic without knowledge of packet contents. NetFlow is one possible solution (and is used in this thesis), but other possible solutions have warranted consideration as well.

To get around the problem that cipher text causes for IDSes, Joglekar et al. [22] introduce software libraries to be shared between the software that invokes the cryptographic protocol and a software monitoring solution. This approach includes anomaly-based and specification-based intrusion detection for some attacks and can be used in situations where it is appropriate to deploy such client stubs onto the hosts. Goh et al. [15] likewise introduce an approach to provide plaintext to a monitoring system, while maintaining cipher text between endpoints. Their solution is a protocol that provides the network-based IDS its own un-enciphered copy of the content with use of a VPN.

Rather than attempting a workaround to inspect plaintext from otherwise encrypted traffic as described above, Foroushani et al. [14] analyze traffic externals such as size and timing and put this data into an adaptive state matrix. The authors infer underlying activity such as scanning and HTTP GET requests due to the small size of these packets. On the other hand, they infer buffer overflows as having unusually large request sizes. They infer flooding by introducing a counter and recognition table. The challenge of this approach is a large false positive rate. Koch and Rodosek [26] propose a method for detecting malicious activity in encrypted command shells. The authors build a catalog of commands and responses that could be used by an attacker and statistically infer these command-response combinations by analyzing size and timing of the encrypted payload. This technique is naturally limited to instances where an attacker is interactively hacking into a system based

upon commands at a command prompt and does not account for more contemporary techniques such as malware intended for web browsers [9].

Redirection-focused Work

When it comes to delivering web-based malware, the use of web browser redirection is well documented. There is a catalog entry in MITRE Corporation's common weakness enumeration (CWE) database [11], and the special case of the "open redirect" makes the SANS Institute's top-25 list of most dangerous software errors [Lam]. Shue et al. discuss the prevelance and mitigation strategies of these redirects on both the client and server-side [39]. In spite of these efforts, the threat of web browser redirection remains widespread, and attackers are using it actively. According to a white paper published by the Sophos Internet security company, the use of this technique is increasingly common and has been seen in as much as 60% of web-based threats seen in commercial security research [19]. The threat is particularly difficult in situation where attackers are adept at hiding the redirection mechanism. Mavrommatis and Provos describe the use of hidden iFRAMEs in webpages that use web browser redirects to send unsuspecting users' browsers to malware-hosting sites [33].

Detecting Botnets with NetFlow

Flow data analysis is quickly becoming a popular method for detecting botnets. As botnet structures become more complicated in attempts to evade traffic

pattern analysis [40], the tools required to detect them need to be robust enough to handle these aberrations. Specifically, works by Iliofotou et al. [21] and Nagaraja et al. [35] have shown that the level of abstraction provided by flows allows a traffic graph to be built. Both of these works utilize ISP-level flow records to build traffic graphs and infer botnet activity based on the graph. Coskun et al. [10] use flow-derived graphs to detect botnet activity at the edge router of enterprise networks. These approaches detect decentralized communication graphs in different ways but all require the existence of at least one known-malicious host in the communication graph in order to label the overall communication structure as being affiliated with a botnet or being otherwise malicious.

Bilge et al. counter the trend of needing a priori insight into a botnet by developing NetFlow heuristics that do not require knowledge about the command and control of the botnet. Their method uses a supervised machine-learning algorithm that is trained on high quality data supplied by a security company. They introduce several features relating to flow size, access patterns, and time intervals—all very promising for detecting botnets when combined with a high-quality training set data [5].

Lee et al. [2] has demonstrated through their system Kopis, the ability to detect previously undiscovered malware domains with a low false positive rate, even without having access to the associated malware. Kopis works by analyzing DNS resolution patterns at the ISP level. Kopis is particularly powerful at identifying new botnets. Meanwhile, Leontiadis et al. [28] infer the architecture of cybercriminal

networks that use a popular redirection scheme, where certain nodes participating in unlicensed online pharmaceutical sales scams act as concentrators to redirect users to the drug sales servers. The redirection component in these schemes offers the perpetrator advantages, such as hiding their underlying infrastructure and allowing it to scale by serving multiple sites with one concentrator.

Combining Redirects, Botnets, and NetFlow

Hu, Knyz and Shin published a paper titled *RB-Seeker: Auto-detection of Redirection Botnets* in 2009 [20]. Their approach uniquely combines the aforementioned topics of redirects, botnets, and NetFlow. Much of the work in this thesis is built upon their prior work. Their approach is to seek out web browser redirection traffic from servers that are suspected to be hosted on botnet nodes and classify them as malicious based upon DNS behavior. Their solution is a three-part system. The first part, known as the Spam Source Subsystem (SSS), detects redirection domains by following URLs harvested from spam emails. The second subsystem, the NetFlow Analysis Subsystem (NAS), uses a statistical analysis technique known as sequential hypothesis testing (SHT) to determine if a sample seen in a NetFlow record is a redirection flow. The final subsystem uses the database from the previous two subsystems over time to probe the suspect domains and collect attributes, which a machine-learned decision function uses to categorize the domain as malicious or benign.

Direction of this Thesis

The work in this thesis begins with the notion of detecting web browser redirection in NetFlow as done in [20]. However, the ideas in this thesis decouple the threat of botnets from the threat of malicious web browser redirection. It is true that botnets may control infrastructure that host malicious redirection servers and malware servers, but such malicious servers can exist on standalone servers, without the presence of botnets. Malicious redirection servers can also exist in cases where an attacker has compromised a legitimate website.

We apply the pertinent methods in [20] for detecting web browser redirection and show that this work can be extended to a different campus network four years after the initial application of the ideas and maintain relevance despite the many changes in the Internet and in web standards. We validate the association of malicious activity with web browser redirection by using a different method than in [20], which adds rigor to the conclusion that there is an association between malicious activity and web browser redirection. The method we use is a blacklist of IP addresses researched and compiled by external security organizations. This method takes the subjectivity out of the determination of maliciousness (at least from our point of view) and simplifies our decision as a Boolean determination.

Using a blacklist of IP addresses is also extremely convenient for NetFlow analysis because IP addresses are available in NetFlow records, whereas domain names and URLs are not. The IP address is thus the most likely field of a NetFlow

record that could intrinsically indicate malicious activity. The reasoning is that there is nothing intrinsically malicious about other fields, such time stamps and byte counts from a single host, because benign traffic could be constructed that could result in almost any value for these fields. Using IP addresses is more robust because IP addresses must be registered with a central authority, so there is at least a starting point for validating the legitimacy of content hosted by a given server using a specific IP address. More importantly, if an IP address cannot be verified at all, then that is all the more reason for it to be suspicious. It is true that there are limitations to this method—not the least of which is that IP addresses may not be isolated to individual entities. These limitations are discussed more in depth at the end of Chapter 5.

Two fundamental questions are addressed in the next two chapters. First, can we infer that a web browser redirect is occurring by examining external size and timing information for a flow? This question applies to all redirections—both malicious and benign. This question is addressed in Chapter 4. Second, can we determine whether a given redirect is benign or malicious? This second question is addressed in Chapter 5.

Chapter 4: Detecting Web Browser Redirection in NetFlow

The first step of the two-step strategy is detecting that a redirect has occurred. (The second step is determining whether or not that redirect is malicious. This second step will be discussed in the next chapter.) The detection of a redirect in NetFlow is actually an inference because NetFlow records do not maintain application-layer content that would be necessary to verify the redirect with absolute certainty. But since the NetFlow records are intentionally filtered by destination port 80, it is reasonable to assume that the flows generally correspond to web browsing activity of some type. Under this assumption, three features are useful for further filtering the NetFlow records as being redirects. These three features are 1) short flow duration, 2) short inter-flow duration, and 3) small flow size. We then create a filter using these features and apply it to the network at the University of Maryland.

Short Flow Duration

Flow duration refers to the duration of flow (b). It can be a difficult indicator to apply because some servers do not terminate their TCP connections after redirecting clients through HTTP 301/302 redirects. Although short flow duration is a highly informative feature in theory—because a server does not *need* much time to transmit the redirect—this feature is somewhat limited in practice because there is no

guarantee that the server will close the TCP connection and thereby terminate the flow after the redirect is served [20].

TABLE 1 below shows values from the previous work by Hu et al., which has characterized features for classifying redirect vs. non-redirect flows. The top third of the table gives first- and second-order statistics for flow duration at a large university network.

TABLE 1 Previously published values for flow duration, inter-flow duration, and flow size on a large university campus network and broken out by redirection vs. normal activity

		Mean	Median	Std dev
Flow duration (ms)	Redirection	305.5	128.6	2159.2
	Normal	33042.3	10028.8	91912.5
Inter-flow duration (ms)	Redirection	392.7	154.4	872.4
	Normal	40132.9	1345.5	87281.0
Flow size (bytes)	Redirection	2401	629	44530
	Normal	51495	4852	192431

Data values reproduced from RB-Seeker. Hu, Knysz, and Shin. NDSS. 2009.

The mean value is just over 300 milliseconds for redirects and over 33,000 milliseconds for normal traffic. With two orders of magnitude separating the means, I choose 500 milliseconds as a reasonable static threshold for filtering out redirects from non-redirects. This value allows for a comfortable 200-millisecond cushion between the redirect mean value and the cut-off threshold.

Short Inter-flow Duration

Short inter-flow duration refers to the time between the start of flow (a) and the start of flow (c). It is the most important feature for inferring redirects—especially malicious redirects—because it is necessary for a redirect to be short to be less intrusive to the user. That being said, short inter-flow duration is not an adequate feature by itself for inferring redirects because web sites that load external content such as images (or even ads) from other sites will have a similar flow pattern of short inter-flow duration.

Fig. 4 from [20] shows the cumulative distribution functions for redirect flows and non-redirect flows on the University of Michigan network. Using this finding as a baseline measurement for choosing a static threshold to separate redirects from normal traffic, I choose a value of time where the separation between the CDFs is relatively high. Visual inspection results in the red dashed line, which is located at approximately 1200 milliseconds. There is reason to err on the side of a slightly longer threshold in order to include more redirects. The first reason is that redirects make up a relatively small portion of overall network traffic, so having a larger sample helps reduce the base rate fallacy error [3]. The second reason is that if an attacker does try to evade this technique by introducing a delay in the redirect, the delay is typically an integer-number of seconds. A delay of one second would be acceptable to pass through the 1200ms filter and still leave 200ms or network latency (a reasonable round-trip time for an Internet host).

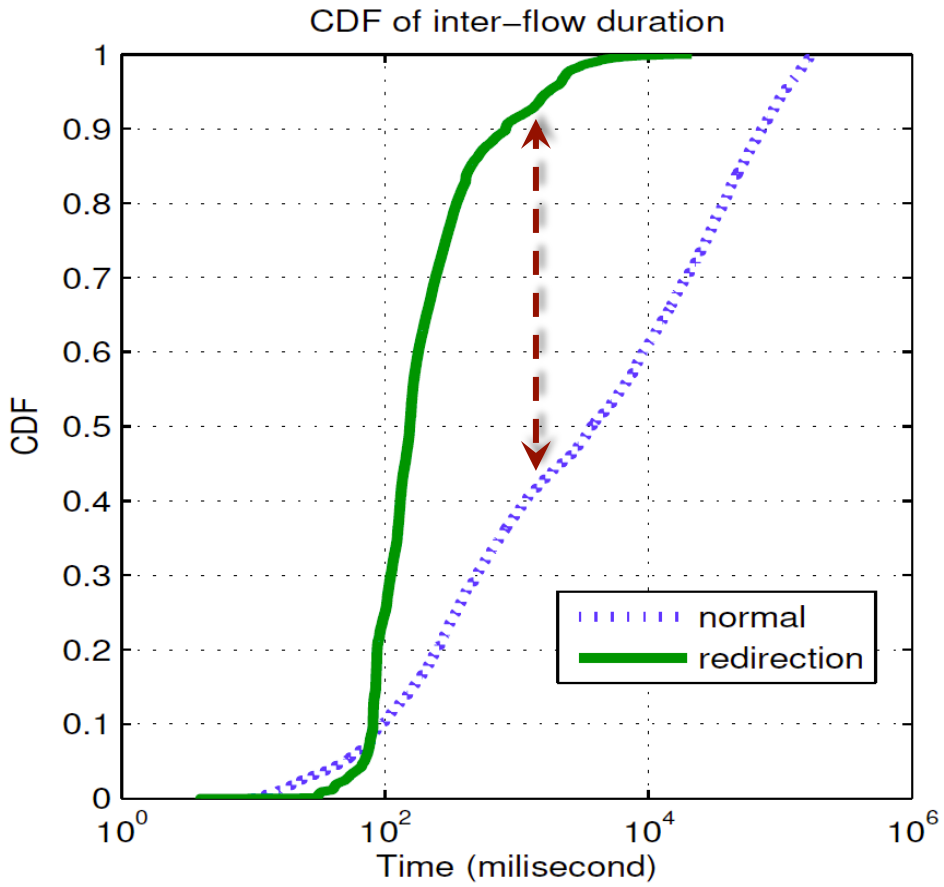


Fig. 4. The red vertical dashed line is overlaid at approximately 1200 milliseconds to indicate a relative maximum between the CDFs of the inter-flow duration times corresponding to the normal vs. redirection flows. (The original figure is courtesy of Hu, Knysz, and Shin. RB-Seeker. NDSS. 2009.)

Small Flow Size

Intuitively, the size of flow (b) is small in the case of redirects because all that is needed is a small HTTP 301/302 message. This small size should distinguish from a concurrent connection loading external content, which should be presumed to be much larger. Exceptions to this small size of flow (b) could be caused by javascript redirects or HTML meta-tag redirects that could be included with a large web page.

Referring back to TABLE 1, the mean value of redirect flows is just over 2400 bytes and over 51,000 for normal flows. This is still a large difference but not two orders of magnitude as for the other two features. I choose 2,500 bytes to allow a small cushion to the upper side of the mean in order to respect the smaller sized non-redirect flows that will inadvertently pass through this static threshold filter. Again, it is preferred to err on the side of a more generous filter for redirects to include more true positive samples and reduce the impact of the base-rate fallacy [3].

Applying the filters

The three static thresholds are applied to the campus NetFlow records. Intuitively, smaller values for these features are associated with redirects since a Server 1 redirect provides a small amount of information to the client in a short amount of time, after which the client usually proceeds immediately to Server 2. The thresholds are as follows:

1) Flow Duration <= 500 ms AND
2) Inter-flow duration <= 1200 ms AND
3) Flow size <= 2500 bytes.

These values are chosen to maximize the number of redirect flows and minimize the number of normal flows for each metric as previously indicated by the red arrows in Fig. 4. When combined, they empirically filter out an average of 0.15% of the University of Maryland campus network traffic as redirects. The methodology of this calculation is explained in Chapter 5. The expected percentages of the individual

thresholds are given in TABLE 3, but first we will describe the parametric models that provide those values. The models are built from the values in TABLE 2, which are reproduced from [20]. This work concludes that the probability distributions for the three features follow lognormal distributions. The reasoning of the authors is based upon a visual inspection of histograms of the data, where the values form a non-negative, heavy-tailed, bell-shaped curve. The maximum likelihood estimates from the previous work are shown below for the inter-flow duration and size features for redirects (R) and normal (N) traffic.

TABLE 2 Maximum likelihood estimates and confidence intervals for an assumed lognormal fit

	μ	95% C.I. of μ	σ	95% C.I. of σ
Inter-R	5.270	[5.260, 5.281]	0.974	[0.966, 0.9812]
Inter-N	7.982	[7.896, 8.067]	2.512	[2.454, 2.574]
Size-R	6.529	[6.517, 6.542]	0.956	[0.948, 0.965]
Size-N	8.423	[8.380, 8.466]	2.093	[2.063, 2.125]

Data values reproduced from RB-Seeker. Hu, Knysz, and Shin. NDSS. 2009.

The flow duration statistics are not included in the prior work. The flow duration is considered the least valuable feature due to the problem of servers not closing out sessions after the redirect is served. This thesis validates and quantifies that claim by reconstructing the distributions from the second-order statistics above and comparing the values of the respective cumulative distribution functions at the filter threshold values. The results are depicted below in TABLE 3.

TABLE 3 Percentages of traffic captured based on thresholds

Estimated Percentage of Traffic Captured

Thresholds		Redirection	Non-Redirect
Flow duration	500 (ms)	53.59%***	36.16%***
Inter-flow Duration	1200 (ms)	96.92%	36.13%
Flow size	2500 (bytes)	91.22%	38.74%

(*** Corresponds to a normal distribution, not lognormal)

Indeed, the flow duration feature provides the least differentiation between redirects and non-redirects, with only a 17-percentage-point spread. Meanwhile, the inter-flow duration and flow size parameters enjoy a much more comfortable 50- to 60-percentage-point spread. In particular, this difference is due to the redirection flow duration being relatively low (less than 54%) rather than to the non-redirect percentage being at a higher percentile. This result is relevant because the flow duration threshold value of 500 milliseconds is generous in allowing flows to pass through the redirect filter. (Recall, the mean value is just over 300 milliseconds.) The implication is that even with a more generous threshold, fewer flows passed through the redirect filter based upon flow duration than based upon the other two features. This observation is consistent with the theory that valid redirects are missed with respect to flow duration due to servers not closing out TCP connections after serving redirects. Due to this phenomenon, the authors in [20] do not include flow duration as a mandatory filtering criterion in their NetFlow analysis system for redirects. Accordingly, they do not include the maximum likelihood estimates for the

first- and second-order statistics for flow duration in their second table, providing estimates and confidence intervals only for inter-flow duration and flow size. Without having the original data set, we revert to the sample mean and standard deviation values in TABLE 1 and model them with a normal distribution as indicated by the double asterisk in TABLE 3.

Because this research focuses on web browser redirection, the set of campus NetFlow records from the gateway router are filtered to outbound flows with TCP port 80 to restrict the sample space to web browsing. Flows with less than three packets are also filtered out because this is the minimum number of packets from the client to establish a TCP session and request a web page. These filters are implemented directly from a standard NetFlow capture utility such as nfdump.

The more difficult step is filtering the flows based upon inter-flow duration because unlike flow size and flow duration, inter-low duration is not directly stored in NetFlow records nor is it computed by standard NetFlow command line tools. A custom analysis script is used to analyze the output from an nfdump log to identify redirects or normal traffic based on flow duration, flow size and inter-flow duration. Flow duration and flow size are straightforward parameters by which to filter the campus NetFlow records since they are standard NetFlow features and can be filtered by nfdump. The inter-flow duration is not maintained in standard NetFlow records, so it is computed in post-processing. The flow records are grouped by source IP address of the client and sorted by time. Two consecutive flows initiated by the same

host IP address within the inter-flow duration threshold value (1,200 ms) are considered to be a redirect with respect to this feature.

Chapter 5: Labeling Malicious Flows

The second step of the two-step strategy is determining whether or not a given flow is malicious. (The first step is determining whether or not a flow is a redirect as discussed in the previous chapter.) We are primarily concerned with the maliciousness of redirect flows, but it is necessary to evaluate the maliciousness of all flows (redirect or not) when building performance metrics. Our motivation is such that if we validate the assumption that redirect flows have a higher probability of being malicious, then it may be helpful for the security community or an overwhelmed system administrator to focus more attention on this subset of activity.

The method used in this research is to label flows malicious if the remote IP address of a flow matches an IP address on a blacklist aggregated from several well-known IP address reputation services [12][18][41]. The set of malicious IP addresses B is the union of the IP addresses in each of the publicly available blacklists B_1, B_2, B_3.

$$B = B_1 \cup B_2 \cup B_3$$

This method reduces the analysis to a simple Boolean determination, which aids simplicity. Either the IP address being considered is an element of B or it is not. This emphasis on simplicity is intentional. Since we are researching the association of redirects with malicious activity we do not want to confound the experiment by introducing uncertainty in the way maliciousness is determined. For example,

maliciousness could be determined by whether or not the corporate intrusion detection firewall alerted the activity, but such an IDS is, itself, imperfect and creates many false-positive alerts. In fact, most commercial IDSes have a variable setting used to turn down the sensitivity of the system to a rate of false positives that is manageable for the network administrator. In fact, from a philosophical perspective it could be argued that the mechanism for determining maliciousness is necessarily imperfect. If it were not, we would simply have used that mechanism for security monitoring to begin with, and there would not be a need for intrusion detection research. A recently published work states this conclusion best by saying, "[G]iven the relative immaturity of the cyber research domain, there is significant value and importance in the simplest approaches" [24].

Other authors who have studied redirects in network flows use a more complicated mechanism for determining maliciousness. They study the relationships among IP addresses, domain names, and autonomous systems to separate out malicious activity, and their approach is particularly tailored to the activities of botnets [20]. The blacklist method used in this thesis confirms the association between redirects and malicious web browsing without requiring follow-on analysis of DNS records and is not necessarily limited to botnets. The advantages of the blacklist method are thus simplicity, generality, better privacy, and less data storage. The black list method is simpler because the determination is either true or false (match or no match). This clarity is preferred because it reduces the amount of inference, improves reproducibility, and reduces possibilities for error. The

determination is also more general in that it is not specific to botnet behavior. A malicious site can be added to the blacklist because it is serving malicious web pages irrespective of whether or not that side is controlled through botnet infrastructure. Finally, the blacklist method allows for more privacy for users and requires less storage because it does not require the use of large DNS logs that contain lists of sites visited by users. These last two issues are legitimate constraints that can limit research using DNS logs for research or security purposes at large organizations concerned about user privacy.

Complications and Limitations

The limitations of using a blacklist for determining maliciousness are relying on an outside source to make the determination. There is no guarantee how often a site reputation service will update its list or how accurate the determination will be. There is still potential for false positives as well as false negatives. False positives would most likely occur due to outdated information if, for example, a legitimate webserver had been compromised to serve exploits but had subsequently been cleaned. Incorporating multiple blacklist sources into a single master blacklist helps combat false negatives. There is greater likelihood that at least one source will correctly mark a deserving site as malicious.

The biggest limitation of using IP addresses as the feature for a blacklist is that publicly routable IP addresses can be shared by many hosts through dynamic host configuration protocol (DHCP) and network address translation (NAT), so there is no

guarantee that a given IP address will always correspond to the same host. Also, many websites can be hosted by the same infrastructure and share the same IP address, so it is possible that a malicious website could blend in with legitimate sites or that one malicious website could generate false positive indications for legitimate sites hosted on the infrastructure.

Increasingly common client-side code can also create difficulties in detecting malicious redirects. For example, if javascript or HTML meta-tags are used to redirect the client, then the attacker could also include a delay to throw off the short inter-flow duration parameter. But such a delay would slow the loading of the final page, making the overall attack less discreet to the user.

Even without purposeful evasion by an attacker, flow size, flow duration, and inter-flow duration can vary from one network to another due to various factors such as geography, available bandwidth, and congestion. Using these parameters in a different campus network explores how they may need to be adapted to varying network environments and adjusted over time as technology and attacker techniques change.

Chapter 6: Data and Results

This chapter presents the experiment and the empirical data that was collected from the experiment in order to show that web browser redirects occur in higher proportion in malicious traffic than in normal traffic. We present the data and the method of analysis that we use to reach our conclusion. At the end of the chapter we discuss some challenges and limitations of the experiment.

Data

The experiment considers full-take (i.e. un-sampled) NetFlow records captured in 20-minute durations twice a day for 12 consecutive days on the university's gateway router. The first sample was taken beginning at 1 pm, and the second sample was taken beginning at 11 pm on each of the 12 days. The reason for choosing these two times was to account for any difference in data between business hours and non-business hours at the university. Each 20-minute sample contained, on average, 36 million total web flows and 55,000 redirect flows (i.e. flows that matched the three-threshold filter). On average, 9,000 flows were associated with malicious activity (i.e. had destination IP addresses matching the blacklist), which is approximately 0.025% of the total traffic.

Analysis

The goal of our first phase of research is to confirm whether or not redirects are still a relevant indicator of malicious activity (e.g. drive-by download attacks). To make this determination we measure the ratio of malicious flows out of redirect flows and compare that value to the ratio of malicious flows out of all flows. We validate this condition by applying the inequality

$$(B_r / R) > (B / A),$$

where the variables are defined as follows:

B_r is the number of malicious flows out of those marked as redirects by our filter (for the given time sample);

R is the total number of redirects (as marked by our filter for the given time sample);

B (bad) is the total number of malicious flows, defined as those going to external IP addresses on the master blacklist; and

A (all) is total number of flows passing through the gateway NetFlow router during the given time interval.

If the inequality is true, then there is a higher proportion of malicious flows in redirects than there is in the general population of flows. To evaluate the relationship above, we derived a performance factor, which is the ratio of the fraction of malicious traffic in the redirect subset to fraction of malicious traffic in the overall subset.

$$\text{Performance Factor} = (B_r / R) / (B / A)$$

If the performance factor is greater than 1, then the assumption that redirect traffic is more likely to be malicious holds.

Results

Fig. 2 is a chart of the base-10 logarithm of the performance factors for each of the 24 20-minute NetFlow samples taken over 12 days. Positive values for a sample indicate a higher concentration of malicious activity in redirect flows than in non-redirect flows for that sample. Negative values indicate the reverse.

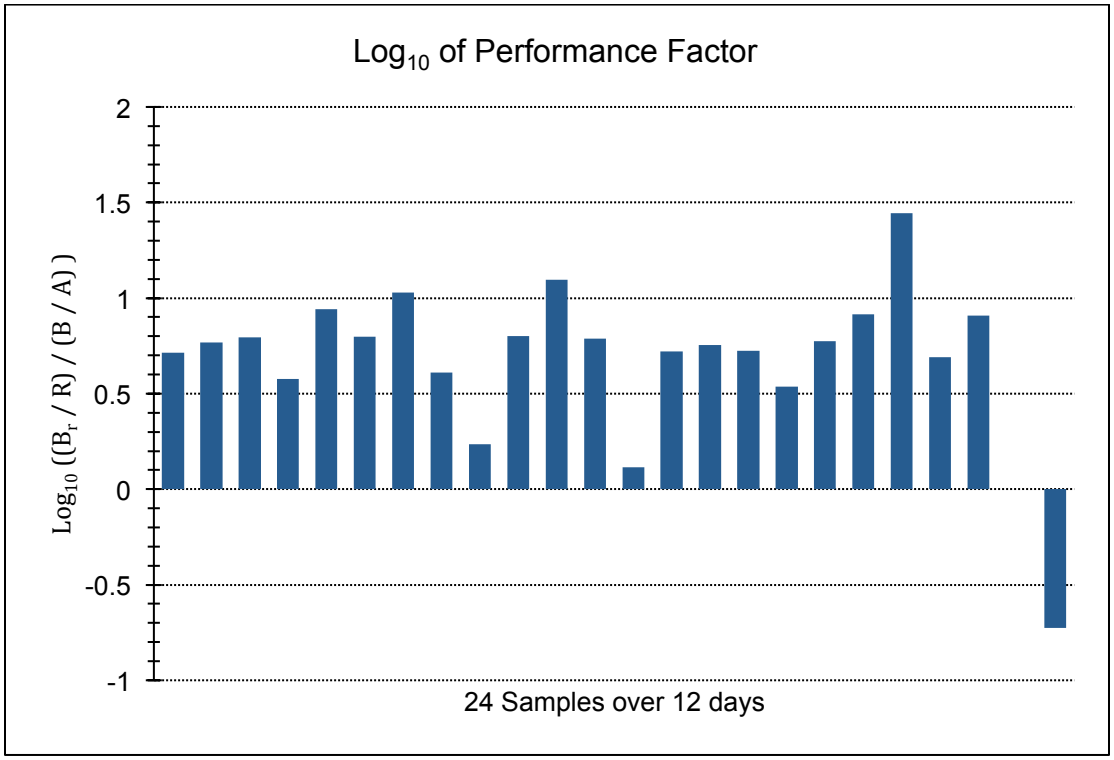


Fig. 5. The log of the performance factor of each sample illustrating the association of redirects with malicious activity.

As shown in TABLE 4 there were, on average, 6.9 times as many redirect flows communicating with IP addresses on the blacklist than there were non-redirect flows communicating with those same IP addresses over the 12-day interval. This

result confirms the belief that redirect flows are associated with malicious activity (assuming the three-threshold redirect filter is working as intended).

TABLE 4 B_r, R, B, and performance factor values over all 24 samples

Date/Time	Total Malicious Redirects	Total Redirects	Total Malicious Flows	Performance Factor
2012-12-03-1300	196	145161	11147	5.1838
2012-12-03-2300	130	87089	8890	5.8397
2012-12-04-1300	236	85017	19439	6.2383
2012-12-04-2300	37	47822	6945	3.7846
2012-12-05-1300	162	50971	16143	8.7356
2012-12-05-2300	140	111621	6870	6.2897
2012-12-06-1300	136	48608	11832	10.7016
2012-12-06-2300	19	22983	7153	4.0898
2012-12-07-1300	26	72298	8897	1.7219
2012-12-07-2300	17	16189	4630	6.3053
2012-12-08-1300	43	19249	5552	12.4815
2012-12-08-2300	22	18401	5459	6.1397
2012-12-09-1300	6	21696	6370	1.2994
2012-12-09-2300	122	86643	9342	5.2550
2012-12-10-1300	119	66388	14016	5.6742
2012-12-10-2300	48	54097	5970	5.2896
2012-12-11-1300	18	19045	11786	3.4409
2012-12-11-2300	61	46710	7388	5.9650
2012-12-12-1300	153	81044	9617	8.1918
2012-12-12-2300	415	73713	7257	27.8010
2012-12-13-1300	95	82925	9441	4.9148
2012-12-13-2300	85	46919	7444	8.0712
2012-12-14-1300	0	8666	5742	0
2012-12-14-2300	1	24195	6768	0.1874
Overall Performance Factor				6.9378

Only two samples do not have performance factors greater than 1. Of these two, the first has no flows browsing to a blacklist, so the performance factor is zero. The second has only one flow browsing to an IP on the blacklist. These are the two smallest instances of malicious flows out of redirect flows, so this outlier behavior is most likely the reason for performance factors less than 1. The next smallest value in any 20-minute sample is six, which was large enough to provide adequate sampling in the experiment to result in a performance factor greater than 1 (1.3 to be exact).

Visual inspection of the data does not indicate significant differences or any observable trends between the 1pm and the 11pm samples in terms of the performance factor or the absolute numbers of malicious or redirect flows. So there is no reason to conclude that there is more or less malicious activity occurring at a specific time of the day.

Challenges and Limitations

One of the challenges of conducting a research experiment involving an entire cross-section of flows at the gateway router of a relatively large campus network is managing the data and the infrastructure needed to support calculations on such relatively large data sets. A sampling interval is often used when collecting NetFlow records so that a flow will be collected for only one out of every N packets, where N is a configurable value that may or may not have a random component [43]. This research used full-take (un-sampled) NetFlow in order to maintain the rigor of a complete sample set and so as not to bias the sample. The average size of a

compressed five-minute NetFlow file used in this research is 134.3 MB, with sizes during peak hours exceeding 400 MB. Each file averaged over 13,248 flows. Even with filtering to remove network traffic not on port 80 and with preprocessing to remove traffic that was not relevant, uncompressed file sizes for the 20-minute exceed 2 gigabytes. The compressed files pertaining to the experiment used 1.7 TB of network storage. The impact of the relatively large data set is limited agility in manipulating the data. Filtering and labeling the data required hours of computation time to compute data points, which limits exploratory and ad hoc manipulation of the data.

Chapter 7: Parametric Probability Distribution Fitting of the Inter-flow Duration Time for Web Browser Redirects

Prior work has modeled the probability distribution function (PDF) of the inter-flow duration time of web browser redirects in order to conduct sequential hypothesis testing of redirection for individual servers [20]. This prior work found the lognormal parametric model to provide the best fit out the three models they tested. This chapter explores the fits of other parametric models and concludes that a generalized extreme value model provides the best fit and best representation of the underlying phenomenon for the over 4,000 timing samples taken for interflow duration at the University of Maryland in 2013. The methods considered for determining the best distribution are maximum likelihood estimation, the Pearson method of moments, the Johnson method of quantiles, and the shape properties of each distribution. We also include Kolmogorov-Smirnoff statistics and chi-square goodness-of-fit statistics, which further support the conclusion that the generalized extreme value function provides the best fit for modeling inter-flow duration time for web browser redirects. In this chapter we use MATLAB for much of the analysis and refer to equations from the software documentation [30][31][32].

Maximum Likelihood Estimation

Given the 4,424 data points corresponding to inter-flow duration samples, we wish to model a PDF $f_X(x)$ for this empirical data using an appropriate parametric

model. Specifically, given the data samples in a vector **x**, we wish to compute the values for θ that maximize the likelihood function $L(\theta)$, where θ is a vector of the parameters of the parametric distribution being tested. The step of multiplying the probabilities assumes that the random trials are independent.

$$L(\theta) = \prod_{x \in X} f(\theta \mid x)$$

We then take the log of the likelihood function so that the product of the probabilities does not approach zero as the number of data point becomes large. The parameter values of θ corresponding to the maximum of the log-likelihood function also correspond to maximum of the likelihood function. This test is computed numerically for fourteen possible candidates of distribution functions. TABLE 5 shows the log-likelihood values corresponding to each of the candidate distributions. The maximum log-likelihood value has the smallest absolute value because log-likelihood values are always non-positive. This smallest magnitude of the negative values corresponds to the distribution with the best fit for this data (under the under the maximum likelihood test).

The probability distribution with the best fit is the generalized extreme value distribution. Interestingly, the probability distribution above with the worst fit is the extreme value distribution in its non-generalized form. The next two sections will discuss these distributions specifically.

TABLE 5 Log-Likelihood Values of each distribution in order of best fit

Distribution	Log-Likelihood Value
Generalized Extreme Value	-7.8988e+03
Log-logistic	-7.9222e+03
Gamma	-8.3777e+03
Logistic	-8.6201e+03
Weibull	-8.7264e+03
Lognormal	-8.7660e+03
Rayleigh	-8.9333e+03
Rician	-8.9333e+03
Generalized Pareto	-9.5885e+03
Exponential	-9.6720e+03
Normal	-9.7444e+03
Birnbaum-Saunders	-1.1331e+04
Inverse Gaussian	-1.1933e+04
Extreme Value	-1.2912e+04

Previous research examined the parametric fits of lognormal, Weibull, and Pareto distributions for inter-flow duration times due to these distributions' qualitative features of having non-negative values and long tails for positive values. These qualities are intuitive because inter-flow duration times cannot be negative and have relatively rare (but existent) high values that occur when the redirect is delayed. We examine how well PDFs of these parametric models fit with respect to the University of Maryland data. As can be seen in Fig. 6, the lognormal and Weibull distributions can be shifted and scaled such that their bell shapes roughly match the bulk of the empirical inter-flow duration samples. However, a close look reveals that there is considerable improvement for having a tighter fit, particularly at the smaller

time values where the lognormal fit and the Weibull fit flare outside the data. (Note: The Pareto distribution was also considered in [20], but that PDF is omitted because it's fit to our data is extremely poor and detracts aesthetically from the resulting chart.) In addition, these two distributions do not rise as high at the peak of the empirical data, nor are they as well centered with the rise in the empirical data, as is the generalized-extreme-value-fitted curve. Let us then examine the generalized extreme value distribution and its special subcase, the extreme value distribution.

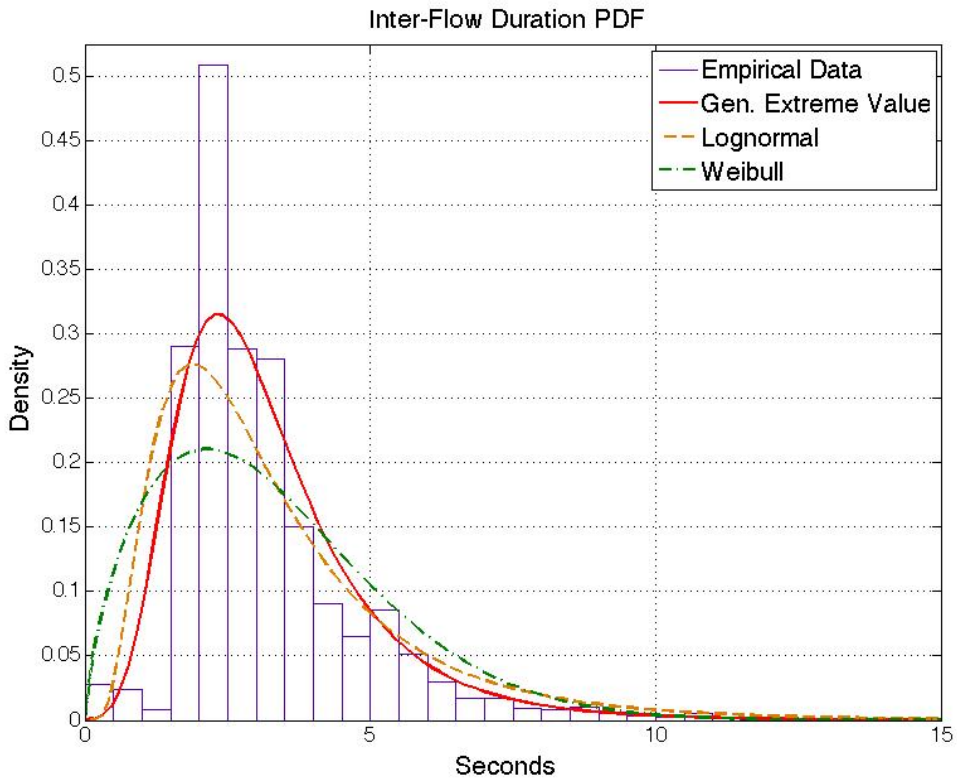


Fig. 6. PDFs of Inter-flow duration measurements are shown for over 4,000 samples from the university network. The Weibull fit (green dashed curve) and lognormal fit (yellow dashed curve) are not as tight fitting to the empirical data (blue bars) as the generalized extreme value curve (solid red line).

Extreme Value Distribution

The extreme value distribution is useful for modeling situations where the outlying observation of a block of observations is the object of focus. There are both maximum and minimum extreme value cases. More specifically, the extreme value distribution can be used to model the smallest or largest value among a large set of independent, identically distributed values of measurements. The breaking tension of a chain, which occurs at the minimum breaking tension of the weakest link, is an example of an application for the minimum extreme value distribution. The highest daily return on a stock could be an example of a maximum extreme value case. The minimum extreme value curve is skewed to the left (i.e. mean < median < mode), whereas the maximum extreme value curve is skewed to the right (i.e. mode < median < mean) [17]. The probability density function of the extreme *minimum* value distribution is expressed below.

$$y = f(x \mid \mu, \sigma) = \frac{1}{\sigma} \exp\left(\frac{x-\mu}{\sigma}\right) \exp\left(-\exp\left(\frac{x-\mu}{\sigma}\right)\right)$$

The PDF of the extreme *maximum* value distribution is found by taking the negative of the two $(x-\mu)/\sigma$ terms above. Both functions have a location parameter μ and a scale parameter σ. As can be seen by the presence of the base of the natural exponent in the equation above, the extreme value distribution is best suited for modeling the

extreme values of phenomena whose tails decay exponentially fast—similarly to the tails of the normal distribution or the exponential distribution. Since the Weibull distribution was previously mentioned, it is worth noting that the extreme value PDF can be related to the Weibull distribution as follows: If a random variable T has a Weibull distribution with scale parameter α and shape parameter β, then ln T has an extreme value distribution with parameters $\mu = \ln \alpha$ and $\sigma = 1/\beta$.

The notion of a maximum extreme value is particularly intuitive for our model of inferring web browser redirection because it relates nicely to the maximum allowable threshold (1200ms) in the initial work for the inter-flow duration filter. The false negatives (undetected redirects) correspond to flows with inter-flow durations in the "long-tail" region of the maximum extreme value PDF.

Next, we will see how the extreme value intuition can be maintained and the fit improved by expanding our model to the generalized extreme value model. Recall that the measurement of inter-flow duration is an observation of the measured time between the start of flow (a) to the redirection server and the start of flow (b) to the termination server. If we let m designate the theoretical best-case round-trip time between a given client and a typical web server on the Internet during a non-congested window of time when throughput is highest, then m could loosely represent a minimum bound on the maximum extreme value of a web browser redirection event. The type II generalized extreme value distribution meets this property and is described next.

Generalized Extreme Value

The generalized extreme value (GEV) function introduces a third parameter, k, to designate the shape of the GEV PDF. The probability density function for the generalized extreme value distribution with location parameter μ, scale parameter σ, and shape parameter $k \neq 0$ is

$$y = f(x \mid k, \mu, \sigma) = \frac{1}{\sigma} \exp\left(-\left(1 + k \frac{x - \mu}{\sigma} \right)^{-\frac{1}{k}} \right) \left(1 + k \frac{x - \mu}{\sigma} \right)^{-1-\frac{1}{k}}.$$

The function above is also subject to the constraint below.

$$1 + k \frac{x - \mu}{\sigma} > 0$$

The condition $k > 0$ corresponds to the type II case, while $k < 0$ corresponds to the type III case. The condition $k = 0$ corresponds to the type I case, which reduces to the non-general form of the maximum extreme value PDF, below.

$$y = f(x \mid 0, \mu, \sigma) = \frac{1}{\sigma} \exp\left(-\exp\left(-\frac{x - \mu}{\sigma} \right) - \frac{x - \mu}{\sigma} \right)$$

Like the extreme value distribution, the generalized extreme value distribution can be used to model the smallest or largest value among a large set of independent,

identically distributed observations of a random phenomenon. The generalized extreme value model unites the three simpler distributions into a single form that allows for a continuous range of shapes composed of any of the three simpler distributions. Types I, II, and III are sometimes also referred to as the Gumbel, Fréchet, and Weibull types. The type II (Fréchet) case is equivalent to taking the reciprocal of values from a standard Weibull distribution.

The domain of the generalized extreme value distribution is (-∞, ∞), but it is worth noting the differences in the domains of each of the three generalized extreme value types. The maximum GEV domains are as follows: the type I has domain (-∞, ∞); the type II has domain (m, ∞); and the type II has domain (-∞, M). The values m and M are used to represent the finite values of the minimum and maximum, respectively. It is easily shown that for a type II GEV distribution

$$m = -\frac{\sigma}{k} + \mu \text{, where } k > 0.$$

Similarly, for a type III GEV distribution

$$M = -\frac{\sigma}{k} + \mu \text{, where } k < 0.$$

The following figure shows the "standard" GEV of maximum values, that is with $\mu = 0$ and $\sigma = 1$. Values of -0.5, 0, and 0.5 are chosen for k to distinguish the

three types. Accordingly, as can be seen below, the values of *m* and *M* occur at -2 and 2, respectively for these sample plots of the standard GEV distribution.

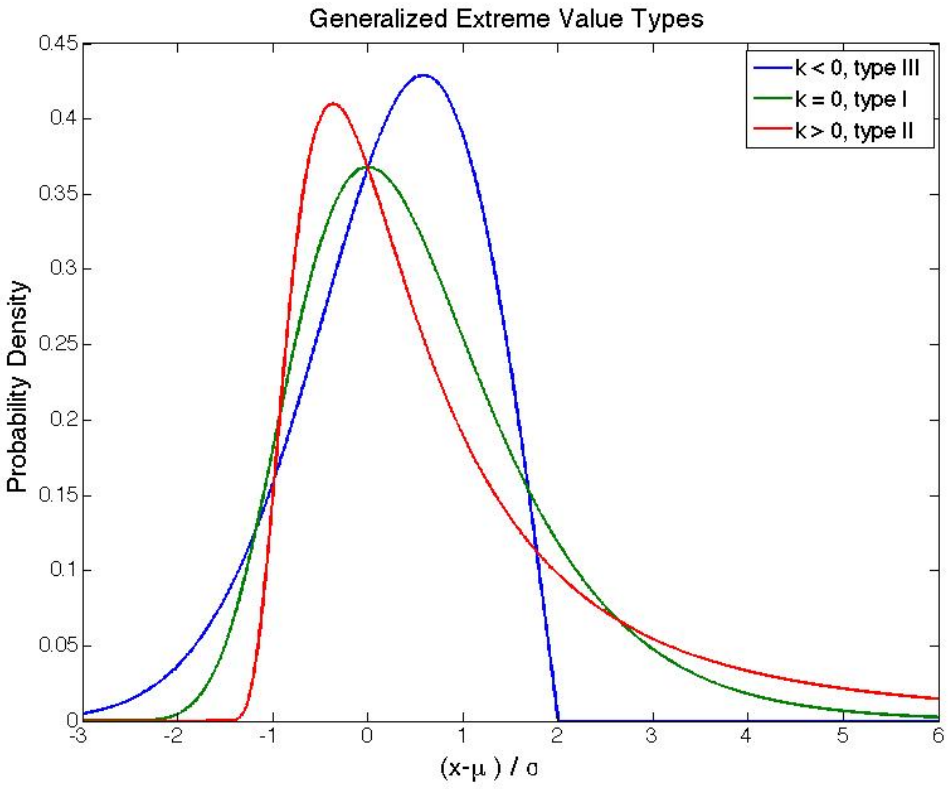


Fig. 7. Samples of the three cases of Generalized Extreme Value PDFs normalized by (x-μ)/σ. The k values are -0.5, 0, and 0.5 for the Type III, Type I, and Type II, respectively.

A final note about the generalized extreme value distribution is with respect to its qualitative properties. When considering the tails, the type III distribution has a finite tail; the type I distribution has an exponentially decreasing tail; and the type II distribution's tail decreases at a less extreme polynomial rate. Since the empirical data has a fairly heavy tail with outliers that can last into the tens of seconds for inter-

flow duration, we are not surprised that the type II distribution had the best fit when we consider the tail of the data. More thought provoking, however, is the way that the model of the type II generalized extreme value distribution fits the underlying phenomenon for inter-flow duration. As previously alluded to, the lower minimum limit, m, allows for the construction of a model where the minimum value of the maximum extreme value distribution cannot fall below a certain threshold. In the context of a web browser redirection traversing a network this lower limit could be represented by a factor of two times the minimum round-trip from the client to nearby servers on a fast, uncongested link. The factor of two occurs due to the fact that there are two round trips that occur during a typical web browser redirect, he first being from client to the redirection server and the second being from client to the target server.

The Pearson Method

The Pearson method of fitting a probability distribution to a set of data points uses the values of mean, standard deviation, skew, and kurtosis of the data. These parameters come from first four moments of the data set. These corresponding values for the inter-flow duration data are as follows:

[mean = 3.2748] [std. dev. =2.1898] [skew = 4.6519] [kurtosis = 40.0567].

Computing the Pearson coefficients using statistical software yields the following values:

$$[c_1 = 0.7756] \quad [c_2 = 1.6299] \quad [c_3 = 0.0748],$$

which correspond to the coefficients of the quadratic polynomial in the following equation,

$$\frac{d}{dx}\ln(p(x)) = \frac{-(a+x)}{c_0 + c_1 x + c_2 x^2}.$$

The Pearson system tests which family of distributions best fits the data. There are eight families as described below.

0 — Normal distribution
1 — Four-parameter beta distribution
2 — Symmetric four-parameter beta distribution
3 — Three-parameter gamma distribution
4 — Not a typical distribution; the density is proportional to:

$$\left(1+2\left(\frac{x-a}{b}\right)\right)-c\cdot\exp\left(-d\cdot\arctan\left(\frac{x-a}{b}\right)\right)$$

5 — Inverse gamma distribution with location and scale parameters
6 — Snedecor's F distribution with location and scale parameters
7 — Student's t distribution with location and scale parameters

Analysis of the empirical inter-flow duration data shows that the distribution family that fits the data best is number 6—the F distribution with added location and scale parameters. The PDF of the standard F distribution for x>0 is provided below

$$y = f(x \mid \nu_1, \nu_2) = \frac{\Gamma\left(\frac{\nu_1+\nu_2}{2}\right)}{\Gamma\left(\frac{\nu_1}{2}\right)\Gamma\left(\frac{\nu_2}{2}\right)}\left(\frac{\nu_1}{\nu_2}\right)^{\frac{\nu_1}{2}} \frac{x^{\frac{\nu_1-2}{2}}}{\left(1+\frac{\nu_1}{\nu_2}x\right)^{\frac{\nu_1+\nu_2}{2}}},$$

where x is the ratio of two chi-square random variables and ν1 and ν2 are the degrees of freedom of the numerator and denominator of that ratio. Adding two additional parameters as done in the Pearson method to shift and scale x allows a very close fit to the empirical data. Below is a figure of the cumulative distribution functions of the empirical CDF of the inter-flow duration data (blue) and the fitted CDF comprised by the Pearson-fitted F location-scale model (red).

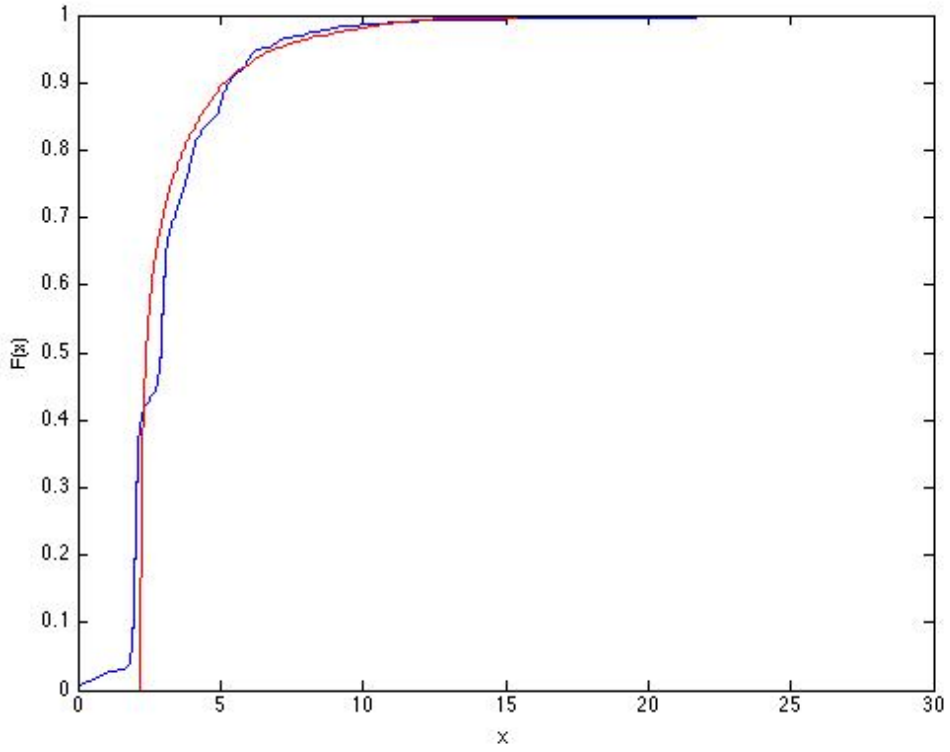


Fig. 8. The best fitting Pearson model CDF (having an F location-scale distribution) is overlaid on the empirical inter-flow duration CDF. The x-axis units are in seconds.

The overall fit is good, providing confidence in the computed moments and derived statistical parameters, but the empirical CDF is not smooth at the beginning (up to

about the first 1.75 seconds). If we zoom in to the early parts of the distribution we see that the empirical CDF increases approximately linearly from zero before beginning its steep ascension.

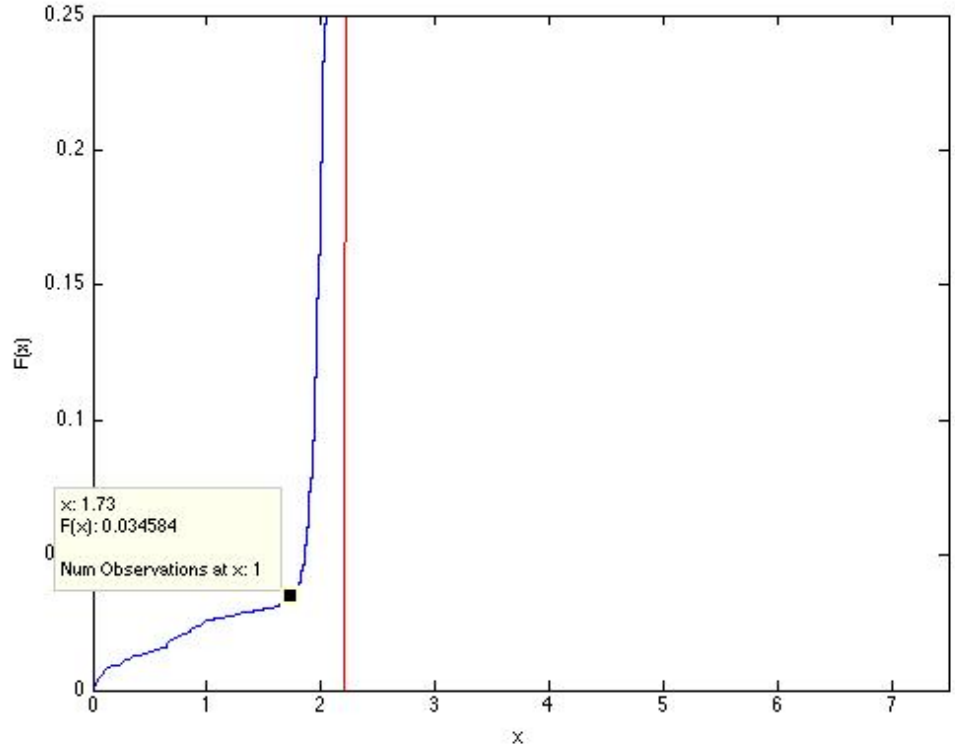


Fig. 9. The early rise of the empirical CDF is approximately linear before rising sharply at about 1.75 seconds.

The initial linear component in the rise of the empirical CDF comprises approximately 3.5% of the samples or about 150 redirects. The notion behind this observation is that there was a small subset of URLs that were accessed in a relatively short period of time compared to the others. Visual inspection of the source and target URLs accessed during this initial phase does not reveal any noticeable differences between these URLs and the overall population of probed URLs. The

initial linear shape could be an experimental aberration, but it is more likely that it is the result of an underlying property of the network. For example, the smaller subset of URLs could correspond to pages that were cached. If the pages themselves were not cached, it is possible that DNS entries were cached.

Although the Pearson method can provide a very close fit in the form of the location-scale F distribution, it should be noted that there is nothing intrinsic about the distribution that matches the underlying phenomenon of inter-flow duration of web browser redirection. Modeling a random variable as the ratio of two chi-square random variables does not have an obvious meaning in the context of explaining variation in network delay for web browsing. Such logic leads to the conclusion that the Pearson method results in a highly correlated match that is not necessarily causally related. We will continue the investigation into modeling inter-flow duration of redirects by examining how the empirical data might be explained by the Johnson method of distribution fitting.

The Johnson Method

Similar to the Pearson method, the Johnson method of fitting a probability distribution to data incorporates four known statistical quantities about the data and maps any unique 4-tuple to a well-defined distribution. Whereas the Pearson method uses moments, the Johnson method uses quantiles. The quantile values are used to transform the normal distribution with either an exponential, logistic, or hyperbolic sine transform. These transforms result in distributions traditionally referred to as

Johnson types S_L, S_B, or S_U, respectively. If the normal distribution is the best fit, then an identity transform is used and is designated as Johnson type S_N. The Johnson method makes use of the following equation.

$$X = \gamma + \delta \cdot \Gamma\left(\frac{Z - \xi}{\lambda}\right)$$

The random variable under study is X; Z is a standard normal random variable; Γ is the appropriate transform (exponential, logistic, or hyperbolic sine); and ξ, λ, γ, and δ are the scale and location parameters. Fitting a distribution to the inter-flow duration times of the redirects results in the *logistic* transform providing the best fit, corresponding to a Johnson S_B distribution. This result is significant because prior research has modeled inter-flow duration with a lognormal PDF, which would correspond to a Johnson S_L distribution. But the Johnson method, when applied directly to the empirical data, discounts the value of the S_L distribution in favor of the S_B distribution. It seems fairly conclusive that (at least for this data set) a lognormal distribution is not the best fit for characterizing the inter-flow duration phenomenon.

Below is a figure of the cumulative distribution functions of the empirical CDF of the inter-flow duration data (blue) and the fitted CDF comprised of the Johnson-fitted logistic model (red).

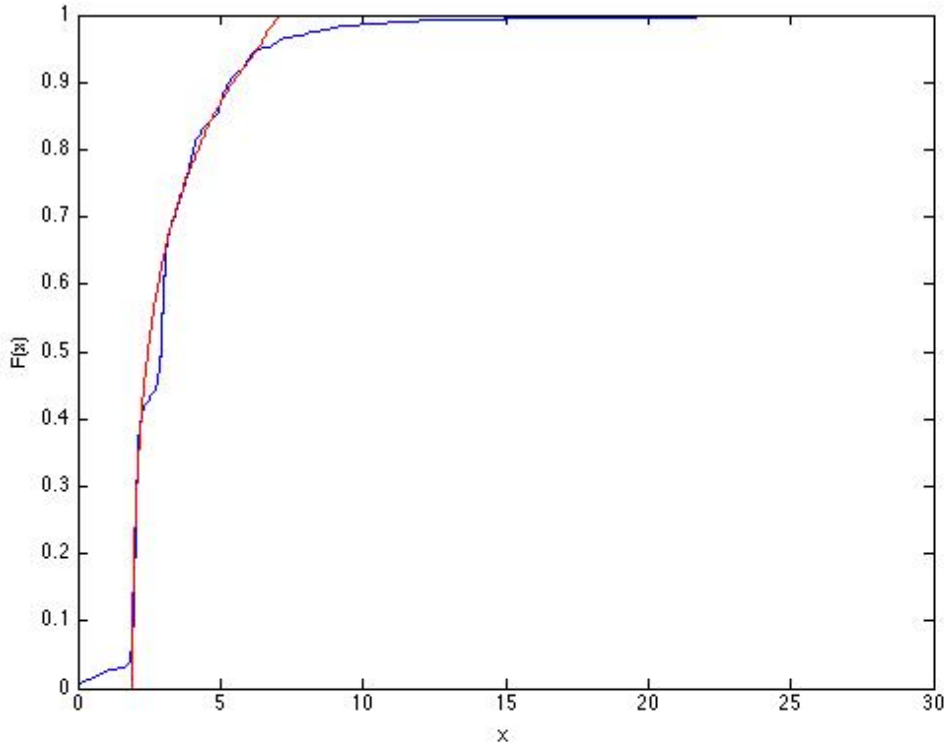


Fig. 10. The best fitting Johnson model CDF (having a logistic transform) is overlaid on the empirical inter-flow duration CDF. The x-axis units are in seconds.

The resulting parameters for the Johnson-fitted model are as follows:

[xi, lambda, gamma, delta] = [0.9244 0.4357 1.8802 5.1811],

and the quantiles were computed using the following standard normal distribution values

[-1.5 -0.5 0.5 1.5],

which, in turn, correspond to the following values in seconds for the empirical inter-flow duration times in seconds

[1.90 2.07 3.30 5.97].

By visual inspection of the fitted CDF with respect to the empirical data, the overall fit is good, providing confidence in the quantile values and derived statistical parameters.

Additional Distribution Attributes

Adding one final round of analysis to the inter-flow duration data. We check the Kolmogorov-Smirnov and chi-square statistics of the empirical data fitted to each of the candidate parametric models. The results are shown in the chart below. The distributions are ordered as they were in TABLE 5 by order of best log-likelihood fit.

TABLE 6 Additional statistical, domain, and qualitative shape data for the distributions

Distribution	**Kolmogorov-Smirnov**		**Chi-Square**		**Domain**	**Bell-Shaped with Right Skew**
	Stat	Rank	Stat	Rank		
Gen. Extreme Val.	0.11067	1	2735.9	1	(a,∞)	Yes
Log-logistic	0.20960	10	2988.9	2	[0,∞)	Yes
Gamma	0.24603	14	5027.5	10	[0,∞)	Yes
Logistic	0.19363	7	6008.7	13	(-∞,∞)	No
Weibull	0.18723	6	5885.0	12	[0,∞)	Yes
Lognormal	0.20771	9	3010.9	3	(0,∞)	Yes
Rayleigh	0.17552	5	5601.4	11	[0,∞)	Yes
Rician	0.19514	8	3407.8	4	[0,∞)	Yes
Generalized Pareto	0.13910	3	n/a	n/a	[a,∞)	No
Exponential	0.38636	15	4778.2	8	[0,∞)	No
Normal	0.21322	11	4404.9	6	(-∞,∞)	No
Birnbaum- Saunders	0.39860	16	7171.8	14	(0,∞)	Yes
Inverse Gaussian	0.22770	12	4356.1	5	(0,∞)	Yes
Extreme Value	0.22818	13	4722.7	7	(-∞,∞)	Yes
Pearson Type VI	0.17447	4	4853.5	9	[0,∞)	Yes
Johnson S_B	0.13835	2	n/a	n/a	[a,b]	Yes

The Kolmogorov-Smirnoff and chi-square columns provide absolute statistical values and relative ranking values so that the distributions can be compared accordingly. Again, the Generalized Extreme Value distribution was the best fit, ranking first in both categories. Statistical values are also provided at the bottom section of the table for the Pearson Type VI (the best-fitting Pearson distribution) and the Johnson S_B distribution (the best-fitting Johnson distribution).

The last two columns in the table present values for the domain and the qualitative shape characteristics for each distribution. Note that these values correspond to this specific data. In general, many of the distributions have shape characteristics that can change widely from one family to another based upon the values of the parameters. Since the empirical data is bell-shaped and skewed to the right, distributions that support that shape type naturally fit to that shape. There were four exceptions that did not fit this shape. The exponential and generalized Pareto distributions were not bell-shaped. They only exist as decaying (monotonically decreasing) functions. The logistic and normal distributions are necessarily symmetric and therefore do not match the right skew in the empirical data.

The domain of the empirical data was bounded at a minimum value greater than zero. The generalized Pareto and Johnson S_B distributions are the only other distributions that fit that criterion. In addition, the Johnson S_B distribution is the only distribution in the table that is bounded on the right by a maximum value.

Synthesis of Distribution-Fitting Results

The results of the distribution-fitting tests are very conclusive in that they all find the generalized extreme value distribution to provide the best fit to the data. The maximum likelihood estimation, the Kolmogorov-Smirnov statistic test, and the chi-square goodness-of-fit test all support this conclusion. More importantly, however, is the notion that there is an underlying relationship in the mechanism of the web browser redirection and the properties of the network that may be causing the data to shape with the generalized extreme value distribution and not just correlate to it. Such a relationship is valuable because it provides greater confidence in the statistical model and in predicting how the model might change when underlying factors change.

One surprising result from the distribution-fitting analysis is how long the inter-flow duration times were. The mean time of over 2000ms was longer than the devised threshold time of 1200ms. The data samples were constructed by running a script that visits known-redirect URLs from inside the campus network, while the timing and size information is collected from the university's gateway router. It is possible that an unaccounted-for variable could have been introduced that delayed the times. It is also possible that the longer times simply reflect a changing trend in malicious redirects. This open question is discussed further in the *future work* section of the concluding chapter.

Chapter 8: Conclusion

This final chapter reviews the essential elements of this thesis and presents ideas for future work.

Summary

In a network security environment where malware is increasingly delivered via web browsers, security researchers seek method to detect such web-based threats. If the malware, itself, cannot be detected due to encryption or unavailable IDS signatures, then alternative indicators are sought. Web browser redirection is one indicator that has been linked to widespread web-based malware. Meanwhile, NetFlow has become a useful tool for security awareness. NetFlow provides the benefit of being unaffected by the use of encryption that hinders content-based intrusion detection systems.

This thesis presented a method for using NetFlow to infer the presence of web browser redirection on a network. We implemented a blacklist-based method for labeling malicious traffic and showed that this malicious traffic occurs 7 times more frequently in network traffic that meets the redirect filter heuristics—confirming the expectation that redirects and malicious activity have a relevant association.

This thesis also presented an in-depth analysis of parametric models for the inter-flow duration time of web browser redirects. We broadened the possible

distribution candidates for this paramount feature and conclude that the generalized extreme value distribution provides superior fit for the traffic samples studied on our university network. This finding connects extreme value theory with intuition of round-trip-time extrema.

Applications and Future Work

The methods presented in this thesis could be used to filter web browsing traffic to send higher risk flows (i.e. probable redirects) to a specialized security system for follow-on processing. This research may also be valuable to the field of network-based application recognition, where the activity in upper-layer protocols is inferred based upon behavioral analysis of network flows.

Future work should incorporate samples of packet capture or similar "ground truth" verification methods to quantify the false positive rate and false negative rate of the redirection flow heuristic. Having verified detection percentages, future work may also wish to experiment with additional detection thresholds and explore the sensitivity as the threshold is varied.

The probability distributions of other key features, such as flow size and flow duration could be analyzed in depth, the way inter-flow duration was analyzed in Chapter 7. These distribution models could also be tested for specificity to the networks of different organizations. The goal would be to determine how universal the parameter thresholds are or understand how they vary among different networks.

The suspicion is that the heuristics will vary over time and across network topologies…much like the unsanctioned activities they seek to uncover.

Bibliography

[1] Aitchison, J. and Brown, J. A. C.,(1957), The Log-normal distribution, Cambridge University Press, New York and London.

[2] Antonakakis, M. et al. 2011. Detecting malware domains at the upper DNS hierarchy. *Proceedings of the 20th USENIX Security Symposium, USENIX Security* (2011), 27.

[3] Axelsson, S. (1999). The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection. Proceedings of the 6th ACM conference on Computer and communications security - CCS '99, 1–7.

[4] Beirlant, J; Y. Goegebeur; J. Segers; J. Teugels (2005): Statistics of Extremes. Theory and Applications. John Wiley & Sons Ltd. 490p

[5] Bilge, Leyla, Davide Balzarotti, William Robertson, Engin Kirda, and Christopher Kruegel. "Disclosure: detecting botnet command and control servers through large-scale NetFlow analysis." In Proceedings of the 28th Annual Computer Security Applications Conference, pp. 129-138. ACM, 2012.

[6] Claise, B. 2004. Cisco systems NetFlow services export version 9.

[7] Claise, B. 2008. Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information.

[8] Coles, S (2001): An Introduction to Statistical Modeling of Extreme Values. Springer Series in Statistics. Springer Verlag London. 208p

[9] Colon, M. 2013. New study finds malware variants skirting AV, mostly delivered via web. *SC Magazine*.

[10] Coskun, B. and Dietrich, S. 2010. Friends of An Enemy : Identifying Local Members of Peer-to-Peer Botnets Using Mutual Contacts Categories and Subject Descriptors. *Proceedings of the 26th Annual Computer Security Applications Conference* (2010), 131–140.

[11] CWE-601: URL Redirection to Untrusted Site ('Open Redirect'): 2012. *http://cwe.mitre.org/data/definitions/601.html*.

[12] DNS-BH – Malware Domain Blocklist: White Paper: *http://www.malwaredomains.com/?page_id=6#Summary*. Accessed: 2013-02-15.

[13] Embrechts, P., C. Klüppelberg, and T. Mikosch. Modelling Extremal Events for Insurance and Finance. New York: Springer, 1997.

[14] Foroushani, V.A. et al. 2008. Intrusion detection in encrypted accesses with SSH protocol to network public servers. *2008 International Conference on Computer and Communication Engineering*. (May. 2008), 314–318.

[15] Goh, V.T. et al. 2010. Experimenting with an Intrusion Detection System for Encrypted Networks. *International Journal of Business Intelligence and Data Mining*. 5, 2 (2010), 172–191.

[16] Gumbel, E. J. (1954), Statistical Theory of Extreme Values and Some Practical Applications, National Bureau of Standards Applied Mathematics Series 33, U.S. Government Printing Office, Washington, D.C.

[17] Gupta, B. C. and Guttman, I. Statistics and Probability with Applications for Engineers an Scientists. New Jersey: Wiley, 2012.

[18] How are URLs Verified to be Malicious?: *http://www.malwaredomainlist.com/forums/index.php?topic=1634.0*. Accessed: 2013-02-15.

[19] Howard, F. 2012. Exploring the Blackhole exploit kit. NakedSecurity. http://sophosnews.files.wordpress.com/2012/03/blackhole_paper_mar2012.pdf Sophos Labs. March (2012).

[20] Hu, X., Knysz, M., and Shin, K.G. 2009. Rb-seeker: Auto-detection of redirection botnets. *Proc. of 16th Annual Network & Distributed System Security Symposium (NDSS)*. (2009).

[21] Iliofotou, M. et al. 2007. Network Monitoring using Traffic Dispersion Graphs (TDGs). *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (2007), 315–320.

[22] Joglekar, S.P. and Tate, S.R. 2005. ProtoMon : Embedded Monitors for Cryptographic Protocol Intrusion Detection and Prevention. *Journal of Universal Computer Science*. 11, 1 (2005), 83–103.

[23] Johnson, N. L., S. Kotz, and N. Balakrishnan. Continuous Univariate Distributions. Vol. 2, Hoboken, NJ: Wiley-Interscience, 1994.

[24] Kent, A. D., Liebrock, L. M., & Neil, J. 2013. Web Adoption : An Attempt Toward Classifying Risky Internet Web Browsing Behavior. LASER 2013 Workshop.

[25] Kotz, S., and S. Nadarajah.Extreme Value Distributions: Theory and Applications. London: Imperial College Press, 2000.

[26] Koch, R. and Rodosek, G.D. 2010. Command Evaluation in Encrypted Remote Sessions. *2010 Fourth International Conference on Network and System Security*. (Sep. 2010), 299–305.

[27] Lam, J. Top 25 Series - Rank 23 - Open Redirect. SANS Software Security Institute. (2010 Mar.) [Online], Available: http://software-security.sans.org/blog/2010/03/25/top-25-series-rank-23-open-redirect

[28] Leontiadis, N. et al. 2011. Measuring and Analyzing Search-Redirection Attacks in the Illicit Online Prescription Drug Trade. *Measurement*. (2011), 1–17.

[29] McHugh, J. (2000). Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Transactions on Information and System Security, 3(4), 262–294.

[30] Mathworks. (2013, Nov). Extreme Value Distribution (R2013) [Online], Available: http://www.mathworks.com/help/stats/extreme-value-distribution.html

[31] Mathworks. (2013, Nov). Generalized Extreme Value Distribution (R2013) [Online], Available: http://www.mathworks.com/help/stats/generalized-extreme-value-distribution.html

[32] Mathworks. (2013, Nov). Modelling Data with the Generalized Extreme Value Distribution (R2013) [Online], Available: http://www.mathworks.com/help/stats/examples/modelling-data-with-the-generalized-extreme-value-distribution.html

[33] Mavrommatis, P. and Provos, N. 2008. All Your iFRAMEs Point to Us. *USENIX Security '08* (San Jose, CA, 2008), 1–16.

[34] McRee, R. 2008. Open Redirect Vulnerabilities: definition and prevention. *INSECURE*. Page 43. Issue 17. July 2008.

[35] Nagaraja, S. et al. 2010. BotGrep : Finding P2P Bots with Structured Graph Analysis. *Proceedings of the 19th USENIX Conference on Security* (2010).

[36] NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/, November, 2013.

[37] Palo Alto Networks. (2013, Mar). The Modern Malware Review, 1st Edition. [Online] Available: http://media.paloaltonetworks.com/documents/The-Modern-Malware-Review-March-2013.pdf

[38] Selvan, Sabari. (2013, Nov). Another Mass IFrame Injection Attack | 350,000 ASP Sites Infected. [Online], Available: http://www.ehackingnews.com/2011/10/another-mass-iframe-injection-attack.html

[39] Shue, C.A. et al. 2008. Exploitable redirects on the web: Identification, prevalence, and defense. *Proceedings of the 2nd conference on USENIX Workshop on offensive technologies* (2008), 1–7.

[40] Stover, S. et al. 2007. Analysis of the Storm and Nugache trojans: P2P is here. *USENIX ;login*. 32, 6 (2007), 18–27.

[41] The Carrot and the Stick Project: *http://tcats.stop-spam.org/tcats/bnbl/*. Accessed: 2013-02-20.

[42] Random Sampled NetFlow. Cisco IOS Software Releases 12.3 T. http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/nfstatsa.html

[43] Testing Whether the Shape Parameter is Zero in the Generalized Extreme-Value Distribution. J. R. M. Hosking. Biometrika , Vol. 71, No. 2 (Aug., 1984), pp. 367-374