2017

# Epidemiology of Browser-Based Malware

Sindhura Jaladhanki
*University of Rhode Island*, sindhu21aug@gmail.com

# EPIDEMIOLOGY OF BROWSER-BASED MALWARE

BY

SINDHURA JALADHANKI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

UNIVERSITY OF RHODE ISLAND

2017

MASTER OF SCIENCE THESIS

OF

SINDHURA JALADHANKI

APPROVED:

Thesis Committee:

Major Professor Lisa DiPippo

Natallia Katenka

Joan Peckham

Yan Sun

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND
2017

## ABSTRACT

The presence of personal financial data, intellectual property, and classified documents on University computer systems makes them particularly attractive to hackers, but not well prepared for their attacks. The University of Rhode Island (URI) is one of the few institutions collecting network traffic data (NetFlow) for inference and analysis of normal and potentially malicious activity. This research focuses on web-based traffic with client-server architecture and adopts simple probability-based transmission models to explore the vulnerability of the URI web-network to anticipated threats. The fact that the URI firewall captures only traffic data in- and out- of URI necessitates the modeling of internal un-observed traffic. Relying on a set of intuitive assumptions, we simulate the spread of infection on the dynamic bipartite graph inferred from observed external and modeled unobserved internal web-browsing traffic and evaluate the susceptibility of URI nodes to threats initiated by random clients and clients from specific countries. Overall, the results suggest higher rates of infection for client nodes compared to servers with maximum rates achieved when infection is initiated randomly. Remarkably, very similar rates are observed when infection is initiated from 100 different clients from each of selected countries (e.g., China, Germany, UK) or from one most active node from Denmark. Interestingly, the daily analysis over a three-month period reveals that the simulated infection rates that are not consistent with the intensity of the traffic and the pattern of network characteristics which are dependent on how the nodes are related in the network, such as assortativity and global clustering coefficient, may indicate the presence of compromised node activity and possible intrusion.

## ACKNOWLEDGMENTS

I would like to acknowledge my advisors, Dr. Lisa Dipippo and Dr. Natallia Katenka for their guidance throughout my study at URI. They have been very supportive of all my efforts in spite of my difficulties trying to find a balance between school, work and personal life. They have granted freedom in developing the analysis, yet were always available for judicial advice. This research and thesis work would not have been possible without their constant support and motivation. I would also like to thank my committee members Dr. Joan Peckham and Dr. Yan Sun for agreeing to review my work and comment on it.

Many thanks to my best friend and husband Uday, and my dearest parents for their constant encouragement and support all through the years.

## TABLE OF CONTENTS

## LIST OF TABLES

TABLE PAGE

## LIST OF FIGURES

FIGURE PAGE

# CHAPTER 1

# INTRODUCTION

## 1.1 MOTIVATION

The Internet provides access to an enormous area of research and information, yet cyber-attacks and virus outbreaks can result in huge monetary losses and exposure of personal and confidential data. This raises a very concerning question, whether the Internet is an information-providing tool or computer infection hazard [1]. The era of the Internet of things (IoT) has brought devices that provide convenience in terms of communication and usage but pose an unnecessary risk to security and exposure of classified and personal information. According to cybersecurity statistics, 51% of US adults suffered some kind of security incident between Dec 1, 2015, and Dec 1, 2016 [2]. Cyber-attacks and malware cause a substantial threat to the country's security and economic development. New viruses evolve rapidly to counter the new methods of computer protection. Outbreaks like distributed denial-of-service (DDoS) result in interruption of a vast number of valid clients' access to planned services and control of their computer assets and activities. And worse, each new generation of outbreaks demonstrates increasing speed, virulence, and sophistication [1]. Network attacks are ascending in number with the development in transmission rates and network sizes. The global spending on information security products and services in 2016 was $81.6 billion, and estimated global cost of cyber-attacks annually was $400 billion [2].

The storage of student/faculty personal financial data, intellectual property, and some classified government documents on the computer systems of academic institutions makes them particularly attractive to hackers [5, 6]. Open networks, expansive volumes of data, scientific research results, and the flexibility of public access expose university computer systems to cyber threats that, unfortunately, come with consequences. For example, in May 2017, a strain of ransomware called 'Wanna-Cry' spread around the world, walloping millions of targets, including UK universities [10]. The University College London (UCL) reported that malware very likely passively spread from a 'compromised' website in the university system [14]. In July 2015, Harvard University announced a data breach that affected as many as eight of its colleges and administrative offices. At about the same time, the networks of six Japanese Universities came under simultaneous cyber-attacks [9]. In Mar 2016, the breach in the library of Concordia University, Canada potentially impacted anyone who had used the affected computers in the past year. Most of the recent cyber-attacks are web-based attacks. While there has been, some attention paid to the problem of web-malware spread on institutional networks [7], very little research has been done to collect and analyze network flow data of a university computer system. This type of analysis could be a valuable tool to understand the communication patterns of web-browsing participants (i.e., clients and servers) in this type of system, to learn the mechanisms by which epidemic spreads, to model the future course of epidemics in the context of existing threats on graphs with non-random structure, and possibly to alert a University's IT staff of a potential intrusion.

Lately, trend of targeted attacks that originate from specific countries have been noticed. One such attack happened in October 2016, the webpage of Kerala University in India was attacked by hackers from Pakistan [9]. In May 2015, the web servers of the College of Engineering at Pennsylvania State University were targeted by two sophisticated cyber-attacks, suspected to have originated from China [9, 10]. In January 2010, Google withdrew its search engine services from China and considered shutting down its operations altogether, citing assaults from Chinese hackers on proprietary code and information from Gmail accounts, aimed at source code repositories of high-tech companies [9].

New web-based cyber threats evolve rapidly masking and hiding malicious code within regular communication activity [11]. The evolution of web-based malware has been facilitated to a large extent by the Web popularity, by the relative simplicity of the web-development, and by the way websites and users get infected. Provos et al. (2007) emphasize the importance of this rising threat and identify four prevalent mechanisms used to inject malicious content on popular web sites: web server security, user contributed content, advertising, and third-party widgets [4]. A single visit to a compromised website is sufficient for a user to get infected and for an attacker to detect and exploit a browser vulnerability. The compromised website, in turn, is used as a vehicle to infect any client who visits this page [8]. Further, compromised clients, unaware of their infection, can transmit the infection to other servers by visiting and uploading user content on websites stored on these servers. Most existing and soon-to-be anticipated computer viruses spread passively among computers without any noticeable reaction from the system and/or end-user [7, 8, 11]. The dissemination

mechanisms of these ‘new’ infections depend on the type of application and the structure of the communication networks inherent in network systems under consideration [1, 12].

To achieve an ability to control and prevent epidemic outbreak on the University of Rhode Island network, this study focuses on the analysis of web-browsing application activity using network flow data collected at the URI firewall in a period of 90 days from February to May of 2014 (Refer Section 2.1). The URI is one of the few academic institutions collecting network traffic (NetFlow) for inference and analysis of normal and potentially malicious activity. The fact that the URI firewall captures only traffic data in- and out- of URI brings the need for modeling of internal URI traffic and an additional layer of complexity of the proposed research. The modeling of internal traffic is based on simple intuitive assumptions that URI nodes (clients and servers) that are active externally are also active internally and the intensity of the internal activity is consistent over time with external activity. Specifically, we utilize the bipartite graph modeling approach proposed by Tarissan et al. for Internet topology networks [25]. This approach takes as input the node degree sequence for both layers and randomly generates a bipartite graph respecting those distributions. We adapt this approach to incorporate overall external activity of URI servers and clients (i.e., strength distribution) and the intensity of traffic over time thereby modeling a dynamic bipartite graph.

To simulate malicious activity that can propagate from clients to servers and servers to clients in a dynamic manner, we combine both observed external traffic and modeled internal traffic and construct a dynamic bipartite network, which will serve as

a basis for SI propagation model similar to one described in [13]. We use the proposed simulation approach to evaluate susceptibility of URI nodes to threats initiated by random clients and clients from specific countries with the most vigorous communication with URI (e.g., China, UK). We perform simulations varying sets of parameters, number of iterations, observation periods. We employ parallel computing techniques to the speed up the simulation process.

A central theme of this study includes the following goals:

1. Developing a model of a network that captures the data traffic flowing into the university network.
    a. Preprocess the URI network flow data
    b. Generate an external web-traffic graph model with clients and servers as nodes and observed flows between theses node as edges.
    c. Build the URI internal traffic stochastically to understand flow of data within URI.
    d. Analyze the network characteristics of external, internal and combined network.
2. Evaluating susceptibility of URI nodes to threats originated from various sources based on simulations, varying transmission parameters, number of iterations, and observation periods.
    a. Evaluate daily the fraction of infected servers, clients, URI servers and URI clients when infection is initiated from various sources.
    b. Understand the pattern of fraction of infected nodes over time to predict the possibility of intrusion.

## 1.2 RELATED WORK

Network Flow data are records that represent aggregated traffic between two hosts. The information saved in a network flow record includes the IP address and port numbers of the source and destination, the protocol type of the traffic, the volume of traffic sent and various other attributes. The data is collected at a granularity that is optimal for tools that aim to enhance network security or provide network situational awareness [16]. General properties of network traffic have been studied intensely for many years [12,13,14,15,16,18]. The majority of these traffic analysis studies have been focused on the packet level, IP flow, protocol information and end-to-end behavior for detection of anomalies. The Virginia Tech, Blacksburg University collected network flow data to perform research on malware propagation, but their research was based on ring-based flow model involving packet and flow data [18]. The IP-flow level of clustering of anomalies of similar behavior [13] was performed by researchers at University of Wisconsin to show that anomalies can be exposed effectively when aggregated with a large amount of additional traffic. In [15], numbers of IP-flow, bytes and packets based analysis were employed to detect anomalies.

Rather than becoming over-whelmed by trying to examine each packet that traverses the network, in our study, we look at higher-level trends of traffic flow across the network. These trends can reveal interesting patterns and provide enough information to be useful that may otherwise be "lost in the noise" if we try to examine raw packet traces. Several analytical papers presented their work on creating visualization tools, which can depict a wide range of information about the characteristics of an entire network on a single screen [16, 17]. Though we involve

identifying network characteristics in this study, our focus is mainly on evaluating fraction of infection over time using simulated epidemic spread on the bipartite network graph.

Epidemic modeling on graphs has been an area of intense interest among researchers working on network-based dynamic process models. Epidemic modeling is concerned with three primary issues: (i) understanding the mechanisms by which epidemics spread, (ii) predicting the future course of epidemics, and (iii) achieving an ability to control the spread of epidemics [23]. Below we provide a brief overview of results for a traditional epidemiological model, followed by analogous models that have emerged in the literature on network-based extensions.

Traditional epidemiological models are based on the assumption of population wide random-mixing; that is, each individual has a small and equal chance of coming into contact with any other individual. In practice, however, each individual has a finite set of contacts to whom they can pass infection. The ensemble of all such contacts forms a 'mixing network'. Models that incorporate network structure avoid the random-mixing assumption by assigning to each individual a finite set of permanent contacts to whom they can transmit infection and from whom they can be infected. [24].

The most commonly used class of continuous-time epidemic models is the class of susceptible-infected (SI) or susceptible-infected-removed (SIR) models. A population of N individuals is divided into three states: susceptible (S), infective (I), and removed (R). In this context "removed" means individuals who are either recovered from the disease and immune to further infection, or dead [19]. The model states that, at any given time t, a new infective will emerge from among the susceptibles (due to contact

with and infection by one of the infected individuals) with instantaneous probability proportional to the product of the number of susceptibles s and the number of infected i. Similarly, infected individuals recover with instantaneous probability proportional to i. These probabilities are scaled by the parameters β and γ, usually referred to as the infection and recovery rates, respectively. The product form for the probability with which infected emerge corresponds to an assumption of 'homogeneous mixing' among members of the population, which asserts that the population is (i) homogeneous and (ii) well mixed, in the sense all individuals have approximately the same number of contacts in the same time, and that all contacts transmit the infection with the same probability.

The underlying assumption of homogeneous mixing is admittedly simple and, for many epidemic processes, too poor of an approximation to reality. As a result, interest has turned increasingly towards 'structured population' models, in which assumed contact patterns take into account some structure(s) within the population of interest [19, 23]. Models introduced in this area include independent household models, two-level mixing models, random network models, and social clustering models. The end effect of all of these models is, in one way or another, to impose restrictions on the contact structure within the population. Often it is convenient to represent this structure as a graph $G = (V, E)$, where the vertices $i \in V$ represent elements of the population and edges $\{i, j\} \in E$ indicate contact between elements $i$ and $j$. The contact implies the possibility for infection. The lack of an edge between vertices indicates that no infection is possible between the two [23].

Web-based communication networks are built on client-server architecture and follow a bipartite graph structure with two sets of nodes and edges that only exist between nodes of the different types. Epidemic behavior usually shows a phase transition with the parameters of the model—a sudden transition from a regime without epidemics to one with. Many of the really interesting cases of epidemic spreading take place on networks that have more structure like bipartite networks [19]. The study [21] represents the spread of sexually transmitted diseases in heterosexual populations and showed that the bipartite nature of the network must be taken into account to model the behavior of the epidemic threshold. Specifically, Gomez-Gardenes et.al. demonstrates that the inclusion of the bipartite structure can strongly affect the epidemic outbreak and can lead to an increase of the epidemic threshold. The results also point out that the larger the population, the greater the gap between the epidemic thresholds predicted. Another study [22] on Vector-borne diseases for which transmission occurs exclusively between vectors and hosts is modeled on a bipartite network. The study states that spreading of the disease strongly depends on the degree distribution of the two classes of nodes. This study also suggests that the present approach is generalizable to other models. Modeling the epidemics of malware within networks in close to real-time, however, still remains a fundamentally open task due to diverse networks and constantly changing attack patterns [18]. The above-mentioned studies serve as effective foundational methods to build an epidemiological model based on a bipartite network. Specifically, we utilize the bipartite graph modeling approach proposed by Tarissan et al. for Internet topology networks [25].

**List of References**


[1] Katenka, N. (with Crovella, M., Kolaczyk, E., and Britton, T.), "Epidemiological Models for Browser-Based Malware", Invited Poster Presentation, Eastern North American Region Conference (ENAR), Baltimore, MD, 2014.

[2] Smith, C., "100 Frightening Cyber Security Statistics and Facts", DMR Stats | Gadgets, Feb 2017.

[3] Stefan Savage, Geoffrey Voelker, George Varghese, Vern Paxson, Nicholas Weaver, "NSF CyberTrust Center Proposal", Center for Internet Epidemiology and Defenses, 2004

[4] Provos, N., McNamee, D., Mavrommatis, P., Wang, K., and Modadugu, N., "The ghost in the browser analysis of web-based malware," in Proceedings of the First Workshop on Hot Topics in Understanding Botnets, 2007.

[5] Keith Wagstaff, Chiarra Sottile, "Cyberattack 101: Why Hackers are Going After Universities," NBC News, Sept 20, 2015.

[6] Chris Bing, "Universities, not health care systems, facing highest number of ransomware attacks," Fedscoop, Sept. 21, 2016.

[7] Charles E. Harris, Laura R. Hammargren, "Higher education's vulnerability to cyber-attacks," University Business, August 2016.

[8] V. Paxson, A. Adams, and M. Mathis, "Experiences with NIMI", In Proceedings of Passive/Active Measurement (PAM), 2000.

[9] Retrieved from "https://advisory.ey.com/cybersecurity/cyber-threats-higher-education-institutions", EY Building a better working world.

[10] Lily H. Newman, “The Biggest Cybersecurity Disasters of 2017 So far,” Security, Wired, Jul 2017.

[11] Alexander Moshchuk, Tanya Bragin, Steven D. Gribble, and Henry M. Levy, A Crawler-based Study of Spyware on the Web, In Proceedings of the 2006 Network and Distributed System Security Symposium, Feb 2006.

[12] Lakhina, Anukool., Crovella, Mark., and Diot, Christophe., "Diagnosing network-wide traffic anomalies." ACM SIGCOMM Computer Communication Review. Vol. 34. No. 4. ACM, 2004

[13] Barford, Paul, et al. "A signal analysis of network traffic anomalies." Proceedings of 2nd ACM SIGCOMM on Internet measurement, 2002.

[14] Barford, Paul, and David Plonka. "Characteristics of network traffic flow anomalies." Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement. ACM, 2001.

[15] Lakhina, Anukool, Mark Crovella, and Christiphe Diot. "Characterization of network-wide anomalies in traffic flows." Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. ACM, 2004.

[16] Lakkaraju, Kiran, William Yurcik, and Adam J. Lee. "NVisionIP: Netflow visualizations of system state for security situational awareness.", Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. ACM, 2004.

[17] Yin, Xiaoxin, et al. "VisFlowConnect: netflow visualizations of link relationships for security situational awareness.", Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. ACM, 2004.

[18] Kayla M. Straub, Avik Sengupta, Joseph M. Ernst, Robert W. McGwier, Merrick Watchorn, Richard Tilley, Randolph Marchany. “Malware Propagation in Fully Connected Networks: A Netflow-Based Analysis”, IEEE 2016.

[19] Newman, M.E.J, “The spread of epidemic disease on networks”, Center for the Study of Complex Systems, University of Michigan, 2002.

[20] Jian Chang, Krishna K. Venkatasubramanian, Andrew G. West, Insup Lee, University of Pennsylvania, “Analyzing and Defending Against Web-based Malware”, ACM Survey, 2013.

[21] Jesus Gomez-Gardenes, Vito Latora, Yamir Moreno, and Elio Profumo, “Spreading of sexually transmitted diseases in heterosexual populations”, Proceedings of National Academy of Sciences USA, 2008.

[22] Donal Bisanzio, Luigi Bertolotti, Laura Tomassone, Giusi Amore, Charlotte Ragagli, Alessandro Mannelli, Mario Giacobini, Paolo Provero, Modeling Spread of Vector-Borne Diseases on Bipartite Networks, PLOS, 2010.

[23] E. D. Kolaczyk, “Statistical Analysis of Network Data: Methods and Models,” Springer Ser. Stat., 2009.

[24] Keeling, J. Matt and Eames, T.D. Ken, “Networks and epidemic models,” J R Soc Interface, Jun 2005.

[25] Fabien Tarissan, Bruno Quoitin, Pascal MéRindol, Benoit Donnet , Jean-Jacques Pansiot, Matthieu Latapy, “To-wards a Bipartite Graph Modeling of the Internet Topology,” Computer Networks, 57(11), 2331-2347, August, 2013.

# CHAPTER 2

# DATA REPRESENTATION AND CHARACTERIZATION

## 2.1 DATA DESCRIPTION

In this study, we analyze URI network flow datasets. The University of Rhode Island (URI) cyber system produces massive amounts of data on a daily basis. The log data produced by this system offers important information about the communication activity, resilience and overall 'health' of the URI network. The University is mainly collecting network traffic for inference and analysis of normal and potentially malicious behaviors. The URI network flow data captures only data flow in and out of the URI firewall. Therefore, we do not have any information about the flow of data between nodes within the URI network. This brings the need to simulate internal data flows at URI stochastically, to construct an internal network model. The datasets are relational and gathered from URI cyber security network between February and May 2014.

The University has deployed NetFlow monitoring systems on its routers for both wireless and wired traffic flows. In wireless data, the IP addresses of URI nodes are assigned dynamically by the system from a small range of addresses and typically the exact machine location remains unknown. In order to the maintain coherence with each machine location and unique IP address for each machine, we consider only wired data in this study. The network flow data sets are comprised of 37 features such as Source IP address (srcIP), Destination IP address (dstIP), IP protocol (pro), Source port (srcPort), Destination port (dstPort), Time Recorded (time), Bytes Sent (bytes), Packets Sent

(packets), Country of Source (srcCountry), Country of Destination (dstCountry), Application (application) and Department (department – for URI only) in CSV (comma-separated values) format. In this study, we only utilize the following features: time, srcIP, dstIP, srcPort, dstPort, application, srcCountry, and dstCountry.

## 2.2 CLIENT - SERVER ARCHITECTURE

This study focuses on the analysis of web-based traffic using a Client - Server Architecture. To comprehend clients and servers: (a) Clients are personal computers on which users run applications. (b) Servers are powerful machines that provide multiple clients with data/services upon browser-generated requests. There is a fundamental difference how clients and servers get infected [7].

Clients get to be distinctly infected when they visit a compromised site. Depending upon the infection classification, the injected malware frequently empowers an attacker to gain remote control over the compromised computer system and can be utilized to steal sensitive information, for example, individual documentation, email passwords and banking accounts. A compromised client, ignorant of its infection, will have the capacity to transmit infections to multiple servers by means of web pages stored on these servers and accessed by client.

Servers get infected when malicious content is injected into websites stored on this server through web server security vulnerabilities in the operating system or installed software, user contributed content (e.g., blogs, uploads), advertising (images, banners) and third-party content (widgets, scripts). Once infected, servers transform into

storage for websites where some portion of the websites is infected with malware.

Once the client or server is infected, the adversaries can even take control over the personal computer or server network. The key strokes and other confidential transactions on the compromised system are at risk from being observed by remote adversaries. The sophistication of adversaries has increased over time and exploits are becoming increasingly more complicated and difficult to analyze [7].

## 2.3 GRAPH-BASED REPRESENTATION

As the network flow data is relational in nature, it can be represented with a graph model. This representation will recognize attributes and examples of normal and anomalous patterns. The standard bipartite graph model is used to demonstrate noteworthy network characteristics and depict the Client - Server architecture.

The network graph is delineated with clients and servers as nodes and edges are connection between them. Formally, a bipartite network is a graph $G = (V, E)$, such that the vertex set $V$ may be partitioned into two disjoint sets, say $V_1$ as servers $S = \{S_1, ..., S_N\}$ and $V_2$ as clients $C = \{C_1, ..., C_N\}$ and each edge in $E$ has one endpoint in $S$ and the other in $C$ [3].

We considered two types of bipartite graph models in our analysis: Static and Dynamic. In Static graph $G_s$ edges, $E= \{e (S_i, C_j), i, j\}$, reflect presence or absence of communication between $S_i$ and $C_j$ over infection period. Whereas in Dynamic graph $G_d$ edges, $E = \{e (S_i, C_j, t_k),i,j,k\}$ reflect one or multiple temporal communications between

$S_i$ and $C_j$ overtime $t_k$ [1].

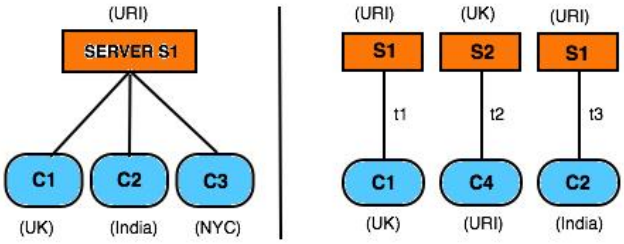


*Figure 1: Static and Dynamic Bipartite networks. Servers (Orange) and Clients (blue)*

In the static network represented in Figure 1, all the clients from UK, India and NYC US are connected to URI Server $S_1$ without time taken component into consideration. The static bipartite graph representation will be used to characterize the daily traffic in terms of graph structure. The dynamic network graph takes time into consideration and though clients $C_1$ from UK and $C_2$ from India are connected to server $S_1$, they are represented separately at different time $t_1$ and $t_3$. We use the dynamic graph to simulate the network and virus propagation in this paper.

## 2.4 DATA PRE-PROCESSING

For web-browsing applications, Internet Assigned Numbers Authority (IANA), a department of ICANN, assigned port number 80 as the official port for HTTP (www) and port number 443 as official port for HTTPS. The IP addresses with port number as 80 or 443 are classified as servers and respectively other IP addresses are classified as clients. After pre-processing of the 90-day dataset, we identify the average number of unique flows per day as 36,459. The pre-processing steps involve selecting only flows

using web-browsing, categorizing IP addresses as servers based on port numbers 80 and 443, and other IP addresses as clients respectively.

The data collection started from 02/10/2014 to 04/22/2014, stopped between 04/23/2014 to 05/06/2014 and resumed from 05/07/2014 to 05/28/2014. During the time period between 4/23/2014 to 5/6/2014, the URI network was claimed to be under real cyber-attack explaining why data was not collected during this period. Graphical representation of the data traffic per day shown in Figure 2 depicts total number of data flows per day over the period of 90-days. The fall in traffic intensity between 03/10/2014 to 03/14/2014 can be explained due to spring break week at the university. We can see the activity of nodes dropping down during the weekends and raising back during the mid-week. This provides some insight into the expected patterns of traffic on the university network.

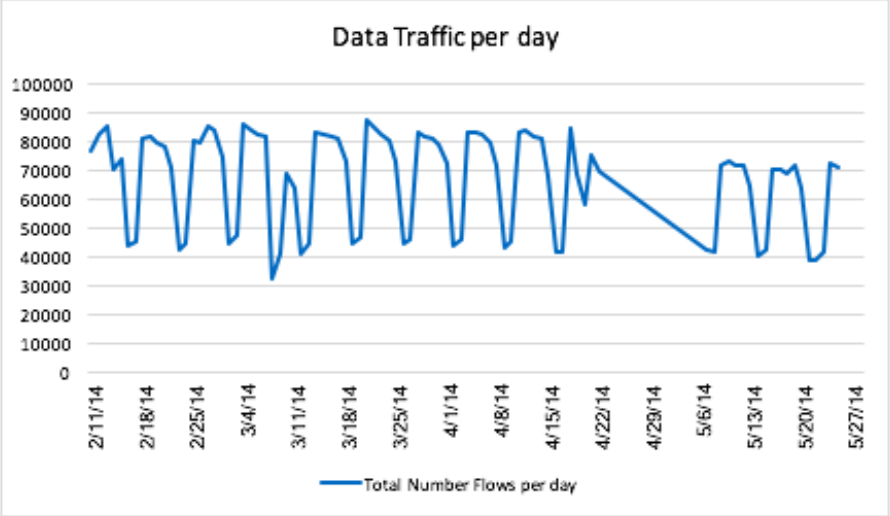


*Figure 2: Total number of data flows per day over the period of 90-days*

Furthermore, the URI servers and clients are classified based on IP addresses using the following mask: '131.128.X.X'. The total number of unique URI servers and clients in the period of 90-days is 843 and 7215 respectively. Figure 3 represents daily percentage of activity of URI servers and clients. The daily percentage is calculated as percentage of unique nodes per day to total number of unique days over the period of 90-days.

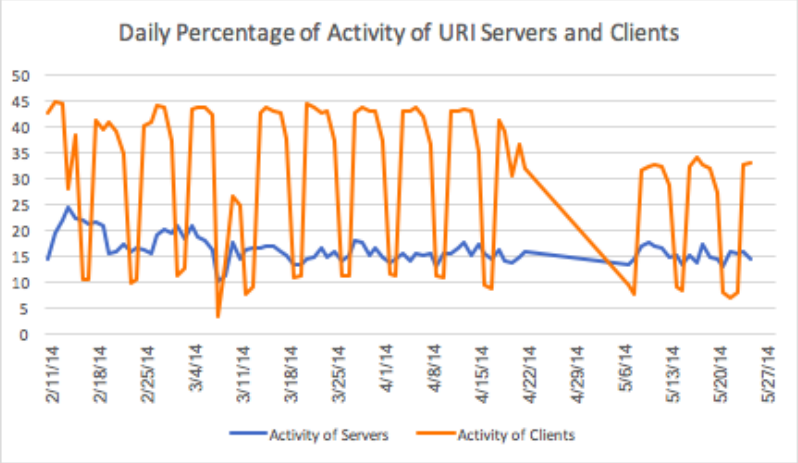


*Figure 3: Daily Percentage of Activity of URI Servers and Clients*

While cyber-attacks and malware can originate from any place, some countries are predominant in initiating such attacks. In May 2015, the web servers of the College of Engineering at Pennsylvania State University were targeted by two sophisticated cyber-attacks, suspected to have originated from China [2]. In order to consider the influence of specific countries on the flow of data in the URI network, we examined data from a subset of the days in the data set. In this sample day 02-12-2014 and sample week between 02-10-2014 to 02-16-2014, we found that countries like Canada, China, India, Denmark, Germany and UK are predominantly influential with more than 100 active clients and servers interacting with the URI network per day. The choice of the

day is based on the high volume of traffic expected in the middle of the week during regular school time. The high number of flows from these countries could be explained by the significant number of students from China at URI and by URI's collaboration with universities at Germany and UK. The column 1 in Table 1 depicts unique number of clients from various countries. The column 2 and 3 depict the total number of data flows which involve URI servers and clients from various countries. Firstly, the total number of unique clients from China are 217. These clients were involved in 417 data flows per day. When we look at a week, 2648 data flows represent consistent activity over the week. Similarly, clients from UK, Canada and India show consistent high activity.

There are certain cases of special notice, such as, Denmark had involvement in 1193 data flows with only 6 unique clients on 02-12-2014. This level of activity might raise an alarm for attention by the IT department because Denmark had less than 20 data flows per day for the rest of the week. As we dig further, Denmark client "93.160.60.22" accessed 39 unique number of URI servers, which included accessing "131.128.1.19 - uri.edu" more than 500 times, "131.128.1.130 - web.uri.edu" more than 200 times. Table 1 shows these countries and the total number of active servers and clients and the total number of data flows on the particular day 02-12-2014 and over the week.

| **Countries** | **Active Clients** (1 day) | **# Data Flows** | | **Active Servers** (1 day) | **# Data Flows** | |
|---|---|---|---|---|---|---|
| | | 1 day | 1 week | | 1 day | 1 week |
| China | 217 | 417 | 2648 | 402 | 985 | 4081 |
| UK | 249 | 301 | 1909 | 193 | 630 | 2641 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Canada | 158 | 200 | 1409 | 122 | 272 | 1637 |
| India | 215 | 250 | 1316 | 3 | 3 | 111 |
| Germany | 76 | 132 | 847 | 91 | 158 | 711 |
| Denmark | 6 | 1193 | 1679 | 7 | 7 | 68 |
| Russia | 44 | 61 | 466 | 53 | 63 | 237 |

*Table 1: Number of Active Nodes (per day & per week) of Most Influential Countries on URI network.*

## 2.5 NETWORK CHARACTERISTICS

Examining the simulated data through a bipartite network identifies some network characteristics that are useful to understand the distribution of nodes in the network and eventually influence the infection spread on the network. Graph partitioning methods are useful precisely because these characteristics will often be unobserved [3]. The presence of high-risk nodes can be quantified through two network topology features, degree assortativity and clustering coefficient.

*Degree of Bipartite Graph Nodes* represents the number of connections from a source node to the destination nodes [3]. The degree provides a good picture of connectivity of the clients and servers and when a node with high degree is infected, chances of infection propagation increases and all the nodes attached to it are highly susceptible.

*Assortativity of Bipartite Graph (r)* is the correlation between the network nodes. In general, $r$ lies between $-1$ and 1. Positive values of $r$ indicate a correlation between nodes of similar degree, while negative values indicate relationships between nodes of

different degree [3]. If the network has a negative value, it shows that high degree nodes tend to attach to low degree nodes. For example, in social network, nodes tend to be connected with other nodes with similar degree values. This tendency is referred to as assortative mixing. On the other hand, technological and biological networks typically show disassortative mixing, as high degree nodes tend to attach to low degree nodes [4].

*Bipartite Projection* is a widely-used method for compressing information about bipartite networks. Bipartite networks are a particular class of complex networks, whose nodes are divided into two sets *X* and *Y*, and only connections between two nodes in different sets are allowed. For the convenience of directly showing the relation structure among a particular set of nodes, bipartite networks are usually compressed by one-mode projection [5]. Specifically, a graph *G1 = (V1, E1)* may be defined on the vertex set *V1* by assigning an edge to any pair of vertices that both have edges in *E* to at least one common vertex in *V2*. Similarly, a graph *G2* may be defined on *V2*. Each of these graphs is called a projection onto its corresponding vertex subset [3]. If nodes '*a*' and '*b*' share at least one common destination, they are connected in the bipartite network projection. In Figure 4, example of a small bipartite graph with clients and servers is presented on the left panel and its two one-mode projections on the right panel. The projection is used in order to determine some of the network analysis methods such as clustering coefficient.

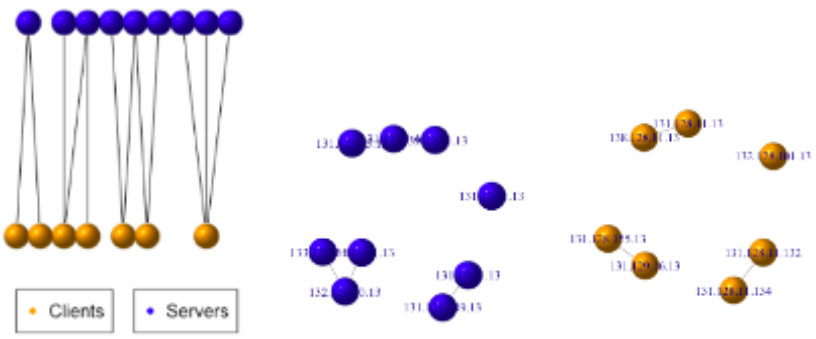


*Figure 4: Example of Bipartite Graph (left) and its projections (right).*

*Clustering coefficient of Projection Graph* is a measure of the degree to which nodes in a graph tend to cluster together. The value of the coefficient lies between 0 and 1 [3]. If the network is highly clustered with coefficient value close to 1, the network forms more connected communities which tend to connect to same node with high density ties. When they form community, all the nodes irrespective of their degree are susceptible to infection. The global clustering coefficient is defined as:

$$Cn = \frac{number\ of\ triangles\ that\ pass\ through\ the\ node\ n}{Maximum\ number\ of\ triangles\ that\ pass\ through\ the\ node\ n}$$

In this formula, the number of triangles or a connected triple is defined to be a connected subgraph consisting of three vertices and two edges. Thus, each triangle forms three connected triplets, explaining the factor of three in the formula. Intuitively, a measure of the frequency with which connected triples 'close' to form triangles will provide some indication of the extent to which edges are 'clustered' in the graph. The clustering coefficients have typically been found to be quite large in real-world networks [3].

**List of References**


[1] Katenka, N. (with Crowella, M., Kolaczyk, E., and Britton, T.), "Epidemiological Models for Browser-Based Malware", Invited Poster Presentation, Eastern North American Region Conference (ENAR), Baltimore, MD, 2014.

[2] Retrieved from "https://advisory.ey.com/cybersecurity/cyber-threats-higher-education-institutions", EY Building a better working world.

[3] E. D. Kolaczyk, "Statistical Analysis of Network Data: Methods and Models," Springer Ser. Stat., 2009.

[4] M.E.J. Newman, "Assortative mixing in networks", Phys. Rev, Lett. 89, 2002.

[5] Tao Zhou, Jie Ren, Matus Medo and Yi-Cheng Zhang, "Bipartite network projection and personal recommendation" in Physical Review, 2007.

[6] Lakkaraju, Kiran, William Yurcik, and Adam J. Lee. "NVisionIP: Netflow visualizations of system state for security situational awareness.", Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. ACM, 2004.

[7] Provos, N., McNamee, D., Mavrommatis, P., Wang, K., and Modadugu, N., "The ghost in the browser analysis of web-based malware," in Proceedings of the First Workshop on Hot Topics in Understanding Botnets, 2007.

# CHAPTER 3

# METHODOLOGY

## 3.1 SIMULATION OF INTERNAL NETWORK

The construction of an internal URI network is a new idea to understand the vulnerability of internal URI nodes to anonymous attacks coming in from various nodes outside the network and spreading within URI. The structure of unobserved internal network traffic will mimic main characteristics of the external (observed) URI network traffic. The simulation of the internal network flow adheres to the client-server architectural framework of the real-time URI network flow. Our main modeling assumptions are the following: URI nodes (clients and servers) that are active externally are also active internally and the intensity of the internal activity is consistent over time with external activity, on a particular day.

We adopt a bipartite graph model to form a URI internal data frame with the URI client IPs, URI server IPs, and timestamps randomly selected from the external traffic features. Specifically, we model a dynamic graph $\tilde{G}_d = (\tilde{S}, \tilde{C}, \tilde{E}_d)$, where each edge in the set $\tilde{E}_d$ reflects one communication between URI server $\tilde{S}_i$ and URI client $\tilde{C}_j$ that occurred at time $\tilde{t}_k$ from observation time period T, i.e., $\tilde{E}_d = \{e\ (\tilde{S}_i, \tilde{C}_j, \tilde{t}_k),\ \tilde{t}_k \in T, k = 1 \dots N_{URIflows}\}$. For each triple $(\tilde{S}_i, \tilde{C}_j, \tilde{t}_k)$, we select randomly:

1. with replacement server $\tilde{S}_i$ from a set of unique, active URI servers $S_{URI} \subset S$ proportionally to the strength of flows observed in the external traffic for $\tilde{S}_i$;

2. with replacement client $\tilde{C}_j$ from a set of unique, active URI clients $C_{URI} \subset C$ proportionally to the strength of flows observed in the external traffic for $\tilde{C}_j$;
3. without replacement timestamp $\tilde{t}_k$ from a set of timestamps recorded in the external traffic.

To ensure the uniqueness of $\tilde{t}_k$s, we add 0.5 seconds of each selected time. The sets of all unique selected servers and clients form sets $\tilde{S}$ and $\tilde{C}$ respectively. Note that the proposed approach produces a dynamic bipartite graph that preserves important properties of the observed external graph structure. The size of the internal network is generated based on a specified percentage of the size of the external network, where size is the number of data flows in the network. We simulate internal networks with three different sizes - 10%, 25% and 50% of the size of the external network and refer to each internal network based on its size in comparison. In order to maintain consistent results, we first build the 50% internal network and form the 25% internal network from the 50% internal network. Similarly, the 10% internal network is formed from the 25% internal network. Based on the understanding of how a university network is typically used, we expect to observe more external web traffic data than internal data. This assumption, however, may not be valid for other organizations such as the banking sector where external communication is limited or restricted.

Figure 5 depicts external network with URI and Non-URI nodes and internal network with URI nodes are combined and sorted based on the time variable (t).

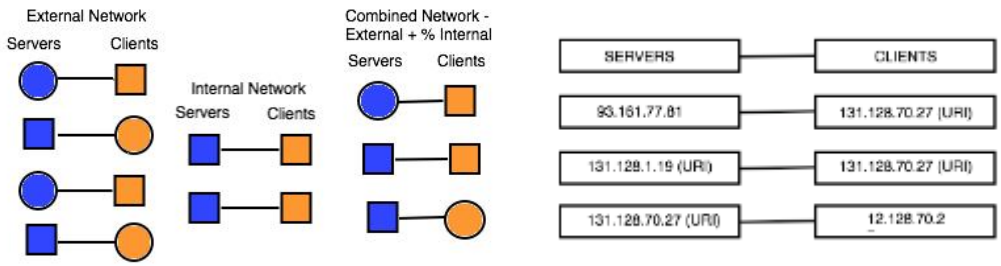


*Figure 5: Network Simulation of Combined Network with External and Internal data flows. Servers (blue) and Clients (orange), URI (square) and Non-URI (circle)*

### 3.2 EPIDEMIC MODELING

In this section, we describe the epidemic modeling by assuming a set of clients $C_i, i \in \{1,2,\dots,N_C\}$ and a set of servers $S_i, i \in \{1,2,\dots,N_S\}$ with the corresponding probabilities of infection and susceptibility:

$$\alpha_S(i) = P(S_i \text{ infected}), \beta_S(i) = P(S_i \text{ susceptible})$$

$$\alpha_C(j) = P(C_j \text{ infected}), \beta_C(j) = P(C_j \text{ susceptible}),$$

And the transmission probabilities computed as follows:

$$p_{CS}(i,j) = P(C_i \to S_j) = \alpha_C(i) \times \beta_S(j)\ I\{C_i \text{ infected}\},$$

$$p_{SC}(j,i) = P(S_j \to C_i) = \alpha_S(j) \times \beta_c(i)\ I\{S_i \text{ infected}\}.$$

Then the fraction of infected servers and clients at time t is defined as:

$$f_S(t) = \frac{Ni_S(t)}{N_s} \text{ and } f_C(t) = \frac{Ni_C(t)}{N_C},$$

where $Ni_S(t)$ and $Ni_C(t)$ are the number of infected servers and clients, respectively.

$$f_{Suri}(t) = \frac{Ni_{Suri}(t)}{N_{Suri}} \text{ and } f_{Curi}(t) = \frac{Ni_{Curi}(t)}{N_{Curi}},$$

where $Ni_{Suri}(t)$ and $Ni_{Curi}(t)$ are the number of infected URI servers and clients

respectively.

In what follows in Section 4, we have adopted the outlined probability-based transmission model to simulate the propagation of computer virus on the dynamic bipartite graphs constructed based on external traffic and combined external and internal traffic. We consider the same transmission probability of infection for all servers and clients, $p_{CS} = p_{SC} = p$, with values set up to *0.1, 0.3* or *0.5*. We perform experiments with 100 initially infected clients that are either randomly selected from a pool of all unique clients, only URI unique clients, or unique active clients from a specified country. The simulation results are summarized with the proportion of infected clients, servers, URI clients, and URI servers. The proportions of infected nodes are estimated for one day, one week, and daily over 90 days. In Figure 6 (right panel), example of simulation results is presented with the proportion of infected clients estimated for one day when P = 0.1, 0.3 and 0.5. To optimize the code and achieve high-speed performance, we used parallel computing method (Refer Section 3.3) to simulate propagation of infection for each value of p and different conditions of initial propagation. We let infection be transmitted from clients to servers and from servers to clients via communication flows ordered in time; thereby analyzing propagation of a simulated infection via nodes communicating directly and/or indirectly via common (overlapping) sets of nodes of different type.

In Figure 6, infection is initially introduced into the network from clients C1 and C2, the infection spreads to server S3 as its level of infection is less than P value. The server S1 is not infected as its level of infection is higher than P value. Also, in the final data flow, the infected server S3 infects client C3 as level of infection of C3 is lower

than P value. Therefore, the infection spread and propagation is analyzed with the client connection directly to servers and indirectly to other clients from connected servers.

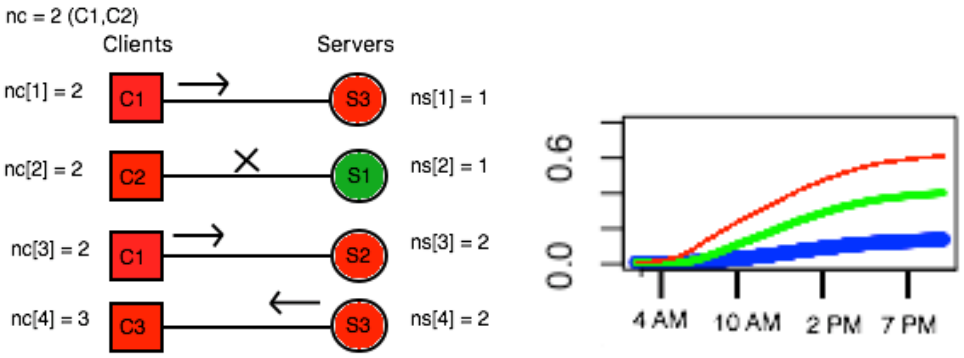


*Figure 6: Epidemic Infection Propagation (left) with Clients (squares) and Servers (circles). Infected nodes (red) and Non-infected nodes (green). Example of simulation results over a day (right) with the proportion of infected clients when P = 0.1 (blue), 0.3 (green) and 0.5 (red).*

### 3.3 PARALLEL COMPUTING

Parallel computing is a type of computation in which many calculations or processes can be carried out simultaneously. Large problems can often be divided into smaller ones, which can then be solved at the same time using multiple processors [1]. As we have considered more than one probability of infection (P = 0.1, 0.3, 0.5) in the study, we have used parallel computing to compute infection propagation for each value of P. This saves computation time and optimizes the code. In Figure 7, the graph presents the computation time taken using different numbers of cores. The sequential computation with function *lapply* takes less time than *for* loop. The best result of 9.093 mins is achieved using 4 cores on a i5 quad core computer system. Nearly 600% speed up is achieved using parallel computing methods and packages in R: *doParallel* package, *foreach*, *lapply* and *mclappy* functions.

lapply is a way to parallelize but tasks are embarrassingly parallel, where elements are calculated independently. First element and second element are independent of each other's results. In parallel package, mclapply is used instead of lapply, where number of clusters are mentioned. It returns a list of the same length as vector (atomic or list), each element of which is the result of applying function to the corresponding element of the vector [1]. The idea behind the foreach package provides a looping construct that can be viewed as a hybrid of the standard for-loop and lapply function. It looks similar to the for-loop, and it evaluates an expression, rather than a function (like in lapply) and returns a value, rather than to cause side-effects. The %do% and %dopar% are binary operators that operate on a foreach object and an R expression. The %do% evaluates the expression sequentially, while %dopar% evaluates it in parallel. We must register a parallel backend to use; else foreach will execute tasks sequentially, even when the %dopar% operator is used. The doParallel package is a "parallel backend" for the foreach package [2]. It provides a mechanism needed to execute foreach loops in parallel. The doParallel package acts as an interface between foreach and the parallel package of R. The registerDoParallel function should be called to register doParallel with foreach [2].

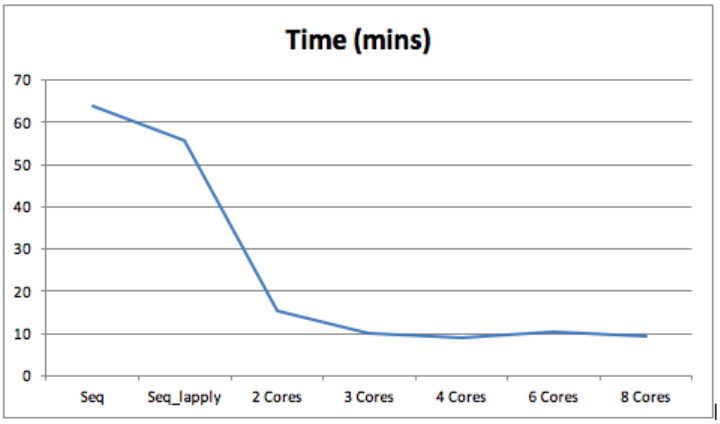


*Figure 7: Parallel Computation: Number of Cores Vs Time taken in minutes.*

## List of References

[1] K Gordon, Max, "How to go parallel in R," G-Forge, Feb 2015.

[2] Weston, Steve and Calaway, Rich, "Getting Started with doParallel and foreach", Cran-R-Project, Oct 2015.

# CHAPTER 4

# PERFORMANCE EVALUATION

We address three types of results in this section: *Graph-Based Characterization, Propagation of Infection, Effects of Time and Network Characteristics over Time.*

## 4.1 GRAPH-BASED CHARACTERIZATION

This section starts with the structural characteristics of internal, external and combined networks formed from the network flow data over one day, Wednesday, 02-12-2014. The choice of the day is based on the high volume of traffic expected in the middle of the week during regular school time. Using the data, we form three networks (internal, external and combined) and compute structural graph characteristics (see Table 2).

The number of unique clients and servers in each network type (e.g., external and combined) gives us an idea of how many nodes of each type are active and the number of data flows determines the size of the network and the total number of connections. The strength of clients and servers determines the connectivity in terms of the average number of connections observed/modeled for clients and servers. In the case of the internal network, the strength of servers is higher than clients since more clients connect to fewer servers. In the external and combined networks (see Table 2), the strength of servers and clients is similar, as there are almost the same number of servers and clients. The presence of high-risk nodes can be quantified through two network

topology characteristics such as degree assortativity and clustering coefficient. The degree assortativity measures the likelihood that nodes will preferentially form unique connections with other nodes that have similar degree distributions. Negative the assortativity degree of all the net-works, particularly in the case of internal networks; suggests that there is high chance of more popular nodes connecting to less active nodes. The values of clustering coefficient obtained from the projection graphs above 0.5 and close to 1 indicate that presence of clustered communities of clients that share common servers that they connect to; and clustered communities of servers that tend to be connected by the same clients. Overall, these results suggest that all the nodes in the network contribute to the propagation of infection to some extent.

| Network Characteristics | 10% Internal | 25% Internal | 50% Internal | External | Ex+ 10% In | Ex+ 25% In | Ex+ 50% In |
|---|---|---|---|---|---|---|---|
| #Unique (C) | 1954 | 2553 | 2858 | 12264 | 12264 | 12264 | 12264 |
| #Unique (S) | 117 | 161 | 178 | 10713 | 10713 | 10713 | 10713 |
| #Flows | 8743 | 21858 | 43716 | 87433 | 96176 | 109291 | 131149 |
| Strength (C) | 4.474 | 8.562 | 15.29 | 7.129 | 7.842 | 8.9115 | 10.694 |
| Strength (S) | 74.73 | 135.7 | 245.5 | 8.161 | 8.977 | 10.202 | 12.242 |
| Assortativity | -0.41 | -0.41 | -0.41 | -0.083 | -0.094 | -0.115 | -0.150 |
| Clustering (S) | 0.717 | 0.698 | 0.629 | 0.756 | 0.725 | 0.699 | 0.673 |
| Clustering (C) | 0.932 | 0.892 | 0.931 | 0.961 | 0.890 | 0.872 | 0.858 |

*Table 2: Network Characteristics of Internal, External and Combined (10%, 25%, 50%) network.*

The log-log plot of node degree distribution for URI clients, URI servers (Figure 8, left and right top panels) and cumulative node degree distribution of servers and clients combined (Figure 8, left bottom panel) demonstrate heavy-tail distribution property also supporting the presence of few highly active nodes in the network. The histogram (Figure 8, right bottom panel) shows the intensity of the traffic computed as frequency of the flows in a given time slot during the day. One can see that peaks hours

of activity are during the working hours between 8:00 AM to 5:30 PM.

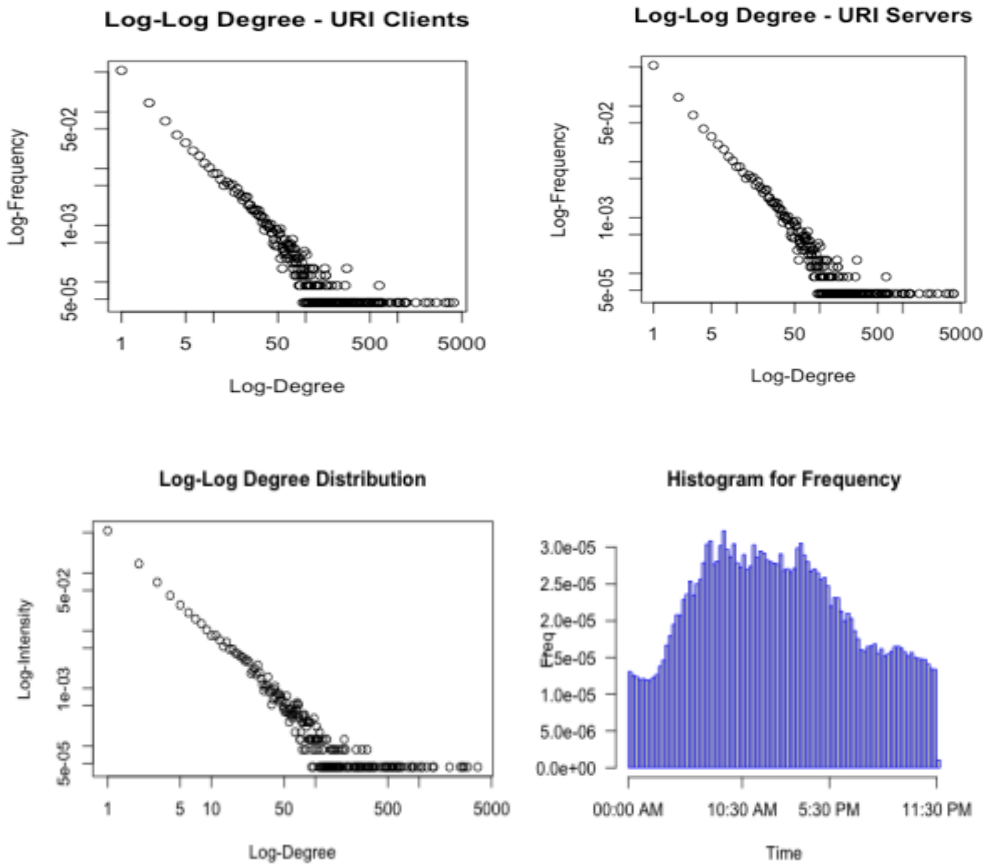


*Figure 8: Log-log plot of URI clients (left top), URI servers (right top), cumulative node strength distribution (left bottom) and Histogram (right bottom) for external network.*

## 4.2 PROPAGATION OF INFECTION

In this section, we describe several experiments that explore the rate of infection of nodes to understand infection spread on the networks. To conduct these experiments, we consider the following networks: (1) only external network derived from network flow data, (2) only internal network with different percentage of total flows 10%, 25% and 50%, (3) combined network with external network and 10% internal network. The

rate of infection is calculated for each probability transmission ($p$) value set up equal to 0.1, 0.3 and 0.5. We initiated the infection with 100 randomly selected clients from the list of: (1) all unique clients, (2) unique URI clients, (3) unique clients from Canada, (4) unique clients from China, (5) unique clients from India and (6) unique clients from UK. To address a special case of unusual activity coming from Denmark, we also initiate the infection from a single node.

In case of the external network, URI clients are not connected to URI servers directly resulting in some unrealistic zero rates of infection (Table 4, first three columns). Specifically, when infection propagation is initiated from the URI clients, none of the URI servers are infected. Similarly, when infection starts with clients from different countries (outside of URI), URI clients are not affected. These results are unrealistic clearly supporting the need for the collection of real internal communication traffic in order to analyze the health of the university network overall.

When we initially infect the network with 100 clients from each of the five countries mentioned in Table 4, the rates of infection for all types of nodes are almost equal with expected variability less than 5%. We started the experiment with the hypothesis that initiating infection with 100 clients of countries with history of attacks would target more important URI nodes and promote the spread of infection, whereas infection initiation with 100 random clients would not target any particular node of interest. The results summarized in Table 4 for India, China, and UK clearly do not support this hypothesis. Overall, the results demonstrate the higher rates of infection for client nodes compared to servers with maximum rates achieved when infection initiated from random nodes. At the same time, the results show that very similar rates when

infection is initiated from 100 different clients from each of selected countries (e.g., China, UK) and from one most active node from Denmark. In the further experiments, we plan to investigate the infection rates when propagation starts with most/least active clients/servers and also vary the number/proportion of nodes to start with.

| Experiments initiated with 100 URI Clients on | Fraction of Infected | P=0.1 | P=0.3 | P=0.5 |
|---|---|---|---|---|
| 10% Internal Network | URI Servers | 0.1957 | 0.3424 | 0.4239 |
| | URI Clients | 0.1601 | 0.3326 | 0.4484 |
| 25% Internal Network | URI Servers | 0.3369 | 0.5326 | 0.6087 |
| | URI Clients | 0.3035 | 0.5330 | 0.6491 |
| 50% Internal Network | URI Servers | 0.4674 | 0.7283 | 0.8261 |
| | URI Clients | 0.4497 | 0.6770 | 0.7687 |

*Table 3: Fraction of Infected Nodes Computed on Internal Networks (10%, 25% & 50%) for Different Transmission Rates (p = {0.1, 0.3, 0.5})*

| Experiments | Fraction of Infected | External Network | | | Combined Network (Ex+10% Int) | | |
|---|---|---|---|---|---|---|---|
| | | P=0.1 | P=0.3 | P=0.5 | P=0.1 | P=0.3 | P=0.5 |
| Random Clients | Servers | 0.1430 | 0.3989 | 0.5887 | 0.1698 | 0.4365 | 0.6226 |
| | Clients | 0.1648 | 0.4193 | 0.5909 | 0.2127 | 0.4618 | 0.6394 |
| | URI Servers | 0.1576 | 0.2283 | 0.2663 | 0.2391 | 0.3804 | 0.4728 |
| | URI Clients | 0.3444 | 0.6813 | 0.8055 | 0.4134 | 0.7231 | 0.8386 |
| URI Clients | Servers | 0.1423 | 0.4006 | 0.5817 | 0.1731 | 0.4353 | 0.6228 |
| | Clients | 0.0908 | 0.1807 | 0.2118 | 0.2116 | 0.4604 | 0.6402 |
| | URI Servers | 0.0 | 0.0 | 0.0 | 0.2446 | 0.3750 | 0.4674 |
| | URI Clients | 0.3450 | 0.6863 | 0.8046 | 0.4144 | 0.7225 | 0.8386 |
| Canada Clients | Servers | 0.0029 | 0.0039 | 0.0049 | 0.1425 | 0.4187 | 0.6178 |
| | Clients | 0.0866 | 0.2437 | 0.3816 | 0.1844 | 0.4415 | 0.6316 |
| | URI Servers | 0.1685 | 0.2283 | 0.2880 | 0.2228 | 0.3478 | 0.4620 |
| | URI Clients | 0.0 | 0.0 | 0.0 | 0.3490 | 0.7095 | 0.8356 |
| China Clients | Servers | 0.0028 | 0.0041 | 0.0049 | 0.1564 | 0.4250 | 0.6181 |
| | Clients | 0.0945 | 0.2476 | 0.3804 | 0.1997 | 0.4536 | 0.6332 |
| | URI Servers | 0.1630 | 0.2391 | 0.2880 | 0.2337 | 0.3641 | 0.4674 |
| | URI Clients | 0.0 | 0.0 | 0.0 | 0.3784 | 0.7160 | 0.8356 |
| India Clients | Servers | 0.0030 | 0.0039 | 0.0048 | 0.1612 | 0.4299 | 0.6199 |
| | Clients | 0.0976 | 0.2548 | 0.3910 | 0.2105 | 0.4658 | 0.6412 |
| | URI Servers | 0.1739 | 0.2283 | 0.2772 | 0.2283 | 0.3641 | 0.4620 |
| | URI Clients | 0.00 | 0.00 | 0.00 | 0.3893 | 0.7185 | 0.8377 |
| UK Clients | Servers | 0.0028 | 0.0038 | 0.0048 | 0.1255 | 0.4187 | 0.6139 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Clients | 0.0874 | 0.2320 | 0.3803 | 0.1685 | 0.4427 | 0.6226 |
| | URI Servers | 0.1630 | 0.2228 | 0.2772 | 0.2011 | 0.3478 | 0.4511 |
| | URI Clients | 0.00 | 0.00 | 0.00 | 0.3112 | 0.7095 | 0.8318 |
| Denmark Client | Servers | 0.0027 | 0.0041 | 0.0051 | 0.1500 | 0.4336 | 0.6197 |
| | Clients | 0.0845 | 0.2358 | 0.3733 | 0.1984 | 0.4598 | 0.6259 |
| | URI Servers | 0.1576 | 0.2391 | 0.2989 | 0.2446 | 0.4402 | 0.5435 |
| | URI Clients | 0.00 | 0.00 | 0.00 | 0.3812 | 0.7135 | 0.8309 |

*Table 4: Fraction of Infected Nodes Computed on Network inferred from External Traffic and Combined External and Internal Traffic (10%) for Different Transmission Rates (p = {0.1, 0.3, 0.5})*

We have also conducted experiments on combined network by infecting 100 randomly selected servers from the list of: (1) all unique servers and (2) unique URI servers sampled from total number of servers. The Table 5 presents the results of rates of infection. When we compare results in Table 4 and Table 5, the higher rates of infection are achieved when infection initiated from client nodes compared to servers. These results are clearly unrealistic, so we conducted further rate analysis experiments with client initiated infection.

| Experiments | Fraction of Infected | P=0.1 | P=0.3 | P=0.5 |
|---|---|---|---|---|
| Random Servers | Servers | 0.1447 | 0.4285 | 0.6189 |
| | Clients | 0.1797 | 0.4398 | 0.6002 |
| | URI Servers | 0.2283 | 0.4130 | 0.8191 |
| | URI Clients | 0.3404 | 0.7055 | 0.8191 |
| URI Servers | Servers | 0.1761 | 0.4395 | 0.6301 |
| | Clients | 0.2147 | 0.4706 | 0.6455 |
| | URI Servers | 0.6359 | 0.7554 | 0.7717 |
| | URI Clients | 0.4066 | 0.8315 | 0.7179 |

*Table 5: Fraction of Infected Nodes Computed on Combined Network for Different Transmission Rates (p = {0.1, 0.3, 0.5}) when infection initiated with 100 Servers.*

While Table 4 estimates presented for a single iteration of propagation, we have also analyzed the variation of our estimates in the network when infection propagation is simulated over 100 iterations on the external and combined networks. The resulting

rates of infection on one fixed internal network estimated over multiple iterations with p = 0.1 showed less than 1% of variability for servers and clients and less than 2% of variability for URI servers and URI clients, respectively. Mean and Standard Deviation of rates of infection on combined network are presented in Table 6. We observed comparable variability when conducted analysis on combined network with external network and variable 10% internal network over multiple interactions.

| Fraction of Infected | 100 Random Clients | 100 URI Clients |
|---|---|---|
| Servers | 0.1542 (0.009) | 0.1575 (0.004) |
| Clients | 0.2007 (0.010) | 0.2026 (0.005) |
| URI Servers | 0.2518 (0.018) | 0.2443 (0.019) |
| URI Clients | 0.3841 (0.019) | 0.3904 (0.010) |

*Table 6: Mean and Standard Deviation (in parenthesis) of fraction of infected nodes on Combined Network with external and 10% internal traffic for p=0.1.*

We have also performed analysis on combined networks with (1) external and 25% internal network and (2) external and 50% internal network. However, we did not see any abnormality in the results (shown in Table 7) and hence chose to consider 10% internal traffic to perform analysis on propagation of infection over time in Section 4.3.

| Experiments | Fraction of Infected | Combined Network (Ex+25% In) | | | Combined Network (Ex+50% Int) | | |
|---|---|---|---|---|---|---|---|
| | | P=0.1 | P=0.3 | P=0.5 | P=0.1 | P=0.3 | P=0.5 |
| Random Clients | Servers | **0.1906** | **0.4566** | **0.6406** | **0.2105** | **0.4725** | **0.6535** |
| | Clients | 0.2333 | 0.4772 | 0.6518 | 0.2582 | 0.4919 | 0.6617 |
| | URI Servers | 0.3696 | 0.5543 | 0.6304 | 0.4783 | 0.7391 | 0.8261 |
| | URI Clients | 0.4809 | 0.7618 | 0.8687 | 0.5670 | 0.8064 | 0.8950 |
| URI Clients | Servers | **0.1929** | **0.4571** | **0.6404** | **0.2114** | **0.4729** | **0.6534** |
| | Clients | 0.2342 | 0.4763 | 0.6520 | 0.2574 | 0.4921 | 0.6619 |
| | URI Servers | 0.3804 | 0.5652 | 0.6250 | 0.4837 | 0.7446 | 0.8261 |
| | URI Clients | 0.4834 | 0.7628 | 0.8687 | 0.5698 | 0.8077 | 0.8947 |
| Canada Clients | Servers | **0.1789** | **0.4494** | **0.6356** | **0.2010** | **0.4690** | **0.6520** |
| | Clients | 0.2223 | 0.4710 | 0.6451 | 0.2492 | 0.4854 | 0.6621 |
| | URI Servers | 0.3587 | 0.5543 | 0.6250 | 0.4565 | 0.7391 | 0.8261 |
| | URI Clients | 0.4540 | 0.7572 | 0.8650 | 0.5482 | 0.8040 | 0.8950 |
| China Clients | Servers | **0.1837** | **0.4514** | **0.6353** | **0.2065** | **0.4705** | **0.6495** |
| | Clients | 0.2305 | 0.4744 | 0.6440 | 0.2565 | 0.4883 | 0.6539 |
| | URI Servers | 0.3641 | 0.5598 | 0.6304 | 0.4837 | 0.7391 | 0.8261 |
| | URI Clients | 0.4683 | 0.7578 | 0.8634 | 0.5599 | 0.8049 | 0.8935 |
| India Clients | Servers | **0.1846** | **0.4538** | **0.6376** | **0.2071** | **0.4705** | **0.6511** |
| | Clients | 0.2360 | 0.4815 | 0.6555 | 0.2613 | 0.4938 | 0.6632 |
| | URI Servers | 0.3696 | 0.5598 | 0.6250 | 0.4783 | 0.7391 | 0.8261 |
| | URI Clients | 0.4695 | 0.7603 | 0.8675 | 0.5599 | 0.8058 | 0.8947 |
| German Clients | Servers | **0.1833** | **0.4495** | **0.6346** | **0.2063** | **0.4707** | **0.6504** |
| | Clients | 0.2277 | 0.4688 | 0.6378 | 0.2562 | 0.4870 | 0.6549 |
| | URI Servers | 0.3641 | 0.5598 | 0.6359 | 0.4837 | 0.7446 | 0.8370 |
| | URI Clients | 0.4667 | 0.7575 | 0.8631 | 0.5593 | 0.8055 | 0.8947 |
| UK Clients | Servers | **0.1801** | **0.4447** | **0.6356** | **0.2033** | **0.4655** | **0.6496** |
| | Clients | 0.2244 | 0.4598 | 0.6459 | 0.2523 | 0.4738 | 0.6555 |
| | URI Servers | 0.3587 | 0.5489 | 0.6250 | 0.4674 | 0.7228 | 0.8261 |
| | URI Clients | 0.4556 | 0.7522 | 0.8650 | 0.5540 | 0.8002 | 0.8935 |

*Table 7: Fraction of Infected Nodes Computed on Network inferred from Combined External and Internal Traffic (25% and 50%) for Different Transmission Rates (p = {0.1, 0.3, 0.5})*

## 4.3 EFFECTS OF TIME

In this section, we compute rates of infection over a period of time. In the quest to understand how rates of infection change over time, we conduct experiments using the 90-day data. Firstly, the graphs in Figure 9 depict the analysis of the dataset over the

week from 02-10-2014 to 02-16-2014 when probability of transmission is 0.1. The chosen week is randomly selected from the 90-day data. The experiments are conducted on combined network with external network and 10% internal network to understand the infection spread by initiating infection with: (1) 100 unique random clients, (2) 100 unique URI clients, (3) 100 unique China clients, and (4) 100 unique UK clients. The high number of data flows from China and UK could be explained by the significant number of students from China at URI and by URI's collaboration with universities at UK. The results demonstrate rates of infection of URI clients higher than rates of URI servers. The days 02-15-2014 and 02-16-2014 are weekend and hence show less rates of infection, relative to the intensity of traffic. The graphs display expected pattern of rates of servers, clients and URI servers over the week when infection is initiated from four different sources. But rates of URI clients vary based on initiated source of infection. This analysis was not evident when rates were calculated for one day in Section 4.2. This particular observation has led us to the following hypothesis: that URI clients are more vulnerable to infection from various sources in this experiment. However, we would need the real internal communication traffic in order to analyze the behavior and vulnerability of URI clients.


*Figure 9: Fraction of Infected Nodes on Combined Network: External + 10% Internal over the week between 02/10/2014 to 02/16/14 when p=0.1*

To understand further how the daily rates of infection change over time after an initial infection, we conduct experiments using the data collected over ninety-day period between February and May 2014. Figure 3 demonstrates the average activity of network nodes summarized separately for URI clients and servers. Figure 10 represents rates of infection estimated daily over ninety-day period. By comparing Figure 3 and Figure 10, one can notice that up until the middle of March, the estimated rates of infection followed the temporal weekly pattern consistent somewhat with the intensity of the traffic. For example, the fall in traffic intensity between 03/10/2014 to 03/14/2014 that can be explained due to spring break week at the university can be also observed in the estimated rates of infection. During the time period between 4/22/2014 to 5/6/2014 the

URI network was claimed to be under real cyber-attack explaining why data was not collected during this period. Remarkably, the intensity of node activity after the spring break and before the attack has not indicated any suspicious pattern; however, at the same time, the rates of infection for URI servers show clear departure from the expected behavior (Figure 10). This particular observation has led us to the following hypothesis: that the simulated infection rates that are not consistent with the intensity of the flow traffic may indicate the presence of compromised node activity and possible intrusion. The dependency that caused the abnormality could be hidden under certain characteristics of dynamic network that needs to be explored further (Refer Section 4.4).

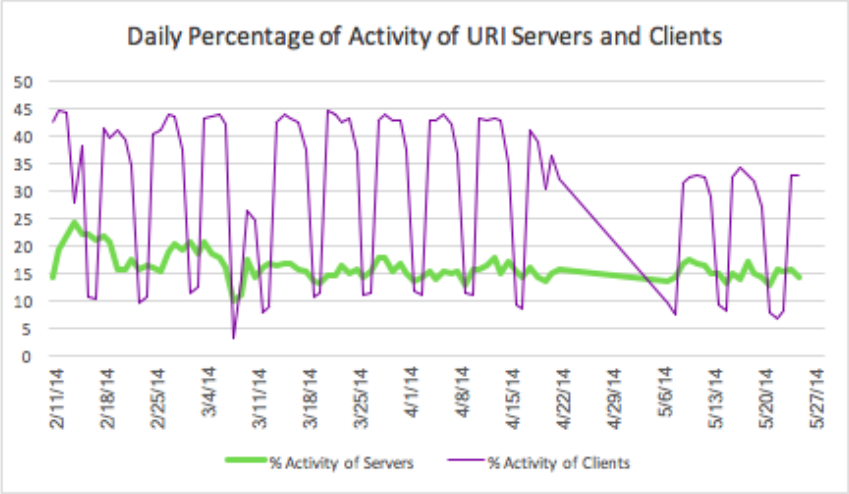


*Figure 3: Daily Percentage of Activity of URI Servers and Clients.*

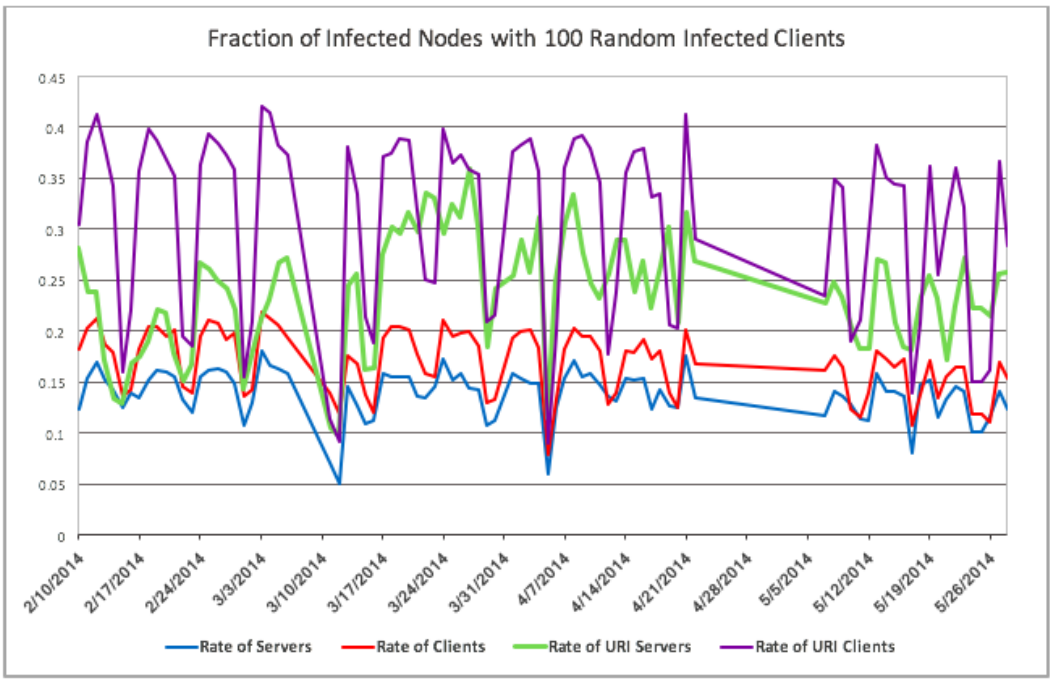


*Figure 10: Fraction of Infected Nodes per day on Combined Network over the 90-day period.*

We have analyzed average of fraction of infected nodes per day in the week over 90-days and Figure 11 depicts the results. The weekends show less activity and week days, specially the mid-week Thursdays show maximum rates of infection. On an average, the rates of infection show proportional pattern to intensity of traffic and grand average states that URI clients are more proven to infection at 31%, URI servers at 24%, overall clients at 18% and servers at 15%.

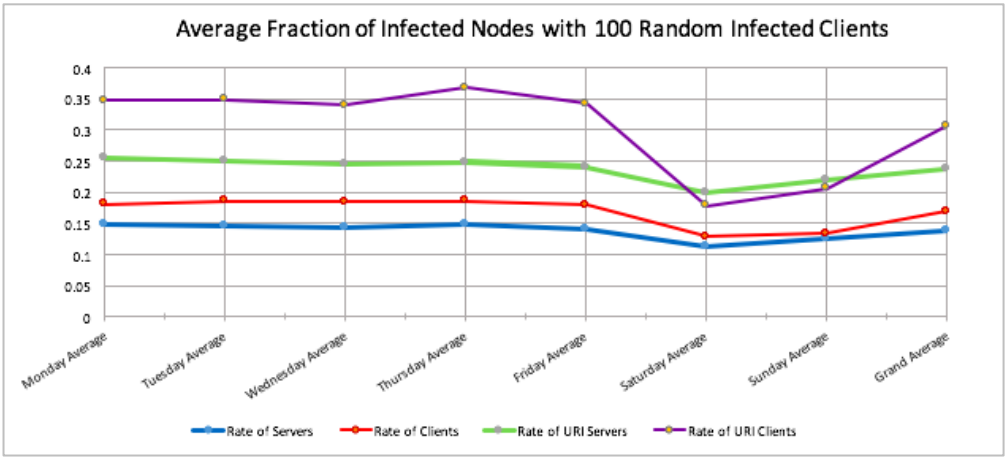


*Figure 11: Average of Fraction of Infected nodes on Combined Network for each day in the week over 90-days when p=0.1 and initial infection starting from 100 Random clients.*

We have further analyzed infection propagation per week over the 90-day period. This analysis helps us to understand the infection spread, when infection propagation continues in the network through the week. The results demonstrate rates of infection of URI clients higher than rates of URI servers. In Figure 10, we could see normal activity with respect to rate of infection of servers, clients and URI clients, whereas the rate of infection of URI servers showed abnormal high during the 3/17/2014 and 4/20/2014. Similar results can be seen in Figure 12 for weekly propagation analysis.

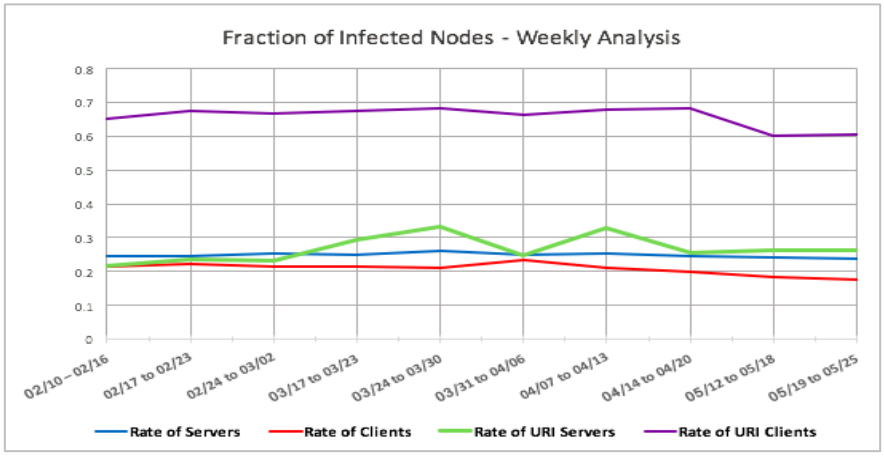


*Figure 12: Fraction of Infected nodes on Combined Network for infection spread weekly over the 90-day period, when p=0.1 and initial infection starting from 100 Random clients.*

## 4.4 NETWORK CHARACTERISTICS OVER TIME

We have focused on the network characteristics of internal traffic (10%) and combined network per day over the time period of 90 days in this section. Figure 13 depicts the unique number of URI clients and URI servers in internal traffic and Figure 14 represents the unique number of clients, servers, URI clients and URI servers in the combined network. However, the pattern of the number of nodes over the period is consistent with the intensity of the flow traffic.

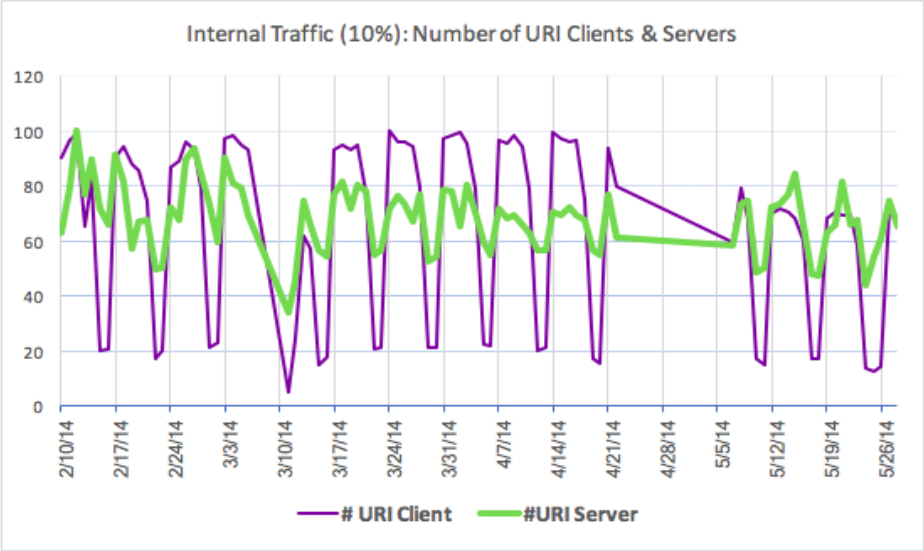


*Figure 13: Number of URI Clients and Servers in Internal Network (10%).*

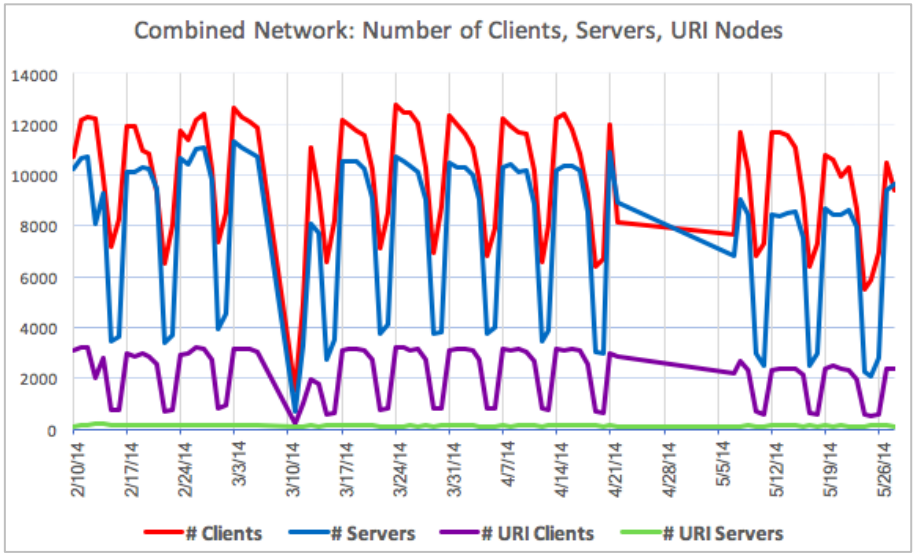


*Figure 14: Number of Clients, Servers, URI Clients & URI Servers in Combined Network.*

*Degree of nodes:* The degree of URI clients and URI servers in internal traffic (Figure 15) and degree of clients and servers in combined network (Figure 16) show a similar pattern to that of the intensity of the flow traffic, which makes it hard to predict the abnormality and dependency. Our initial findings show that internal traffic preserves the node degree and time pattern.

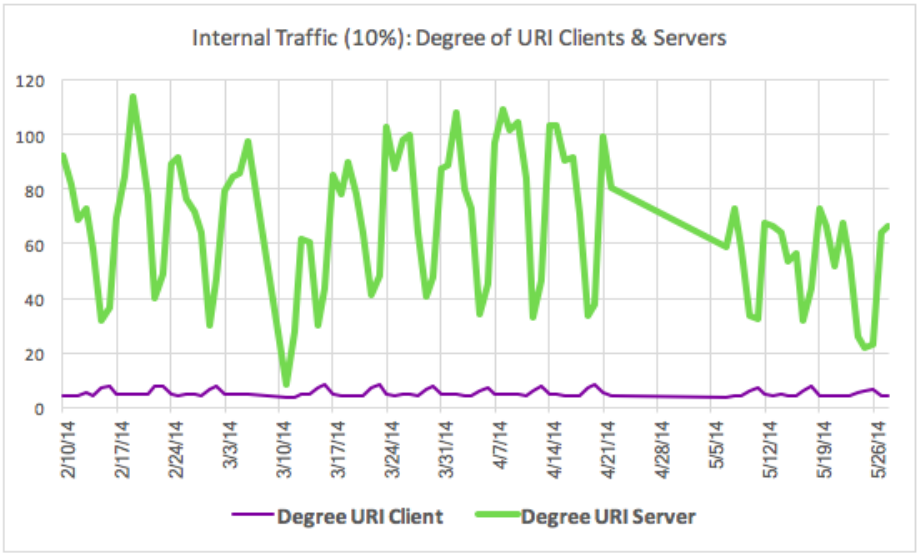


*Figure 15: Degree of URI Clients and Servers in Internal Network (10%).*

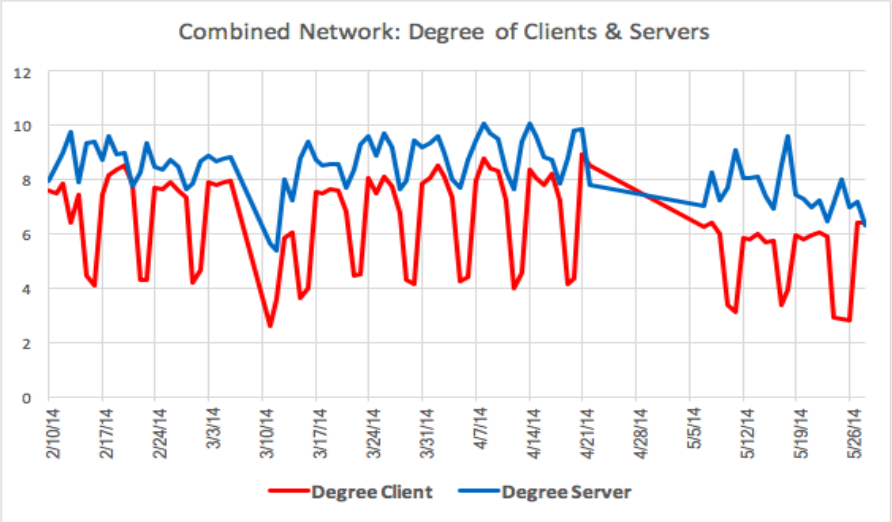


*Figure 16: Degree of Clients and Servers in Combined Network.*

*Assortativity:* Assortativity is a preference of nodes to attach to others that are similar in the network. Though the specific measure of similarity may vary, network theorists often examine assortativity in terms of a node's degree [1]. We can notice that the degree distribution of clients and servers (Figure 15, 16) followed the temporal weekly pattern consistent with the intensity of the traffic. But the assortativity pattern (Figure 17, 18) in internal and combined network shows high variability and an interesting pattern that is not consistent with the intensity of flow traffic.

Technological and biological networks typically show disassortative mixing, or dissortativity, where high degree nodes tend to attach to low degree nodes [1]. Our initial findings validate that our University network is disassortative in nature, which can be explained by expected selective communication behavior pattern and heavy tailed distribution of nodes (Figure 8). But Figure 18 clearly depicts positive values of assortativity, making the network random or assortative. Assortativity of zero value indicates close to random connectivity, which is unusual for a University network.

Positive assortativity is even more unusual, as it would imply communication only between popular URI servers and very active clients. Remarkably, the intensity of node activity after the spring break and before the attack has not indicated any suspicious pattern; however, at the same time, the network structure alters and pattern of assortativity shows clear departure from the expected behavior (Figure 18). This particular observation may indicate the presence of compromised node activity and promising future direction to predict possible intrusion.

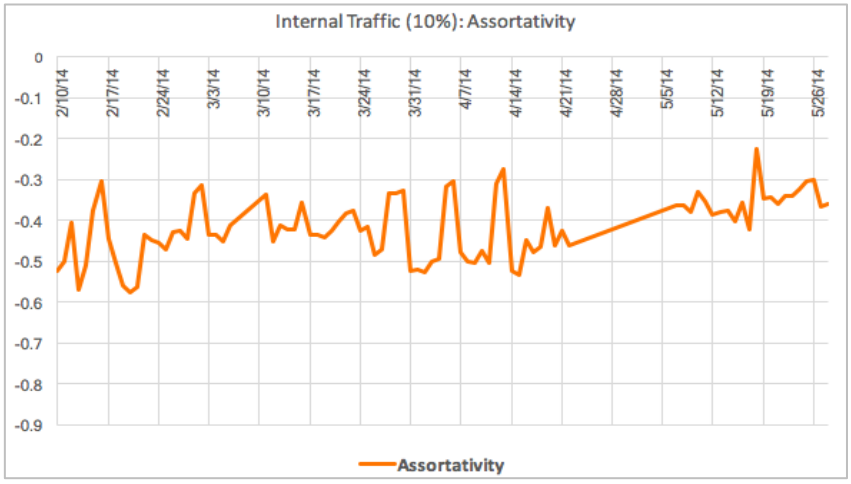


*Figure 17: Assotativity of Internal Network (10%).*

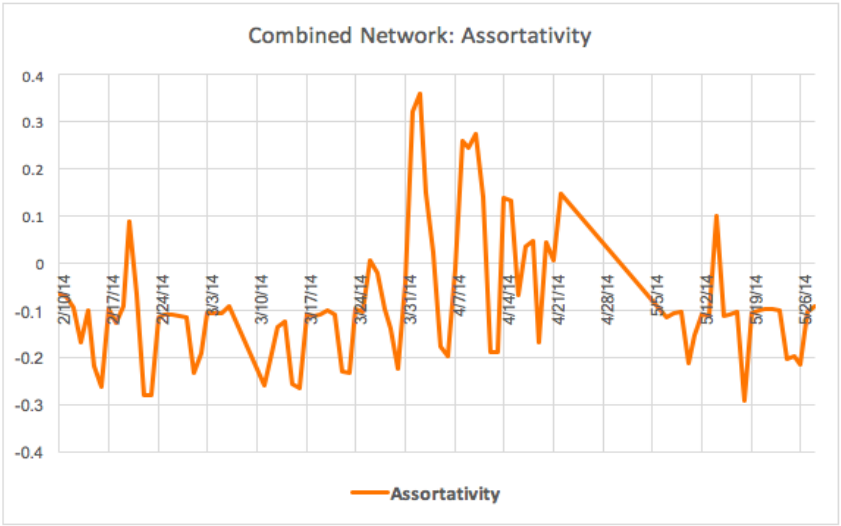


*Figure 18: Assortativity of Combined Network.*

*Clustering coefficient of nodes:* The clustering coefficient is another property which is dependent on how the nodes are related in the network and based on the projection of network. The below graphs in Figure 19 and 20 depict global and local clustering coefficient of internal and combined networks, respectively. The local clustering coefficient of nodes shows a similar pattern to that of the intensity of the flow traffic, but the global clustering coefficient of servers shows interesting pattern that is not consistent with the traffic flow intensity. This may indicate compromised node activity and needs to be further investigated.

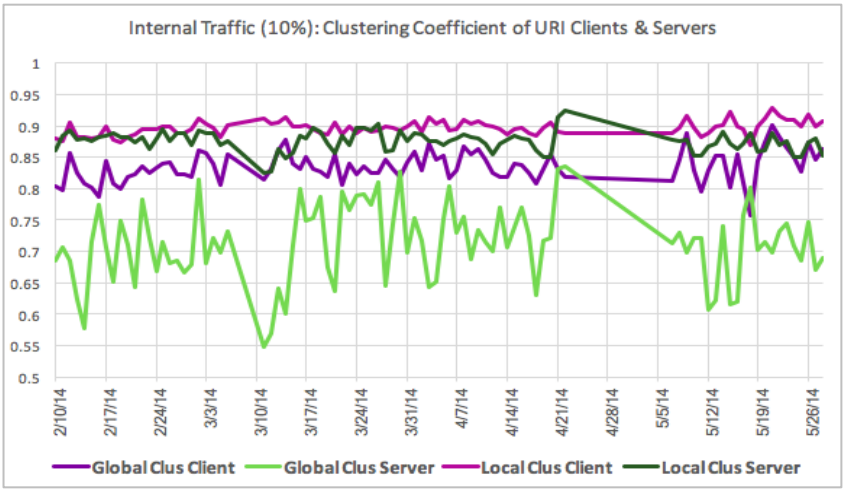


*Figure 19: Global and Local Clustering Coefficient of URI Clients & Servers in Internal Network.*

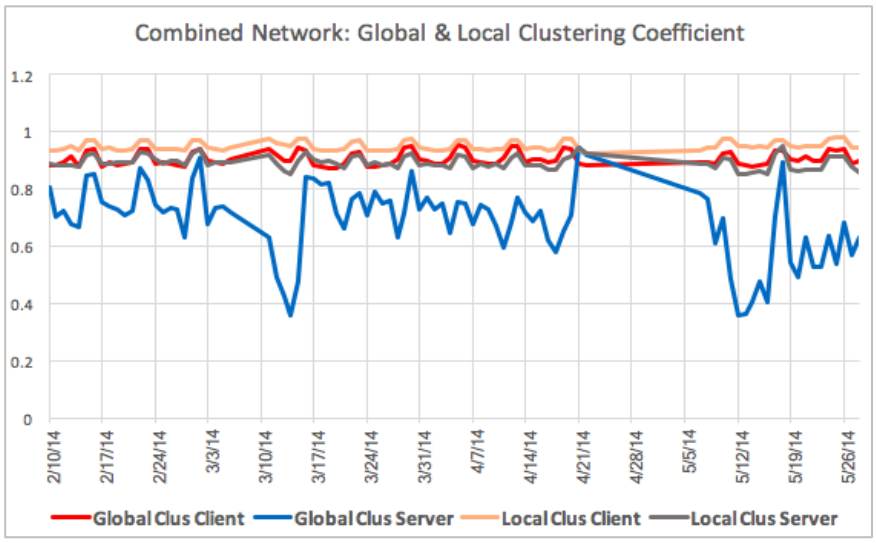


*Figure 20: Global and Local Clustering Coefficient of Clients and Servers in Combined Network.*

## List of References

[1] Newman, M.E.J, “Mixing patterns in networks”, Phys. Rev. E 67, 2003

# CHAPTER 5

# CONCLUSION

In summary, this study presents epidemiological study of Web-based malware for university network with partially observed flow data. It offers a framework that helps to (a) represent the network flow data in a form of a bipartite graph, (b) model an internal university network traffic from the observed external flow data using with limited information and a set of simple assumptions; and (c) analyze the spread of infection with and with-out the simulated internal network. The proposed methodology confirms the fact that rates of infection are incomplete without internal network and motivates the collection of real internal university data traffic in future. Additionally, we introduced country based infection simulation for detection of university network behavior over infection flowing in from different countries with history of cyber-attacks. The rates of infection, however, on the network proved to be similar when infection starts with random clients of these countries. Overall, the results suggest higher rates of infection for client nodes compared to servers with maximum rates achieved when infection initiated randomly. At the same time, the results of a simulation experiment when infection starts from one active node from Denmark inspires further analysis in this direction. In addition, the daily analysis over a three-month period reveals that the simulated infection rates that are not consistent with the intensity of the traffic and the pattern of network characteristics which are dependent on how the nodes are related in the network, such as assortativity and global clustering coefficient, may

indicate the presence of compromised node activity and possible intrusion. This finding would serve as promising future course of research.

Finally, in this paper, we have considered the same probability of infection for clients and servers over time, whereas in reality clients and servers have variable protection and susceptibility levels, and classification of different nodes with different probabilities of infection is also in the future scope of research. Looking at some of the research works on disease spread and vaccination, the ideology of active and passive spread could be introduced to enhance the research further. Also, we intend to develop a tool to help observe the epidemic spread in real time as a future course of the study.

# APPENDICES

## 6.1 APPENDIX A: R CODE.

```
# 1. Extracting all Servers and Clients
serverList <- c ()
clientList <- c ()

## Read excel CSV: example dataset 03/27/2014
flow ← read.csv ("flows-131.128.5.84-2014-03-27.csv", header=T, as.is=T, sep=",",
row.names = NULL)[,c(1,5,6,8,14,15)]
IPAddress <- subset (flow, flow$application == "web-browsing")

## Extract serverlist and clientlist
serverList <- IPAddress$destination_address
clientList <- IPAddress$source_address

## Total number of unique clients and servers
clienttotal <- length(unique(clientList))
servertotal <- length(unique(serverList))

URIcltotal <- length (unique (clientList [grepl("^131.128", clientList)]))
URIsrtotal <- length (unique (serverList [grepl("^131.128", serverList)]))

## Generate a vector of random variables on [0,1] interval, the same length as the
number of flows -Probability of infection transmission
xinfectedList <- runif(nrow(IPAddress), min = 0, max = 1)

## Create dataframe with - determined Client and Server List & Receive Time
Comp_Net_Flow <- data.frame (ClientsIP = clientList,
                    ServersIP = serverList,
                    receive_time = IPAddress$receive_time,
                    rxif = xinfectedList)

## 2. Construction of Internal Network

##Extract URI servers and clients using IP address starting with 131.128.X.X
URI_clients <- clientList [grepl("^131.128", clientList)]
URI_servers <- serverList [grepl("^131.128", serverList)]
```

```
## Consider all unique URI clients and servers; Sampling not required.
URIclienttotal <- length (URI_clients)
URIservertotal <- length (URI_servers)

## Calculate the size (total data flows) for each Internal network 50%, 25% and 10%

Random_10percent <- round (0.1 * nrow(Comp_Net_Flow))
Random_25percent <- round (0.25 * nrow(Comp_Net_Flow))
Random_50percent <- round (0.50 * nrow(Comp_Net_Flow))
xinfectedList50 <- runif (Random_50percent, min = 0, max = 1)

## Create 10% and 25% out of 50% of the data flows.
Random_50_Flow <- data.frame
                (ClientsIP = sample (URI_clients, Random_50percent, replace=T),
                 ServersIP = sample (URI_servers, Random_50percent, replace=T),
                 receive_time = sample (IPAddress$receive_time, Random_50percent),
                 rxif = xinfectedList50)

Random_25_Flow <- Random_50_Flow [
                        sample(nrow(Random_50_Flow), Random_25percent), ]
Random_10_Flow <- Random_25_Flow [
                        sample(nrow(Random_25_Flow), Random_10percent), ]

## For External network: timestamp * 10 + vector(i) 1 to length
Comp_Net_Flow$receive_time <- as.numeric
                (Comp_Net_Flow$receive_time) * 10+c (1:nrow (Comp_Net_Flow))

## For Internal network: timestamp * 10 + vector(i) 1 to length + 0.5
Random_10_Flow$receive_time <- as.numeric
        (Random_10_Flow$receive_time) * 10+c (1:nrow (Random_10_Flow))+0.5
Random_25_Flow$receive_time <- as.numeric
        (Random_25_Flow$receive_time) * 10+c (1:nrow (Random_25_Flow))+0.5
Random_50_Flow$receive_time <- as.numeric
        (Random_50_Flow$receive_time) * 10+c (1:nrow (Random_50_Flow))+0.5

## Complete list - adding external flow and internal flow
web_flow_dly <- rbind.data.frame(Comp_Net_Flow,Random_10_Flow)

## Sort(order) the data frame based on timestamp.
```

```
web_flow_dly <- web_flow_dly[order(web_flow_dly$receive_time),]

## Total length of the data frame.
nrow(web_flow_dly)

#########################################################################

## 3. Infection Propagation and Rate of Infection

stochastic_infection <- function(p)
{
        infectedserverList <- c()
        infectedclientList <- c()
        infectedserverURIList <- c()
        infectedclientURIList <- c()

        ##counter variables nc[] and ns[] and for URI nodes ncuri[] and nsuri[]
        nc <- c()
        ns <- c()
        ncuri <- c()
        nsuri <- c()

        ##infect randomly 100 clients.
        #(i) Random 100 clients from netflow data
        #(ii) Random 100 URI clients
        #(iii) Random 100 Clients from Top Countries -- Canada China India UK

          infectedclientList <-unique(clientList)
          [sample(length(unique(clientList)),100, replace=FALSE)]

          infectedclientList <- unique(URI_clients)
          [sample(length(unique(URI_clients)),100, replace=FALSE)]

         countrylist <- (subset (IPAddress, (IPAddress$destination_port == 80 |
        IPAddress$destination_port == 443) & (IPAddress$source_country ==
        "China")))
         infectedclientList <- unique(countrylist$source_address)
        [sample(length(unique(countrylist$source_address)),100,replace=)]

        ##loop result: infected clients and server list
        for (i in 1:nrow (web_flow_dly)){
```

```
  if (is.element(web_flow_dly$ClientsIP[i], infectedclientList))
  {
    if(web_flow_dly$rxif[i] <= p) {
      infectedserverList <- union (infectedserverList, web_flow_dly$ServersIP[i])
    }
  }
  if (is.element (web_flow_dly$ServersIP[i], infectedserverList))
  {
    if(web_flow_dly$rxif[i] <= p) {
      infectedclientList <- union (infectedclientList, web_flow_dly$ClientsIP[i])
    }
  }

  nc[i] <- length(infectedclientList)
  ns[i] <- length(infectedserverList)
  ncuri[i] <- length (infectedclientList[grepl("^131.128", infectedclientList)])
  nsuri[i] <- length (infectedserverList[grepl("^131.128", infectedserverList)])

  URI_infclients <- infectedclientList[grepl("^131.128", infectedclientList)]
  URI_infservers <- infectedserverList[grepl("^131.128", infectedserverList)]

 }
  return (list (infectedserverList, infectedclientList, ns/servertotal, nc/clienttotal,
  nsuri/URIsrtotal, ncuri/URIcltotal, URI_infservers, URI_infclients))
}
```

## 4. Parallel Computation using doparallel and foreach

```
library(doParallel)
library(foreach)

## Numbers of cores in the system
no_cores <- detectCores()
## Number of cluster - increase more than available cores or same.
cl<-makeCluster(no_cores-1)
## To make the process parallel
registerDoParallel(cl)
```

## Calling function in parallel for each of p based on different datasets

```
totalp <- foreach (p = c(0.1, 0.3, 0.5)) %dopar% stochastic_infection(p)
stopCluster(cl)

## Rate of Infection Calculation for total clients and servers
rateinf_clients_0.1 = length(totalp[[1]][[2]])/clienttotal
rateinf_servers_0.1 = length(totalp[[1]][[1]])/servertotal
rateinf_clients_0.3 = length(totalp[[2]][[2]])/clienttotal
rateinf_servers_0.3 = length(totalp[[2]][[1]])/servertotal
rateinf_clients_0.5 = length(totalp[[3]][[2]])/clienttotal
rateinf_servers_0.5 = length(totalp[[3]][[1]])/servertotal

## Rate of Infection Calculation for URI clients and servers
URIrateinf_clients_0.1 = length(totalp[[1]][[8]])/URIcltotal
URIrateinf_servers_0.1 = length(totalp[[1]][[7]])/URIsrtotal
URIrateinf_clients_0.3 = length(totalp[[2]][[8]])/URIcltotal
URIrateinf_servers_0.3 = length(totalp[[2]][[7]])/URIsrtotal
URIrateinf_clients_0.5 = length(totalp[[3]][[8]])/URIcltotal
URIrateinf_servers_0.5 = length(totalp[[3]][[7]])/URIsrtotal

#########################################################################

##5. Network Characteristics

# Create bipartite graph to explore characteristics
IPTableuni = table(Random_Intr_Flow$ServersIP, Random_Intr_Flow$ClientsIP)
IPuni = cbind(IPTableuni)
graphuni <- as.matrix(IPuni)
datauri <- graph.incidence(graphuni,multiple=T, mode= c("all", "out", "in", "total"))

# Average degree of nodes
datauri.degree = degree(datauri)
mean(datauri.degree[V(datauri)$type==T]) #Clients
mean(datauri.degree[V(datauri)$type==F]) #Servers

# Assortativity of network
assortativity_degree(datauri)

# Log-log plot
dd.data<-degree.distribution(datat)
d<-1:max(datat.degree)-1
ind<-(dd.data!=0)
```

```
plot(d[ind],dd.data[ind],log="xy",col="black", xlab=c("Log-Degree"), ylab=c("Log-Frequency"), main="Log-Log Degree Distribution")
# Histogram
hr <- as.numeric(Comp_Net_Flow$receive_time)
hist(hr, main="Histogram for Frequency", xlab="Time", ylab="Freq",
border="blue", col="grey", las=1, xaxt="n", breaks=100, prob = TRUE)
axis(1, at=c(1,17000, 30000, 46000), labels= c("00:00 AM", "10:30 AM", "5:30 PM", "11:30 PM"))

# Bipartite projection
bi.proj <- bipartite.projection(datauri)
server.net <- bi.proj$proj1
client.net <- bi.proj$proj2

# Global Clustering coefficient
transitivity(client.net, type = "global") #Clients
transitivity(server.net, type = "global") #Servers

# Local Clustering coefficient
data.cl <- transitivity(client.net, type = "local") #Clients
mean(data.cl[which(data.cl != "NaN")])
data.cl <- transitivity(server.net, type = "local") #Servers
mean(data.cl[which(data.cl != "NaN")])
```

# BIBLIOGRAPHY


Barford, Paul, et al. "A signal analysis of network traffic anomalies." in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement, ACM,* 2002.

Bing, Chris, "Universities, not health care systems, facing highest number of ransomware attacks," *Fedscoop*, Sept. 21, 2016.

Bisanzio Donal, Bertolotti, Luigi, Tomassone, Laura, Amore, Giusi, Charlotte Ragagli, Alessandro Mannelli, Mario Giacobini, Paolo Provero, "Modeling the Spread of Vector-Borne Diseases on Bipartite Networks," *PLOS*, 2010.

Chang, Jian, Venkatasubramanian, Krishna K., West, Andrew G., Insup Lee, "Analyzing and Defending Against Web-based Malware", *ACM Survey*, 2013.

Gomez-Gardenes, Jesus, Latora, Vito, Moreno, Yamir and Elio, Profumo, "Spreading of sexually transmitted diseases in heterosexual populations," in Proceedings of National Academy of Sciences USA, 2008.

Harris E. Charles, Hammargren R. Laura, "Higher education's vulnerability to cyber-attacks," *University Business*, August 2016.

Katenka, N. (with Crowella, M., Kolaczyk, E., and Britton, T.), "Epidemiological Models for Browser-Based Malware", *Invited Poster Presentation, Eastern North American Region Conference (ENAR),* Baltimore, MD, 2014.

Keeling, J. Matt and Eames, T.D. Ken, "Networks and epidemic models," *J R Soc Interface,* Jun 2005.

Kolaczyk, E. D., “Statistical Analysis of Network Data: Methods and Models,” *Springer Ser. Stat.*, 2009.

Lakhina, Anukool, Crovella, Mark and Diot, Christophe, "Diagnosing net-work-wide traffic anomalies," in *Proceedings of ACM SIGCOMM Computer Communication Review. Vol. 34. No. 4. ACM*, 2004.

Lakhina, Anukool, Crovella, Mark and Diot, Christophe, "Characterization of network wide anomalies in traffic flows," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, ACM*, 2004.

Lakkaraju, Kiran, William Yurcik, and Adam J. Lee. "NVisionIP: netflow visualizations of system state for security situational awareness.", *in Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, ACM*, 2004.

Gordon, Max, “How to go parallel in R,” *G-Forge*, Feb 2015.

Moshchuk, Alexander, Bragin, Tanya, Gribble, D. Steven, and Levy, M. Henry, “A Crawler-based Study of Spyware on the Web,” in *Proceedings of the 2006 Net-work and Distributed System Security Symposium*, pages 17–33, Feb 2006.

Newman, Lily H., “The Biggest Cybersecurity Disasters of 2017 So far”, Security, Wired, Jul 2017.

Newman, M.E.J, “The spread of epidemic disease on networks”, *Center for the Study of Complex Systems, University of Michigan*, 2002.

Newman, M.E.J, “Mixing patterns in networks”, *Phys. Rev. E 67, 2003.*

Paxson, V., Adams, A. and Mathis M., "Experiences with NIMI", In *Proceedings of Passive/Active Measurement (PAM),* 2000.

Provos, N., D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The ghost in the browser analysis of web-based malware," in *Proceedings of the First Workshop on Hot Topics in Understanding Botnets*, 2007.

Savage, Stefan, Voelker Geoffrey, Varghese George, Vern Paxson, Nicholas Weaver, "NSF CyberTrust Center Proposal", *Center for Internet Epidemiology and Defenses*, 2004-2009.

Straub, M. Kayla, Sengupta, Avik, Ernst, M. Joseph, McGwier W. Robert, Merrick Watchorn, Tilley, Richard and Marchany, Randolph. "Malware Propagation in Fully Connected Networks: A Netflow-Based Analysis," *IEEE*, 2016.

Straub, M. Kayla, Sengupta, Avik, Ernst, M. Joseph, McGwier W. Robert, Merrick Watchorn, Tilley, Richard and Marchany, Randolph. "Malware Propagation in Fully Connected Networks: A Netflow-Based Analysis," *IEEE*, 2016.

Tarissan, Fabien, Bruno Quoitin, Pascal MéRindol, Benoit Donnet , Jean-Jacques Pansiot, Matthieu Latapy, "To-wards a Bipartite Graph Modeling of the Internet Topology," Computer Networks, 57(11), 2331-2347, August, 2013.

Wagstaff Keith, Sottile Chiarra, "Cyberattack 101: Why Hackers are Going After Universities," *NBC News*, Sept 20, 2015.

Weston, Steve and Calaway, Rich, "Getting Started with doParallel and foreach", Cran-R-Project, Oct 2015.

Yin, Xiaoxin, et al. "VisFlowConnect: netflow visualizations of link relationships for security situational awareness.", *in Proceedings of the 2004 ACM work-shop on Visualization and data mining for computer security*, *ACM*, 2004.

Yi-Min Wang, Doug Beck, Xuxian Jiang, Roussi Roussev, Chad Verbowski, Shuo Chen, and Sam King, "Automated Web Patrol with Strider HoneyMonkeys", in *Proceedings of the 2006 Network and Distributed System Security Symposium*, pages 35–49, 2006.