NCFTA Canada — National Cyber-Forensics and Training Alliance CANADA

DEFENCE R&D DÉFENSE

Industrie Canada Industry Canada

National Cyber Forensics and Training Alliance (NCFTA) Canada

In Collaboration with

DRDC Centre for Security Science and Industry Canada

Final Report

Analysis of Darknet Space for Predictive Indicators of Cyber Threat Activity

Request for Proposal No: U4408-118202/A
PWGSC File No: 06SS.U4408-118202
PSTP Project No: 02-34SESEC
File No: QCL-9-307S0 (028)
Supervised by: DR. M. DEBBABI AND DR. B. FUNG
Executed by: L. APIKIAN, H. BINSALLEEH, E. BOU-HARB, A. BOUKHTOUTA
F. IQBAL, S. DINH, C. FACHKHA AND S. MOKHOV

Contents

List of Figures

List of Tables

Acknowledgments

The research reported in this document is the result of a fruitful collaboration between NCFTA Canada, the Centre of Security Science at National Defence and Industry Canada under the Public Security Technical Program (PSTP), contract: Request for Proposal No: U4408-118202/A, PWGSC File No: 06ss.U4408-118202, PSTP Project No: 02-34SeSec and File No: QCL-9-307S0 (028).

Executive Summary

Today, the safety and security of our society is entirely dependent on having a secure critical infrastructure. This infrastructure is controlled and operated using cyberspace: a network of numerous interconnected computers. Recent events demonstrate that cyberspace could be subjected, at the speed of light and in full anonymity, to debilitating, intimidating and disrupting attacks that might lead to severe security and economic issues, and even to the endangerment and loss of human lives. These attacks could be carried out by a wide spectrum of individuals such as criminals, cyber-terrorists, terrorists and foreign governments. The existence of widely available encryption and anonymizing techniques also makes the surveillance and investigation of the attacks through cyberspace much harder than in traditional communication systems. In this context, cyber threat intelligence is of vital importance and at the same time poses many new challenges. A promising approach to gathering cyber threat intelligence is to collect and analyze traffic destined to unused Internet addresses known as darknets. Since these addresses do not correspond to any legitimate host or device, any observed traffic will fall into the following four categories: Propagation attempts from software worms or botnets, network scanning or probing, backscatter from spoofed IP addresses in denial-of-service (DoS) attacks, or mis-configurations. Understanding this kind of traffic may thus provide valuable information for detecting ongoing attacks and predicting future attack trends in order to facilitate both real-time and proactive defence of cyberspace. As such, there is a desideratum that consists in developing a capability for the analysis of darknets for the purpose of preventing, detecting, assessing, mitigating and attributing cyber attacks.

The primary intent of this study is to address the elaboration of such a capability through the study of darknets and the design and implementation of analysis/correlation modules that will transform darknet-related network information in conjunction with other malware/network data into precious cyber intelligence. The latter could be used in preventing, detecting, assessing, mitigating and attributing cyber attacks. Examples of such intelligence are:

- Darknet traffic characterization and profiling (e.g., sources, destinations, ISPs, protocols, ports, and vulnerabilities)
- Detection of darknet-triggered infections/intrusions and their corresponding sources and targets
- Detection, tracking and geo-localization of cybercrime servers
- Detection, tracking and geo-localization of depots/repositories of stolen information
- Detection of corporate/governmental brand abuses
- Identification and sizing of cyber attacks
- Forecasting and prediction using cyber intelligence

Defining Darknets

The term "darknet" encompasses several definitions, some of which are mentioned as follows:

- A set of unallocated computer network addresses and communication ports belonging either to the public cyberspace or to an organization. Such unallocated space could be maliciously utilized for initiating concealed communications or launching cyber attacks.
- An environment that provides communication anonymity. This is related to the task of achieving private communication as coined by Microsoft researchers [1, 2].
- Material that exists on the public cyberspace, which is untraceable and inaccessible to web services such as search engines.
- Peer-to-peer communication platforms including social networks, chat channels and file-sharing environments. Such platforms are generally encrypted, and thus their communication data is concealed.

Darknet Threats

Darknets could be utilized to detect cyber attacks with severe consequences. The following enumerates such possible attacks:

- Advanced persistent threats targeting critical cyber infrastructures and services triggered by organized, highly specialized cyber criminals and/or terrorists
- Disruption attacks causing severe denial-of-service issues specifically targeted towards e-commerce servers and corporate and governmental networks
- Attacks using covert channels to disclose sensitive information and distribute malware and offensive content (i.e., child exploitation)
- Zero-day attacks exploiting unknown vulnerabilities
- Attacks causing various brand abuses including phishing, brandjacking, counterfeiting, false association and piracy

Project Objectives

The objectives of this project are to:

- Define darknet techniques, tools and technologies and evaluate their readiness levels
- Identify areas that require research and estimate time horizons for each of them
- Characterize and profile darknets in terms of sources, destinations, ISPs, protocols, ports, vulnerabilities, etc.
- Determine the key components and characteristics of darknet information required to predict attacks on the Government of Canada networks
- Analyze prevalent trends, threats, and risks associated with darknets
- Design and implement a testbed for automatic cyber intelligence generation. Such intelligence is meant to be used in preventing, detecting, mitigating and attributing darknets-based attacks
- Provide recommendations on mitigating techniques together with strategic advice on the capability road-map of activities/initiatives that are meant to address the identified gaps

Report Overview

This document is meant to be a final report. It primarily defines darknets and their taxonomies. Moreover, it presents the state-of-the-art in terms of research contributions and techniques, tools and technologies. Furthermore, it identifies the gaps in terms of science and technology. Additionally, it presents our darknet analysis, our malware analysis and brand protection frameworks, in addition to time series analysis and threats correlation for the purpose of prediction and hence mitigation.

1 Introduction

Nowadays, Information and Communication Technologies (ICT) pervade almost every aspect of the socio-economical interactions prevalent in the dynamics of our society. Furthermore, the safety and security of our society depends on the so-called critical infrastructure, which spans over of public and private organizations in the sectors of government, defence, energy, telecommunications, public health, emergency services, agriculture, food, water, finance and transportation. This infrastructure is controlled and operated using ICT technologies. The latter are considered as the nervous system of our infrastructure as they control physical entities such as power grids, trains, aircrafts, energy pipelines, nuclear plants, radars and stock markets. Recent events demonstrated that individuals, corporations and governmental organizations could be subjected, at the speed of light and in full anonymity, to amplified, large-scale, debilitating, intimidating and disrupting attacks that might lead to severe privacy/security and economic consequences, and even to the endangerment and loss of human lives (e.g., child exploitation, cyber-terrorism, denial-of-service, information theft, hate crimes, defamation, bullying, identity theft and fraud). These attacks might be carried out by a spectrum of individuals such as criminals, cyber-terrorists, terrorists and foreign government spies. Moreover, as the closest approximation of perfect anarchy, the Internet becomes an attractive tool to terrorists for spreading messages, recruiting supporters, planning and coordinating attacks. In this context, it is a national duty of paramount importance to protect:

- The millions of Canadian citizens who are ICT users, especially that a sizable fraction corresponds to children, teenagers and seniors
- Public and private sector institutions that expose their assets, operations and services over the Internet
- Critical infrastructure (energy, transportation, healthcare, water distribution, etc.)

The cyberspace security challenge takes an allure of a continuous background conflict due to the fact that computer attack tools are more sophisticated and hackers are capable of launching worldwide impacting assaults. For instance, in July 2009, Damballa ranked Zeus botnet as the number one threat with 3.6 million infections in United States[1]. Zeus bots were estimated as responsible for 44% of banking malware infections [3]. Moreover, Symantec Corporation dubbed the Zeus crimeware toolkit as 'the King of the Underground Crimeware Toolkits' [4]. Another significant example is Mariposa botnet (Known also as Butterfly), which is a new generation botnet. It was claimed that 13 million machines got infected in 190 countries across the globe by this botnet once it appeared in May 2009 [5]. Mariposa bots are frequently altered to evade antivirus detection. They are able to download and execute malicious code on the fly, which makes the botnet extremely harmful. Mariposa can be associated with other botnets since it has the capability to infect machines with other bots. The icing on the cake was the Stuxnet malware. It was first seen by Sergey Ulasen who handed it to the Antivirus Division of a security company, namely, VirusBlokAda[2], which contacted Microsoft and announced the existence of Stuxnet. Further investigation done by Symantec engineers revealed that among 38,000 infections, about 22,000

[1] `http://blog.damballa.com/?p=569`
[2] `http://anti-virus.by/en/index.shtml`

were in Iran [6]. The reason behind its distribution resides in the fact that malware was created to infect Programmable Logic Controllers (PLCs) of nuclear stations. It was the first time a critical industry was attacked through malware. Such an event is interpreted as a step forward towards the emergence of a new cyber war, which might have severe negative impacts on the national and the international security. This event constitutes an endangerment of human lives, especially when a critical facility such as a nuclear power-plant can be reached through cyberspace. Furthermore, the existence of widely available encryption and anonymizing techniques makes the surveillance and the investigation of cyber attacks a much harder problem. In this context, the availability of relevant cyber threat intelligence is of paramount importance.

A promising approach to gather cyber threat intelligence is to collect and analyze traffic destined to unused Internet addresses known as darknets. Since these addresses do not correspond to any legitimate host or device, any observed traffic falls into the following four categories: Propagation attempts from software worms or botnets, network scanning or probing, backscatter from spoofed IP addresses in denial-of-service attacks, or mis-configuration of network devices. Understanding such traffic may thus provide valuable relevant information for preventing, detecting, assessing, mitigating and attributing attacks. In addition, it might contribute to the prediction and forecasting of future attacks to facilitate both real-time and proactive defence of cyberspace.

The term *darknet* refers also to concealed communication, or an unallocated pool of IP addresses. It is described as a network telescope used as an Internet monitoring sensor, a black hole or an Internet sink. In accordance with security, darknet is seen as a double-edged sword since it has a negative as well as a positive impact on cyberspace security. The negative side of darknet lies in providing anonymity for information publishers, which makes tracing them difficult. Moreover, it grants terrorists, spies and criminals the ability to post inflammatory propaganda or hatred slogans without being identified or traced. In addition, darknet can afford a safe environment for a virtual black market where stolen information like credit card numbers, stolen intellectual property and email address lists can be traded. Besides, it serves as a suitable platform for hackers to launch Advanced Persistent Threats (APTs) that can target corporate and governmental organizations. APTs can be coordinated silently through darknets to conduct an attack on a specific target within a controlled time frame. Conversely, the positive side of darknets is vested into the deployment of unused public IP addresses for traffic monitoring. Unused IP addresses are routable public addresses, which are not associated to hosts. These addresses are not advertised to the public. As a result, any traffic directed to such IPs can be categorized as unsolicited traffic. The latter can be a good source to find illicit activities and communications. In this project, we consider a darknet as a medium that needs to be investigated thoroughly to find interesting insights related to cyber security. For this purpose, we expect to show how darknet traffic can contribute to security axioms, namely, prevention, detection, assessment, mitigation and attribution.

1.1 Objectives

The primary objectives of this study are to:

- Define darknet techniques, tools and technologies and evaluate their readiness levels
- Identify areas that require research and estimate time horizons for each of them
- Characterize and profile darknets in terms of sources, destinations, ISPs, protocols, ports, vulnerabilities, etc.
- Determine the key components and characteristics of darknet information required to predict attacks on the Government of Canada networks
- Analyze prevalent trends, threats, and risks associated with darknets
- Design and implement a testbed for automatic cyber intelligence generation. Such intelligence is meant to be used in preventing, detecting, mitigating and attributing darknet-based attacks
- Provide recommendations on mitigating techniques together with strategic advice on the capability road-map of activities/initiatives that are meant to address the identified research gaps

1.2 Report Organization

The remainder of this final report is organized as follows: In Section 2, we present relevant background information on darknets including their concepts, characteristics and threats. In Section 3, we present the state-of-the-art in terms of darknets along with the underlying research gaps. In Section 4, we discuss the methodology of the project by presenting the underlying architecture and the components of the experimental testbed. In Section 5, we present the darknet analysis results on profiling and characterizing darknet traffic and threats. Furthermore, in Section 6, we introduce a cyber-intelligence framework that allows the detection of malicious domains and IPs. Additionally, Section 7 presents a system that provides security services in the area of brand protection. Moreover, in Sections 8 and 9, we elaborate on our work in the area of threats prediction, for the purpose of mitigation, by discussing time series analysis and threat correlation using data mining approaches. Finally, Section 10 states the conclusion and pinpoints several recommendations.

2 Background Information

In this section, we aim at deepening the understanding of the fundamental concepts and widening the definitions of various related topics of significance. Moreover, we elaborate on common definitions of the darknet terms and review some of the most prominent tools and technologies utilized to deploy a darknet-like infrastructure.

2.1 Fundamental Concepts

The term darknet started to emerge into the Internet community at the beginning of this millennium. First, it was coined by Microsoft researchers to define "a collection of networks and technologies used to share digital content" [1, 2]. Later, the term was associated with other meanings and definitions. Up to this moment, no single definition has been globally accepted. Thus, *darknet*, may refer to, but not limited to, the following:

Darknet as Private P2P Communication: *Darknet in peer-to-peer (P2P) communication (typically file sharing and message exchange) refers to any type of closed, private, and concealed communication between groups of people.* It represents a mixture of cordoned-off encrypted peer-to-peer networks that overlay the existing Internet design. Such darknets, often consisting of a tight-knit group of people, are conceived based on trust and common interests. Moreover, joining such networks require an invitation from trusted members.

Darknet as Dark Address Space: *Usually refers to routable public IP addresses that are not publicized or advertised to the Internet community.* These IP addresses have neither assigned hosts nor DNS entries or search engines' indexing. Therefore, noticing the online existence of these elements is not simple without prior knowledge. This address space can be used either for malicious activities or for benign traffic monitoring.

Darknet as Dark Web: Also known as *Invisible Web* or *Deep Web*. It refers to digital content, which exists in the public cyberspace. It is known to be untraceable and inaccessible by regular search engines. Such cyberspace content remains concealed because there are neither registration records among domain name servers nor direct link pointing to it.

Darknet as Dark Fiber Network: Most fiber network service providers have "overcapacity" optical fibers that are installed but not utilized. These optical fibers are also known as *dark fibers* due to the fact that they can be purchased or leased by individuals or organizations. Such optical fiber network constitutes a dark network as it can be used as a medium where unknown information flows.

2.2 Darknet Characteristics

In many cases, darknets reveal some similarities, which are mirrored by the following characteristics:

Anonymity: Darknets were initially conceived to provide anonymity for network nodes. It is hard for network nodes to predict who published the data, who read it and where it is stored.

Decentralization: Most darknets implement a decentralized network architecture. The purpose of decentralization in computer networks is to avoid a single point of failure and to provide protection against censorship and denial-of-service attacks. Consequently, it helps in maintaining a high degree of reliability, scalability and sustained availability to the network.

Concealment from the Public: Darknets do not publish their existence or their content to the public. Therefore, joining such networks requires an invitation from existing members or a connection to the exact dark IP addresses.

Distributed Storage: The content of darknets is distributed over local nodes' storage spaces. The distribution of storage provides huge data storage capability for darknets and facilitates easy backup services at each node.

Internet Infrastructure: Internet provides the basic infrastructure for darknets. The ubiquitous availability of Internet access provides an easy accessibility to darknet users, which results in reduced deployment cost for darknets.

Dependant on Existing Protocols: Darknets depend excessively on existing network infrastructure and routing protocols. Nevertheless, they implement additional layers on top of the existing protocols. Moreover, it is common to implement non-standard routing protocols such as key-based ones to provide scalability and anonymity.

2.3 Threats

In this section, we aim at showing how darknet channels can be exploited to launch significant cyber attacks. Consequently, we introduce in the following the various threats that can be conducted through darknets.

Advanced Persistent Threats: These are stealthy cyber attacks, which target corporate and governmental organizations. They require humans' coordination involvement rather than relying on just automated code. A human operator usually has a great knowledge of intelligence-gathering techniques such as telephone calls interception and network communication spoofing. The operators monitor victims and interact with them in order to achieve objectives of their planned attacks. Darknets can provide means to conceal communication between hackers to operate their cyber strikes.

Brand Abuse: Darknets can provide a safe environment for brand abusers to accomplish their goals without being traced. Their abuses include spamming, phishing, brand hijacking, counterfeiting, privacy, and malware distribution. These abuses have a drastic and negative impact on the reputation of commercial organizations.

Zero-day Attacks: Zero-day attacks are those that are found in the wild without yet being analyzed and/or patched. Darknets can be means to launch Zero-day massive infections and vulnerabilities'

exploits.

Distributed Denial of Service (DDoS): It is an attempt to make a computer or network resources unavailable. It consists of attacks that are deployed by one person or a group of people to temporarily or indefinitely shutdown services. The timing of such attacks can be coordinated to exploit the availability of critical organization infrastructure by directing enormous flood of Internet traffic to a small set of targeted IP addresses belonging to a target organization. By flooding the available bandwidth with intensive traffic, DDoS perpetrators can effectively bring down a service with potential loss of financial revenue. In addition, DDoS attacks can be coordinated via botnets, which overlay their traffic via darknets to avoid detection of botmasters.

Selling Stolen Information: Organized criminals specialized in identity theft can use darknets to trade stolen information with potential buyers without being concerned about law enforcement agencies. This can include credit cards information, accounts credentials, personal and/or medical information of individuals.

Child Exploitation: A group of pedophiles can create a small tight-knot community using darknets to share child-pornographic content. By deploying friend-to-friend networks, pedophiles can set up friend invitation policies to join their network. Such networks are hard to be identified and infiltrated by law enforcement. Moreover, these hidden networks can be a reliable medium to perpetuate human traffic.

Cyber Terrorism: Recently, terrorists create and use numerous Internet websites to recruit agents to be trained for bombing attempts and murders. These websites can be easily concealed through darknet cyberspace. By exploiting the distributed nature of darknets, the web servers can avoid a single point of failure and make their tracking difficult.

Throughout the cyberspace, governments and companies are increasingly targeted by these threats, which can be performed by criminals seeking economic or military advantage. We demonstrated that such threats can be perpetuated through darknets. The existence of theses threats is a real challenge for IT security and it stresses the importance to investigate the dark cyberspace in order to elaborate and corroborate prevention, detection, mitigation and investigation techniques. Next section is dedicated to introduce the various artifacts (tools and technologies) that can be used in darknets.

3 Literature Review

In this section, we give an overview of the current state-of-the-art that is tightly connected to the darknet topic. We aim at putting forward a clear discussion about darknet-related research, which backs the different tools and technologies as well as IT research and academia efforts. Thus, we try to make a concise review of the darknets' literature in order to let potential readers grasp thoroughly the various aspects related to darknets.

3.1 State-of-the-Art: Tools & Technology

The main intent of darknet deployment is to provide an online anonymity and privacy for communication. As a result, numerous tools and services have been developed for this purpose. These tools aim at securing Internet communication. A significant number of these tools are publicly available as free and open source software. In the following, we describe the pertinent ones.

Freenet: It is the outcome of an idea that was proposed by Clark et al [7, 8] from the University of Edinburgh. The authors of this project claim that Internet usually has two main points of failure. They consider that Internet represents a non-secure storage repository for files. Moreover, Internet does not provide any protection for users' privacy. Freenet is considered as an independent location, which gives a platform for the distribution of files between many computers. These files can be anonymously stored and requested. The design of Freenet is based on five criteria: anonymity for users, deniability for storage, resistance against the denial of access to information by third parties, dynamic routing and storage efficiency, and decentralization of all network functions. Thus, Freenet users can establish a peer-to-peer network, where each node has its own data store. The latter can be securely accessed by other peers to upload and download files. The routing in Freenet network is based on a dynamic routing table containing two entries: addresses and keys. Freenet can be used as a platform to publish websites and to build databases in an anonymous environment. Freenet constitutes a hidden cyberspace that cannot be accessed by public Internet search engines. In order to protect the anonymity of nodes requesting information, Freenet has the ability to encrypt communication. This is done using four types of keys:

- CHK: Content Hash Keys are used to encrypt static content like MP3 or PDF files. It encloses the hash of a file, the description key, which encrypts the file and the cryptographic settings that are used for the encryption.
- SSK: Signed Subspace Keys are used for dynamic dark sites, which tend to change their content. It encompasses the public key hash, the document decryption key, the cryptographic settings, the username and the current version of the site.
- USK: Updateable Subspace Keys are used to link to the latest version of a dark site. USKs are used as wrappers for SSKs. It allows keeping out-of-sight searching for more recent versions of sites.
- KSK: Keyword Signed Keys are used to encrypt saved named-pages in Freenet.

Freenet has started to come into general use. Its network is constituted of thousands of nodes, where participants can publish data anonymously in their storage space.

WASTE: It is a tool that aims at building an easy friend-to-friend network platform. It was integrated initially by Justin Frankel in 2003. The author claims that it allows users to communicate securely and anonymously. It is a software that enables distributed communication for trusted groups of users. A group of users may encompass from 10 to 50 nodes. It has been introduced to allow small companies or small teams in larger companies to efficiently cooperate independently of the network architecture. WASTE[3] is known to be a partially portable tool that is implemented as client-server application for 32-bit Windows. The client application is limited to Mac OS whereas the server application is limited to Linux, FreeBSD and Mac OS. WASTE project contributors aim at completing the portability of the tool since it is an open source software. WASTE network has a mesh-based architecture. It is decentralized and it allows peers to connect to each others individually. Nodes share RSA public keys for communication sessions. The connections are encrypted using the Blowfish algorithm in PCBC mode. WASTE is composed of four services: instant messaging, group chat, file searching and file transfer (upload and download). In order to prevent data collision between WASTE networks, each network is identified by a unique name known as password or passphrase. The assignment of a passphrase to a WASTE network permits to bridge it with other WASTE networks. However, some WASTE networks do not have an identifier and are known as Nullnets.

Tor: The Onion Routing (Tor)[4] project is originally designed and developed for the U.S. Naval Research Laboratory. It is considered as a circuit-based low-latency anonymous communication service [9]. The intent of developing Tor was to ensure private communication between different governmental organizations. Moreover, it can be employed by individuals like law enforcement officers and journalists to securely exchange digital documents and to communicate secretly over the Internet. In 2004, Dingledin et al. introduced a second generation Tor that aims at enforcing secrecy, congestion control and integrity. Tor is a distributed network designed to anonymize applications based on TCP. It can improve the privacy for web browsing, secure shell and instant messaging. The nodes in Tor networks build circuit, where each node has information only about its predecessor and successor. Tor is based on the following features:

- Perfect Forward Secrecy: Tor prevents hostile nodes to record traffic and ask successor nodes to decrypt messages. This feature is achieved using a telescoping path building-design to create circuits. A node that initiates a traffic, is entitled to negotiate session keys with each successive hop in the circuit. These keys are deleted subsequently. As a result compromised nodes can no longer decrypt old traffic.

- Separation of application protocols: Tor requires a separate proxy for each application protocol. It uses common proxy interface, which supports the majority of TCP-based programs. Tor aims at bringing forward a new layer for anonymity by separating application protocols.

[3] `http://sourceforge.net/projects/waste/`
[4] `http://www.onion-router.net/`

- No mixing, padding, or traffic shaping: Tor does not support padding and traffic shaping. The rationale behind this is that no proven and convenient design for traffic shaping improves the anonymity.

- Circuit sharing by multiple TCP: Tor builds a circuit for each application protocol. In order to improve traffic efficiency, it adapts a multiplexing mechanism for transmitting TCP streams through each circuit.

- Leaky-pipe circuit topology: Tor gives the ability to initiator nodes to redirect traffic to other nodes in order to avoid traffic jam and possible volume attacks.

- Congestion control: Tor has a decentralized congestion control, where end-to-end acknowledgment packets are used. The nodes at the edges of the network can detect congestion or flooding. As a result, they send less data until the congestion sinks to a normal level.

- Directory servers: Tor has trusted nodes, which act as directory servers. These are signed directories that describe their known routers and their current status. The users can download their content via HTTP protocol.

- Variable exit policies: Tor has a mechanism installed in each node, which sets connection policy. According to this mechanism the hosts and ports are advertised through which nodes get connected.

- End-to-end integrity checking: Any node within the circuit has the ability to change the content of data. In order to avoid such an attack, Tor verifies data integrity before it leaves the network.

- Rendez-vous points and hidden services: This feature allows nodes to negotiate rendez-vous points to connect with hidden servers. The authors elaborated a protocol that permits two nodes to select a rendez-vous node, which plays a role of a hidden server for both of them.

Vanish: It is a tool that is developed at the University of Washington. It was initially introduced at USENIX Security 2009 [10] by Geambasu et al and aims at protecting the privacy of previously archived data against malicious attacks. The authors wanted to make sure that data become inaccessible after a specific time, which is set by a user. Vanish project is based on cryptographic techniques and distributed systems. The authors has initially integrated a prototype on the top of BitTorrent Distributed Hash Table (DHT). Vanish allows self-destruction of data via BitTorrent peer-to-peer network. The main motivation behind integrating this tool resides in the fact that sybil attacks (peer-to-peer reputation system is subverted by counterfeiting identities) can target peer-to-peer DHTs. Vanish encrypts data with random keys that are not known by users. It tears down the local copy of the key and scatters the key parts across random entries in DHT. The authors justified the use of Vuze DHT as a platform for Vanish as follows: First, Vuze DHT is a huge decentralized representation of peer-to-peer networks. As a result, nodes are spread through many countries. It is hard for attackers to recover scattered part of keys within such networks. Second, DHTs are considered as resilient and reliable distributed storage. They ensure that data can be available for a certain period of time. Third, DHTs tend to clean periodically their content. This property makes data unavailable over time. It is more likely that the data cannot be recovered retroactively. In order to corroborate privacy of data, the authors introduced

the concept of a vanishing data object (VDO). A VDO wraps user data and protects its content from retroactive information. The wrapped data become unreadable after the expiration of a VDO. The key contributions of Vanish prototype implementation are:

- New method for data privacy, which is based on self-destruction of data.
- The prototype is implemented on the top of existing DHTs, which can corroborate privacy in peer-to-peer networks.
- The users can get greater control over the lifetimes of their sprinkled data across the Internet.

Vanish seems to be a good candidate tool to be deployed in peer-to-peer darknets, since it controls the availability of data. The user data can be kept secret and cannot be attacked retroactively. Moreover, Vanish can be coupled with Tor to achieve both anonymity and privacy.

Veiled: It is a tool proposed by HP researchers Wood and Hoffman [11, 12]. It is a darknet tool that is integrated as a plugin for existing browsers. It aims at securing anonymous online communication. It is implemented using PHP and Javascript, which works in any HTML5 web-browser. Veiled architecture is composed of routers and clients. The routers can be considered as peers; they are single server script files. The clients are pure Javascript and HTML web-browser plugins. The clients can exchange messages through Veiled network via routers. The routers provide COMET connections to clients. COMET is a PHP hack, which aims at making HTTP push possible through Veiled clients. They provide Ajax target controls for clients in order to link to web-content. They also furnishes transitory storage for clients. Each veiled router can hook with another veiled router. The hooking allows the transfer of veiled messages through Veiled network. Veiled clients can receive content from routers. They make callbacks for COMET data pipe for HTTP Get Requests. The clients use an API to use the storage of browsers and Java-script encryption and decryption routines. We notice three kinds of communication in Veiled network:

- Client-Client communication: The communication between clients has two types of messages, namely, multicast and routed. The messages are sent through HTTP network layer.
- Client-Router communication: The clients initiate COMET connection with routers. The connection can be based on HTTP authentication or SSL. The client sends hidden frames, which are streamed via scripting function calls. The clients use AJAX to send local messages or events to routers.
- Router-Router communication: The connection between routers is HTTP or HTTPS. The routers user JSON and PHP socket API. The initial connection between two routers contains connection back-information. The routers are mutually accessible and the routing between peers is AODV (Ad-hoc On-demand Distance Vector) routing protocol [13].

Veiled has the following features, namely, private chat, Redundant Distributed File Storage (RDFS), web in the web, distributed Javascript jobs and server failover. The private chat protocol is based on a set of cryptographic algorithms. Once a client wants to chat with another client, he receives

client aliases and public keys. The chat starts from the initiator who generates an AES (Advanced Encryption System) key. He encrypts this key with remote client's public key and sends it. The clients can start chatting by encrypting messages with the AES key. RDFS is based on browser storage and secure upload and download of files. RDFS protocol is twofold: upload and download. When a client wants to upload a file, he selects a file and submits HTML forms. The router slices the file into 1k chunks and multicasts a storage request on darknet. The client waits for chunks to be registered in order to let other clients receive routed data. When a client wants to download a file, he multicasts the file identifier. The other clients check if their data stores contain the file identifier. Once found, the file is sent through routers to the client. The *web in the web* is one of the interesting features in Veiled darknet. It is built on the top of the file distribution. The web-content is retrieved from Magnet Hashes URI. Veiled clients have a Javascript API that allows supporting embedded images and frames. The distributed Javascript job is supported by a client API, which runs the different jobs and reports the results. In order to avoid security problems such as Javascript XSS attacks and interblocking threads, this API runs in a sandbox environment. In order to avoid routing problems, a server failover feature is defined. It allows to publicize connect-back details to local clients and inform them which peer routers are down. If a given connection goes down, a client can retry it or connect to another router.

Gnutella: It is a large peer-to-peer protocol and the first that aimed at decentralizing peer-to-peer networks. Gnutella[5] was designed to let users swap media files through the Internet. In 2007, it was the most popular file sharing protocol representing 40.5% of peer-to-peer computers [14]. Gnutella clients work as follows: Once a client node is connected, it requests a list of addresses. A client tries to connect to the nodes that are listed. It connects to reach nodes and caches the addresses that are not tried so far. It discards the addresses that are marked as invalid. Newer versions of Gnutella protocol are based on two kind of nodes, namely, the ultra nodes and leaf nodes. An ultra node is considered as a higher outdegree peer, which connects to 32 other ultra nodes. The leaf nodes are usually connected to 3 ultra nodes. The maximum number of hops for a peer-to-peer query is 4. Gnutella protocol uses a Query Routing Table (QRT), which is a table containing hashed keywords. Each leaf node sends its QRT to its ultra nodes. These nodes merge all QRTs of their leaf nodes in addtion to their own QRT. The ultra nodes play the role of a coordinator between query peers and peers that have query results. These results are delivered through UDP protocol to nodes that initiated the queries. Gnutella employs five packet types, namely:

- ping to discover hosts on a network
- pong to reply to a ping
- query to search for a file
- query hit to reply to a query
- push to download request for firewalled nodes

Gnutella has many extensions such as SHA-1 checksums for packets' integrity, query and query hit transmission via UDP, dynamic queries via TCP, slices parallel downloading (swarming download

[5] `http://www.gnu.org/philosophy/gnutella.html`

technique) and content exchange known as download mesh. Gnutella protocol has been considered as a good platform for peer-to-peer networks. In addition, its source code is available and can be used to develop new tools for dark peer-to-peer networks.

BitTorrent: It is a group of torrents confined to a limited number of darknet user communities [15]. These torrents are accessed only by the inner circle of the darknet community that is represented by the registered users. To become part of the community, a user needs an invitation from an existing user and has to register. Once registered, the new user obtains a unique "passkey", which will be used to identify and authenticate the access to the dark torrents. Usually, torrent files retrieved from dark BitTorrent sites contain the tracker IP address and additional meta-data. Whenever a user connects to the private IP, the tracker-IP performs a validation of the "passkey" submitted by the user. Once the validation of the passkey completes successfully, the tracker provides the user with the current online status of the required torrent file. Dark BitTorrents manifest some unique characteristics such as a "ratio incentive" policy whereby the users can be encouraged to increase their upload to download ratios. By keeping track of these ratios, the users with low values are issued a warning. On the other hand, a high-ratio user is given the privilege to invite new users.

OneSwarm: Unlike BitTorrent and Gnutella, which are vulnerable to user behavior monitoring, OneSwarm[6] main focus is to achieve a better privacy in P2P applications. OneSwarm provides explicit privacy control for users to share data to specific people instead of random users. This mechanism is called Friend-to-Friend (F2F) information sharing. The three main features of OneSwarm are:

- Privacy, which is achieved by using the source address of the user. The data is not sent directly between a specific sender and recipient, but is forwarded through several intermediate users.
- A user-friendly web-based interface that provides real-time video and audio services in different formats.
- A free tool that is compatible with Torrent clients.

3.2 State-of-the-Art: Research Projects

This section reviews existing research projects initiated by different commercial, academic, and government-backed organizations to investigate darknet communication.

Bro: Paxon describes Bro [16] as a standalone system for detecting network intruders in real-time by passively monitoring a network-link over which potential intruders' traffic transits. An overview of the system's design is provided with emphases on high-speed (FDDI-rate) monitoring, real-time notification, clear separation of mechanism, policy, and extensibility (expanding Bro with new protocols analyzer). Bro is composed of an event engine and a policy-script-interpreter. The event-engine is a layer that performs integrity check on packet headers. In the case of a corrupted header, Bro generates an event and discards the packet. Otherwise, the event-engine records the tuple of source and destination IP addresses and TCP or UDP port numbers. The event-engine saves the entire packet

[6] `http://www.oneswarm.org/`

or its header to a trace file. Bro records the header if the packet is a SYN/FIN/RST packet. The different records represent high-level events, which are processed by a policy-script-interpreter. The latter is considered as a set of Bro language scripts, which are applied to the different events. This interpreter tries to correlate events and synthesize them to generate real-time detection notifications using syslog. A number of attacks that attempt to subvert passive monitoring systems and related defense mechanisms are discussed. Six case studies were conducted on Bro. The Bro analysis enclosed Finger, FTP, Portmapper, Ident, Telnet and Rlogin. Bro can run on Unix, FreeBSD, IRIX, SunOS, and Solaris platforms. Bro is publicly available as an open source software.

SPADE: SPectrum Analysis for Distinction and Extraction (SPADE) [17] is used to extract malware features. The proposed techniques are based on spectrum analysis, which applies Discrete Fourier Transform (DFT) to darknet data and derives correlation coefficients between different series of malware network data. SPADE algorithm consists of eight steps, which are as follows:

- Hamming Window Function for Series Data: This function aims to characterize oscillations between different data series of destination IPs.
- Discrete Fourier Transform: It allows to derive spectrum from a series of data. This spectrum represents the characteristics of the original scanning behavior.
- Removal of High-Frequency Bands: The data series are affected by some networking events such as packet losses and disorders in packets' sequences. These events are represented by high frequency bands. SPADE discards frequency bands that are higher than a certain predefined vertical threshold.
- Extraction of Maximum Value Indices: SPADE extracts the peaks of frequency components.
- Removal of Fundamental Frequency: The purpose of this step is to adjust the number of oscillation cycles. This is done by normalization of index values.
- Standardization of Harmonic Structure: SPADE standardizes the weight of each normalized index value and derives a series of deviation values.
- Synchronization and Alignment of two data series: SPADE adjusts two data series based on the index values of their base frequencies.
- Derivation of Correlation Coefficient: SPADE computes the correlation coefficient between two series of data.

The authors of SPADE initially applied the algorithm on five data series generated from five bots belonging to the same botnet. They managed to compute correlation coefficients for each couple of bots. The coefficient average was 0.73, which signifies that SPADE can recognize infected hosts. These hosts are infected by bots belonging to the same botnet. The authors conducted another experiment where bots belong to different botnets. They obtained dissimilar scan patterns and the coefficient average was 0.41. This average signifies that SPADE can distinguish between hosts that are infected by bots belonging to different botnets. These experiments demonstrate SPADE's capability to detect

similarities and dissimilarities within the same and different bot classes. Notably, SPADE is a part of the Nicter project, which is described in the following paragraph.

Nicter: Nicter stands for Network Incident analysis Center for Tactical Emergency Response. It is a large-scale network incident analysis system [18]. It represents a system that is capable of capturing and analyzing malware executables. The identification of malware propagation is the primary purpose of this project. Nicter is composed of four components, namely, the Macro analysis System (MacS), the Micro analysis System (MicS), the Network and malware enchaining System (NemeSys), and the Incident Handling System (IHS). The MacS is a set of distributed sensors deployed in universities and corporations to collect darknet data. This data is used to detect malware's scans. The darknet traffic is an input to an analysis-engine (a darknet black hole monitoring), which detects new scan patterns. Meanwhile, the MicS seizes malware by using honeypots, web crawlers and dummy email accounts. Malware are analyzed to extract their behaviors. The analysis results are saved in a database, namely, Malware knowledge Pool (MNOP). The NemeSys provides as output the correlation results obtained from MacS and MicS components. The result tuples contain IPs and the list of malware infecting the different hosts. The IHS helps to compile the different results into incident reports that can be sent to ISPs, governments and users. Nicter has been operating for more than five years for analyzing trends of cyber attacks. They managed to prove the importance of the large-scale network monitoring. Nicter monitoring has shown the ability to apprehend different malware propagation strategies.

Network Telescope: The Network Telescope [19, 20] is a system proposed by researchers at the Cooperate Association for Internet Data Analysis (CAIDA). The intent is to monitor pandemic and epidemic cyber incidents through the unused (dark) address space. Moore et al. proposed Network Telescope as an efficient and effective darknet traffic monitoring system by using sensors and virtual machines. The Network telescope is assimilated to astronomical telescopes. The reason behind this similarity is that large and sensitive telescopes have a higher probability to observe new phenomena. The Network Telescope can monitor large chunks of unused address space. The concept of Network Telescope is based on single host terminology and network terminology. The former represents the networked computing device that sends traffic collected by telescopes. The transmission of data can be uniform or random throughout the address space. Each host can choose its targets and send to it data. Such actions constitute events that may be influenced by external characters. The term event is used to describe single host actions. The network terminology is tightly linked to IP addresses. Each IP address represents a host and each telescope has a set of IPs. University of California deployed network telescopes to monitor a single unused Internet address space of /8 – a range of 224 address space. Since a network telescope uses a passive model, it monitors packets without further interaction with the attacker. This passive nature makes it difficult to capture enough intelligence about TCP based attacks, as the payload is only transmitted after the TCP handshake. Therefore, TCP handshake will be monitored as an attempt to initiate connection. The lack of distributiveness combined with its passiveness, makes it difficult for network telescopes to differentiate malicious traffic from mis-configured traffic. Moreover, it represents an easy target to be fingerprinted by attackers.

The *Leurre.com* Project: This project was launched in 2003 for the purpose of collecting Internet threats using worldwide distributed Honeynet[21]. It is composed of multiple honeypots, which are distributed in 30 different countries. The terms used in the context of this projects including platform architecture, logs collection, data uploading mechanism and data enrichment mechanism, are explained below.

- Platform architecture: The main objective of *Leurre.com* project is to make a concise comparison between unsolicited traffic in distinct locations. In order to achieve this objective, a common platform architecture was installed in each location. The architecture is based on Honeyd, which is configured to run three virtual hosts with different IPs. In each virtual machine, a set of ports are opened (FTP, Telnet, Netbios name service, etc.). TCPdump runs to capture network traces on each virtual machine. A reverse firewall is set up to let the system drop connections that are initiated by the system.

- Data collection mechanism: It is an automated mechanism that allows collecting network traces through an encrypted connection. This mechanism runs through a script that downloads log files, which are stored in a server.

- Data uploading mechanism: It is a set of Perl programs that parses log files and generates different abstract data, which is stored into a database. The abstract data encompasses, for instance, source IPs, port sequences, large and tiny sessions, and clusters of network fingerprints.

- Information enrichment: It allows adding a layer of knowledge about the collected data. It contains geographical information, operating systems fingerprints and domain names. The ScriptGen technology [22, 23] is used to detect malicious activities in network traces.

The *Leurre.com* project is part of WOMBAT project described hereafter.

WOMBAT(Worldwide Observatory on Malicious Behavior and Attack Threats) : It is a research project funded by the European Union under the 7th Framework Program (FP7) for Research [24]. This project aims at providing new artifacts to understand emerging threats. WOMBAT is used to collect raw data and analyze it in order to identify different threat phenomena. The authors claim that WOMBAT can discover trends of attacks by understanding the behavior of threats. For this, the designers of WOMBAT have developed mechanisms for automatically collecting and analyzing malware. WOMBAT has the following features:

- Improving data acquisition technologies: WOMBAT project shares information with its partners, including, SGNET (The *Leurre.com* Project) [21], Argos [25], Nepenthes [26], NoAH project[7], and SANS Internet Storm Center [27]. In addition, it uses honeypots like Shelia [28], Bluebat [29], HoneySpider Network [30], and Paranoid Android [31]. It also uses historical archive of malicious URLs, namely HARMUR [32].

- Addressing data sharing problems: WOMBAT adopts strategies to transform the data provided by different partners into a uniform format. For this purpose, the authors reviewed the different

[7] `http://www.fp6-noah.org`

types of data in order to make relevant correlation. Trust issue is a prerequisite to let the partners share their data. The authors of this project have set up legal policies and barriers to avoid the use of sensitive data such as zero-day malware to perpetuate cyber attacks. In order to improve effectiveness of WOMBAT project, the authors have created an API, called WAPI. The purpose of this API is to ease the usage of data that is shared in WOMBAT project.

- Data enrichment: This feature provides the capability to bring the data to an advanced comprehensive level in order to collect threat intelligence. For this purpose, the authors have used various technologies to create variants of malware (packing tools) and analyze dynamic behavior of malware such as REANIMATOR [33].
- Threat intelligence: The ultimate goal of WOMBAT is to find the root cause of observed attacks and build a knowledge base for predicting future threats.

Internet Storm Center (ISC): The ISC project[8] is started in 2001 in order to monitor ISP customers and trigger alert messages about online attacks. The main purpose of this project lies in providing a mitigation strategy to counter malicious attacks. This project is initiated by incident handling volunteers, who use their expertise to prevent and detect intrusions. ISC uses DShield firewall [34] in order to correlate suspicious events. In addition to DShield firewall, ISC has Webhoneypot, which is a web server incident logs submission system. All submissions are archived and compiled to generate reports. ISC records important and harmful attacks and corresponding unique URLs and headers.

Internet Motion Sensor (IMS): It is a distributed globally scoped Internet threat monitoring system deployed at the University of Michigan [35]. It has the ability to monitor dark IP space. IMS is based on a distributed blackhole network with lightweight responder, a payload signature and caching mechanism. These capabilities are used to generate new insights about worms, DoS, and scan activities. The IMS architecture is composed of a distributed blackhole network. The latter enhances the visibility of distant suspicious events. IMS has a responder, which is designed to split traffic services. The payload signature and caching mechanism decreases the overhead generated from request payload storages. The authors have presented IMS architecture in the context of a 28 block, 18 organization distributed deployment. IMS uses passive sensors to collect UDP and ICMP traffic. It also uses lightweight sensors to gather TCP three-way handshakes. These active sensors reply to the first SYN packets of TCP handshake protocol in order to capture malicious payload traffic. The authors have examined three distinct large-scale events namely, the Blaster worm, the Bagle backdoor scanning and the SCO denial-of-service attacks. One of the drawbacks of this monitoring system is its inability to detect application-level threats such as NetBIOS attacks.

Black Holes: Cooke et al. [36] from the University of Michigan devoted their research on understanding the impact of sensors emplacement and the distribution of unused address space on IMS darknet data. This study aims at deploying ten distributed blackhole sensors in ISPs, enterprises and universities with address blocks that range from /25 to /8 IP address space. In addition, this research attempts to identify the importance of sensor placement as a significant factor in generating measurements from unused space. Moreover, the authors analyzed empirical results, which demonstrate the

[8] `http://isc.sans.org/`

differences between traffic observed on different sensors. The Black Holes are designed to maintain a high-level of interactivity to differentiate traffic, to characterize emerging threats and to provide a visibility into Internet threats beyond geographical and operational boundaries. In this work, Cooke et al. observed some phenomena on IMS deployment. They started by looking at the packet rate with each sensor. They noticed that the amount of traffic can differ between sensors. The authors justified this by the fact that smaller blocks of IPs are closer to live hosts than larger blocks. The main reason of such a claim is as follows: any person or malicious program that tends to use scanning algorithms that most likely have local address preferences. Cooke et al. characterized protocols and services on local /16 and /8 address spaces. They found out that the amount of local /8 traffic varies and some sensors had significant amount of local /8 traffic but it is not the case for other sensors with the greatest load of overall traffic. The authors noticed that 99% of global traffic is based on TCP. This is due to the fact that IMS sensors respond to TCP packets. Moreover, the authors wanted to identify the targeted destination ports. Ports 445 and 135 are consistent in all sensors. The authors focused on port 135 because it is used by Windows DCOM RPC service which has been the target of several vulnerabilities and exploits. In particular, TCP port 135 is the infection vector of Blaster worm. The authors have considered this worm as a case study object. They studied the propagation of Blaster worm. The latter chooses an inital target address in the same local /16 address space. Afterwards, it scans sequentially blocks of 20 addresses starting from that inital address. In order to corroborate their study, the authors explain the differences between observed sensors. They itemized the following properties, which provide valuable insights about differences:

- Filtering policy: It represents different techniques deployed in firewalls and routers, which can affect reachability of traffic. For instance, ISPs tend to install access control lists and content signature filters to block notorious traffic.

- Propagation strategy: Worm scanning algorithms or propagation strategies often have a bias towards local addresses. For this reason, a huge load of traffic can be seen on these addresses.

- Sensor address visibility: The lack of reachability of address space is mainly due to network failure, misconfiguration and improper security policy. Such practices can have a bad effect on Black Hole Sensors visibility.

- Resource constraints: Another important aspect that impacts on reachability is resource constraints. They determine the performance and availability of data within sensors.

- Statistical variations: Another cause of differences between sensor results from the fact that traffic is sampled from a pool of address space by ignoring factors mentioned previously, and expecting a uniformly random scanning algorithm. However, a big variance between sensors can be noticed. This is more likely due to an erroneous sampling process.

Cooke et al. explore issues associated with Black Holes monitoring. They demonstrated the differences that may exist in observations done at sensors deployed in different locations. They showed that differences may occur in aggregated data, protocols and ports.

Internet Sink (iSink): Yegneswaran et al. [37, 38], from University of Wisconsin-Madison, developed Internet Sink to monitor unused IP address spaces. The iSink system was conceived to address the scalability issue that is related to large address spaces. It incorporates passive detection and monitoring sensors as well as honeynet components. For the passive detection, it uses Argus [39] (IP flow measurement tools) and employs a stateless responder based on the Click modular router platform [40] (a toolkit for building a high-performance network system) for the purpose of active monitoring. The active responder is able to reply to packet requests from the application layer such as HTTP, NetBIOS, and DCERPC. It also has a NAT filter, which performs address translation between the active monitor and the honeynet. An off-line Network Intrusion Detection System (NIDS) is used to log traffic filtering information. The iSink system comes into two distributions, the "campus-enterprise" iSink, which monitors four large unused IP ranges of class B/16 and the "service-provider" iSink, which monitors the class B/8. iSink authors elaborated a darknet case study to analyze the collected traffic. The analysis is composed of four components:

- Analysis of backscatter packets: This analysis aims at characterizing responses to spoofed DoS attacks. The authors established a time series graph of the backscatter packet volume. They noticed that the most common responses to a SYN flood packets are TCP packets with ACK/RST. They observed a less common short duration spikes for SYN/ACK and SYN/ACK/RST responses. They attributed the exceeded ICMP packets to routing loops or DoS flooding with low TTLs.

- Investigating unique periodic probes: The authors managed to observe the time periodicity on collected data for the purpose of intrusion detection. They isolated TCP flows for two scanning services running through ports 139 (Server Message Block protocol over NetBIOS) and 445 (Direct Server Message Block protocol).

- SMTP hot-spot analysis: The authors managed to identify SMTP hot-spot. They discovered the existence of an IP address, which attracted a large number of SMTP (Simple Mail Transfer Protocol) scans. This IP is bound to 14.000 IPs, which results in 4.5 million scans.

- Experiences with recent worms: iSink deployment showed its relevancy in detection of worms such as Sasser, which uses *lsarpc* exploit. Moreover, the authors managed through iSink to observe different Sasser variants and other malware, namely, Agobot and RRBOT.CC.

In order to improve the effectiveness of their tool, iSink's authors integrated connection sampling in their architecture. The main reasons behind using this approach are: reduction of bandwidth requirements, improvement of scalability and simplification of data analysis.

Cymru's Darknet: The Darknet project deployed in 2004 by Team Cymru Community [41] is a passive Internet threat monitoring system. Its main purpose is to set a platform to collect packets susceptible to be sent by malware. This darknet is deployed to host flow collectors, backscatter detectors, packet sniffers and intrusion detection systems. Team Cymru aimed to increase awareness about threats and enhance mitigation against malware. This is achieved by detecting dangerous traffic and identifying the sources of such traffic. Team Cymru darknet uses Simple Network Management Protocol (SNMP). It calculates statistics about inbound, outbound and dropped packets. This team used an upstream

router, where secure JunOS template is installed. This router supports SNMP to collect traffic. There are two reasons behind using SNMP:

- Detection of uncommon malicious network activities, which gives hints about the existence of malware.
- Detection of any outbound traffic, which is considered as a bad sign and an alert needs to be generated.

Team Cymru router is configured to export traffic to external tools, namely, Silk[9] and Flow-tools[10]. These tools are used for integrating network monitoring infrastructure. Team cymru integrated a darknet server, where TCPdump[11], Argus and IP Filter[12] tools are installed. Argus and tcpdump are configured to listen on the NIC interface whereas IP Filter is configured to block everything crossing the NIC interface. Team Cymru provides a nice guideline to set up a darknet. It can be seen as a marked trail to build darknets within companies or institutions.

Billy Goat: This project [42] is a specialized darknet traffic monitoring system deployed by IBM and its customer networks. It is used for worm detection. Billy Goat differs from other monitoring systems since it focuses on specific attacks and dynamic characteristics of worms. By taking advantage of worm propagation strategies, Billy Goat monitors unused IP address spaces that are randomly scanned by worms. It consists of an active feigning server responder that reacts to application-level worm service requests. It possesses a detection mechanism for network-service worms, which exploit network services such as HTTP, MS/RPC, MS/SQL, etc. The architecture of the specialized sensors provides Billy Goat with a "plug and play" implementation. Billy Goat is based on a virtualization mechanism and a data repository. The virtualization mechanism uses standard programming models, and responds to multiple IP addresses. This feature eases the creation of new feigning services and integrating them. The data repository is a storage for all IP and application-level information. All the feigning servers are integrated by using a specialized Java framework. In order to improve effectiveness of Billy Goat at the time when the network performance diminishes, Worm Detection Systems (WDSs) require distributed architectures. For this purpose, each Billy Goat has the ability to analyze and report events detected locally. All Billy Goat sensors monitor data on a centralized repository. This feature allows the detection of infected machines that do not have Billy Goat sensors installed. Billy Goat data analysis is an iterative process, which attempts to identify worm activities. This identification is based on the aggregation of the following methods:

- The use of worm detection analysis described in [43].
- Collecting the set of exploits and matching them with known worms.
- Observation of indicators such as scanning in order to identify worms.

[9] `http://silktools.sourceforge.net/`
[10] `http://www.splintered.net/sw/ow-tools/`
[11] `http://www.tcpdump.org/`
[12] `http://coombs.anu.edu.au/~avalon/`

When a worm is identified, the finding is labeled as an alarm. Suspicious but unidentifiable findings (large scanning activities) are labeled as warnings. All other data are identified with unknown. Billy Goat deployment has four modes, namely, static routes, ARP spoofing, default LAN route and ICMP-based. Since Billy Goat is designed to detect automated attacks by worm, it lacks the ability to remain concealed, which makes it easy to be fingerprinted and detected by attackers.

3.3 New Darknet Aspects

A thorough review of Darknet literature uncovers the existence of new aspects (concepts) that are associated with darknet topic. These aspects include dark clouds and dark social networks, which are discussed below.

Dark Clouds: Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software and information are provided to computers and other devices as a utility (like the electricity grid) over a network (typically the Internet). The term "Dark Cloud" is characterized by the use of dark private and uncensored communication channels to retrieve, store and share information in a concealed manner on the cloud. The most significant challenges for dark clouds are twofold: The first one is represented by the fact that the cloud aims by default to have a privacy-preserving approach and hence all information is (to a certain extent) private or secure. The second challenge is illustrated by the fact that the cloud is inter-networked with the whole Internet (and other services). Hence, the darknet channels extend to the Internet and can play the role of a recipient of concealed information on the cloud.

Dark Social Networks: A social network is a social structure made up of individuals (or organizations) called "nodes", which are connected by one or more specific types of interdependency, such as friendship, kinship, common interest, financial exchanges, dislike, sexual relationships, or relationships of beliefs, knowledge or prestige. Over the past ten years, social networks have extensively flourished and their use has become ubiquitous. The term "Dark Social Networks" refers to concealed (dark) communications that may take place on the social network platform. Although such communications may be harmless, they may be used for illicit purposes such as to maliciously collaborate in criminal and cyber incidents and threats. Thus, there is a need to investigate them and uncover their details.

3.4 Comparative Study

In this part, we aim at providing a comparative study for the different tools discussed in Section 3.1 as well as darknet projects, discussed in Section 3.2. The comparative study of different tools are based on the following features:

- Design: This feature mirrors key requirements in the design of a given tool.
- Protocols: The list of protocols used by a darknet tool.
- Platforms: The list of operating systems where darknet tools can be installed.

- Deployability: Observations that can be noticed on the deployment of darknet tools.
- Security: This feature describes security schema (e.g. encryption/decryption) that are adapted by darknet tools as well as security concerns and flaws.

In another strand, we make a comparative study of the different darknet projects. This comparison is based on the following features:

- Intent: The main objective of a darknet deployment.
- Deployability: It is the description of the address spaces that are used as well as the number of sensors that are deployed in different locations.
- Manageability: It describes how a given darknet analysis is managed and conducted.

The comparative study of darknet tools is depicted in Figure 1, while the one for darknet academic projects is shown in Figure 2 and Figure 3.

Pages 32 to / à 34

are withheld pursuant to section

sont retenues en vertu de l'article

16(2)(c)

of the Access to Information Act

de la Loi sur l'accès à l'information

3.5 Research Gaps

In this report, we have reviewed various state-of-the-art darknet tools and techniques. We have discovered that the research community has shown a notable yet insufficient interest in this topic. While these proposals are now firmly established and constitute important contributions in dealing with darknets, the fact remains that:

- Most of the proposed techniques and technologies provide very limited support for cyber attacks prevention.
- Limited advances have been done on assessing the severity and scope of cyber attacks.
- The proposed mechanisms are very limited in terms of attribution of cyber incidents.
- Very little has been achieved on the correlation of different network/malware information sources.
- The issue of brand abuse including brandjacking, counterfeiting, false association, and piracy is rarely studied.
- Lack of automation when it comes to the identification and geo-localization of command-and-control botnet servers and their bot-herders.
- Insufficient effort to analyze the emergent cyber threats in the setting of large scale social networks.
- Lack of a consolidated and comprehensive framework for automatic generation of cyber intelligence.

4 Proposed Approach

This section presents the methodology of the proposed approach and its internal processes. Moreover, it highlights and describes the architecture and its modules.

4.1 Methodology

This section provides an overview of the proposed methodology by presenting the order in which different components and processes are executed to achieve the desired goals. The presented approach, as illustrated in Figure 4, is composed of the following:

1. Macro and micro analysis, which investigates darknet traffic from two perspectives, namely, feature-based and anomaly-based.

2. Correlation techniques between darknet streams and various sources including spam traps, passive DNS and malware feeds. Such correlation should lead to decisive and significant results.

3. Cyber intelligence generation based on the previous analysis and correlation methodologies. Such serviceable results are characterized as criminal and commercial.

4. Risk assessment analysis to formulate recommendations and enhance best practices for the purpose preventing, detecting, assessing, mitigating and attributing cyber attacks.

5. Prediction and forecasting analysis performed on the relevant generated intelligence for predicting network anomalies and future attacks.

Note that, the above components are elaborated as follows:

4.1.1 Preliminary Analysis

The procedure is initiated by retrieving raw darknet feeds followed by a preprocessing step to extract selected content. The extracted information that is relevant to our analysis is accumulated in data warehouses and repositories for further processing. Amalgamating the extracted darknet data in a single place facilitates the application of diversified analysis mechanisms for achieving various goals. In addition, the repositories serve as result containers, holding intermediate outcomes that are piped for further processing at a following level.

In general, we analyze the darknet data from two main perspectives: Micro-level and Macro-level. Micro-Level analysis is further divided into feature-based and threat-based analysis where the former deals with retrieving general darknet profiling statistics (e.g., traffic distribution, application ports and protocols, IP classes, detected operating system and source countries and service providers) and the latter investigates specific threats triggered from darknets. At the macro-level, we envision the application of both statistical and data mining methods for the purpose of:

Page 37

is withheld pursuant to section

est retenue en vertu de l'article

16(2)(c)

of the Access to Information Act

de la Loi sur l'accès à l'information

- Identifying patterns (e.g., bot connections to command-and-control servers).
- Isolating pieces of information that do not follow these patterns (e.g., network anomalies, bot-herder connections).
- Predicting and forecasting attacks based on change detection in the darknet traffic.

Candidate techniques that will be used as macro-level analysis on the darknet traffic are Principal Component Analysis (PCA) [44] and Discrete Wavelet Transform (DWT) [45]. Such mechanisms could be used to identify anomalous behavior of the darknet traffic. In recent years, PCA and DWT have emerged as powerful techniques for detecting numerous network-wide anomalies. These methods are applied in conjunction with packet time series analysis.

4.1.2 Darknet Correlation

The analysis of the correlated darknet results coupled with other sources would provide a deep insights into the inner workings of malicious darknet traffic and thus the corresponding generated cyber intelligence would aid in detecting *criminal attacks* including DDoS, botnet command-and-control servers, APTs, and Zero-day attacks and *financial attacks* such as brand abuse (e.g., Grey market selling, piracy, false association). A third attack type can be characterized as both, a criminal and a commercial threat, such as malware and offensive content. A simple example of correlation would be to learn the behavior of known malware by developing a model using malware samples and innovative classifiers. The developed model is then applied to identify malicious activities on the darknet. Common classifiers include: (1) probabilistic classifiers (e.g., Bayesian classifiers and their variants [46]); (2) decision trees [47]; and (3) Support Vector Machine (SVM) [48] and its variants. Each classifier category has its own limitations in terms of classification accuracy, scalability, and interoperability. An extensive survey on text categorization suggests that probabilistic classifiers have poor accuracy while decision tree is not scalable to support high dimensional data. SVM outperforms most classifiers in terms of accuracy as well as scalability but it is a black box method as it fails to interpret its findings; therefore, SVM is not suitable for evidence collection and presentation, which are important steps in cyber forensics. In [49], J48 (a decision tree variant) is built using network features extracted from DNS traffic, which is then employed to identify malicious domains from darknet. For instance, to identify whether a given domain extracted from passive DNS queries directed to unused address space is malicious or benign, we can employ a classification built using malicious domains stored in the malware database.

4.1.3 Prediction & Forecasting

By monitoring the macroscopic behavior of darknet traffic, we can reveal critical and significant amount of information about the future behavior of such networks. Recent studies, e.g. [50], show that analyzing the spatial and temporal correlation of unwanted traffic time series can help in predicting the future behavior of a network address space. We intend to investigate different self-similarity techniques for analyzing the dependence of darknet traffic time series for prediction and forecasting. Our presented work in this area is showing in Section 8 and 9.

4.1.4 Risk Assessment and Recommendations

This final step consists in undertaking a risk assessment analysis whereby we will be able (using the generated cyber intelligence) to suggest recommendations and enhance best practices for the sole purpose of alerting and preventing unexpected future incidents and threats.

4.2 Architecture

In this section, we elaborate further on the proposed methodology by discussing the inner mechanisms of our work and revealing the architecture as depicted in Figure 5.

The infrastructure is based on the utilization of various data sources. The primary darknet feeds coupled with other resources such as spam, passive DNS, malware and threat track data feeds constitute the raw information that will be employed, correlated and analyzed throughout this architecture. The ideology is to transform such raw data to information and then renovate such information to relevant data to generate serviceable cyber intelligence that could be of interest and benefit to various agencies and corporations.

The architecture aims at developing an efficient, an agile, a complex and a sophisticated back end environment that is capable to process in near real-time the raw input data sources. To accomplish that, numerous modules are realized or are currently being implemented. Note that, the modules can interrelate to accomplish a certain task. For instance, the basic yet highly required parsing module is used to crawl raw data, extract relevant features (using the specialized module) and save them in data repositories. Such repositories may exist as databases or indexing engines. The latter may be used for faster processing times and efficient storage capabilities. Another significant module is the correlation engine. Since there is a need to generate complex and applicable cyber intelligence, the correlation engine is vital and very desirable. For example, using the correlation engine, we will be able to analyze and correlate spamming botnets and malware-infected hosts triggered from dark networks. Another example would be the capability of matching passive DNS data with the darknet data to pinpoint botnet command-and-control centers. Moreover, other modules such as abuse characterization, malicious domain identification and content analyzer could be employed to detect attacks against financial institutions such as phishing, false associations and typo-squatting.

Our infrastructure, as discussed in the methodology (Section 4.1), intends to target two types of cyber threats, namely, commercial threats and criminal threats. The former deals with threats or abuses against brands (hence achieving brand protection) including those targeting financial institutions. The latter deals with botnets, stolen information (i.e., Credit cards) drop locations, APTs and fast flux. Such information would be beneficial to law enforcement and governmental agencies.

The goal of this architecture is not only to generate cyber intelligence but also to provide as well a serviceable cyber intelligence infrastructure. The aim is to develop a state-of-art front-end system that is capable of: 1) delivering timely alerts about current abuses towards specific brands and institutions for the purpose of mitigation and 2) enabling access to personalized, web-based real-time information monitoring and sharing about criminal as well as commercial threats.

Page 40

is withheld pursuant to section

est retenue en vertu de l'article

16(2)(c)

of the Access to Information Act

de la Loi sur l'accès à l'information

4.3 Testbed and Experiments

In this section, we present the elements of our testbed. The latter is used to conduct experiments on darknet data along with other sources of security information. The elements of this testbed are:

Dataset: We are receiving important security near real-time network data and daily malware feeds from our NCFTA partners. These data sources include:

1. Darknet PCAP traffic files
2. Passive DNS logs
3. Spam trap logs
4. HTTP/HTTPS traffic files
5. Several GB of daily malware feeds

Hardware: The proposed network consists of the following:

1. 2 Servers
2. 10 Desktop computers
3. 2 Network-Attached Storage (NAS) units with several TB of disk space
4. Cisco switches, routers, IDS and IPS units, firewall units, MARS (for network security monitoring)

In order to achieve preliminary analysis, we use a server with the following descriptions:

- Servers: Two Dell PowerEdge T710 and T410
- Operating Systems: Ubuntu 11.04 and Windows XP/7
- Memory: 72 GB of RAM on T710, 32GB on T410, and 12/8 GB on the desktop computers
- Processors: Intel Xeon X5660 at 2.8Ghz on Servers and core i7 on desktop computers
- Hard Drive: Several TB of disk space

Software: Diverse software packages are used for management, monitoring and data analysis. The software list is as follows:

1. In-house C and Java code
2. In-house Python, shell, Perl, PHP and Javascript script code
3. GFI Sandbox
4. Snort [51]

5. Wireshark[13]
6. IPtables [52, 53]
7. Passive OS Fingerprinting (p0f)[14]
8. AppID, opendpi
9. Data mining APIs such as Weka and R
10. Statistical toolboxes such as Excel, Matlab

Database: In order to save the different chunks of data, we opted for the following databases:

- PostgreSQL 8.4.8, installation guide found at `http://www.postgresql.org/download/`, this database is used to store darknet data;
- MySQL 5.5.8, installation guide found at `http://www.mysql.com/`, this database is used to store intelligence collected from malware analysis.

[13] `http://www.wireshark.org/`
[14] `http://www.sans.org/security-resources/idfaq/p0f.php`

5 Darknet Analysis

This work provides the darknet analysis of more than 450 million packets of darknet data collected since September 16^{th} 2011. It includes the analysis of relevant network-based traffic required to compute statistics and intelligence that can serve to detect attacks as well as to geolocate and identify malicious domains, ISPs, victims, protocols and services used in dark communication. Thus, this section gives a sneak peak about darknet data intelligence. This step renders an initial effort of an ongoing work, which aims at helping National Defence agencies, public safety and corporations to gather knowledge about current trends in the dark side of cyberspace.

Our work is based on data collected from the Internet Systems Consortium (ISC) Security Information Exchange (SIE)[15], which is a trusted, private framework for information sharing in the Internet Security field. It is composed of participants that handle real-time sensors and send the acquired data to SIE. Therefore, a significant number of packets are received from network operators (including ISPs, enterprises, academic institutions, and research organizations), law enforcement (internationally), security companies (including anti-virus developers, intrusion detection service providers) and research (including academia, government, and commercial). Using graphs, block diagrams and visual aids, this report represents the data in a scientific, clear and concise manner. These findings can be very informative for IT professionals in relation to potential threats and vulnerabilities and can provide insights about weaknesses in their own security architecture or/and policies.

The profiling part of the report is divided into three main parts: The first is a general summary of the analyzed data. The second represents the darknet profiling based on statistics and features corresponding to protocols, geolocation (e.g., country level or city level), and ports. The third part depicts darknet profiling, which is based on threats and vulnerabilities. In other words, the second and the third subparts are respectively the feature-based and threat-based profiling of the darknet data.

5.1 Summary

The analyzed darknet data constitute a total of 25 GBytes of data with a packet size limit of 65535 bytes (header). Note that, the final version of this project, will include results pertaining to a larger sample of analyzed darknet traffic. Below is some interesting summarized set of findings from the analyzed packets:

- is still the primary target of attacks, whereas is ranked third.
- lead the countries that represent sources of suspicious activities.
- is among the top most source of suspicious activities.
- ISPs in are among the top most source of generating suspicious activities.

[15] `https://sie.isc.org/`

- Code execution attempts in MSN chat messages is a noticeable threat found in dark communication.
- Windows operating systems are the most used on the dark cyberspace.
- Scanning events represent significant activities in darknets.

The packets distribution reveals an instability flow. Figure 6 illustrates this distribution for one day data sample. It is noticeable that the amount of information collected is acceptable to further analyze the data and represent various cyber results.

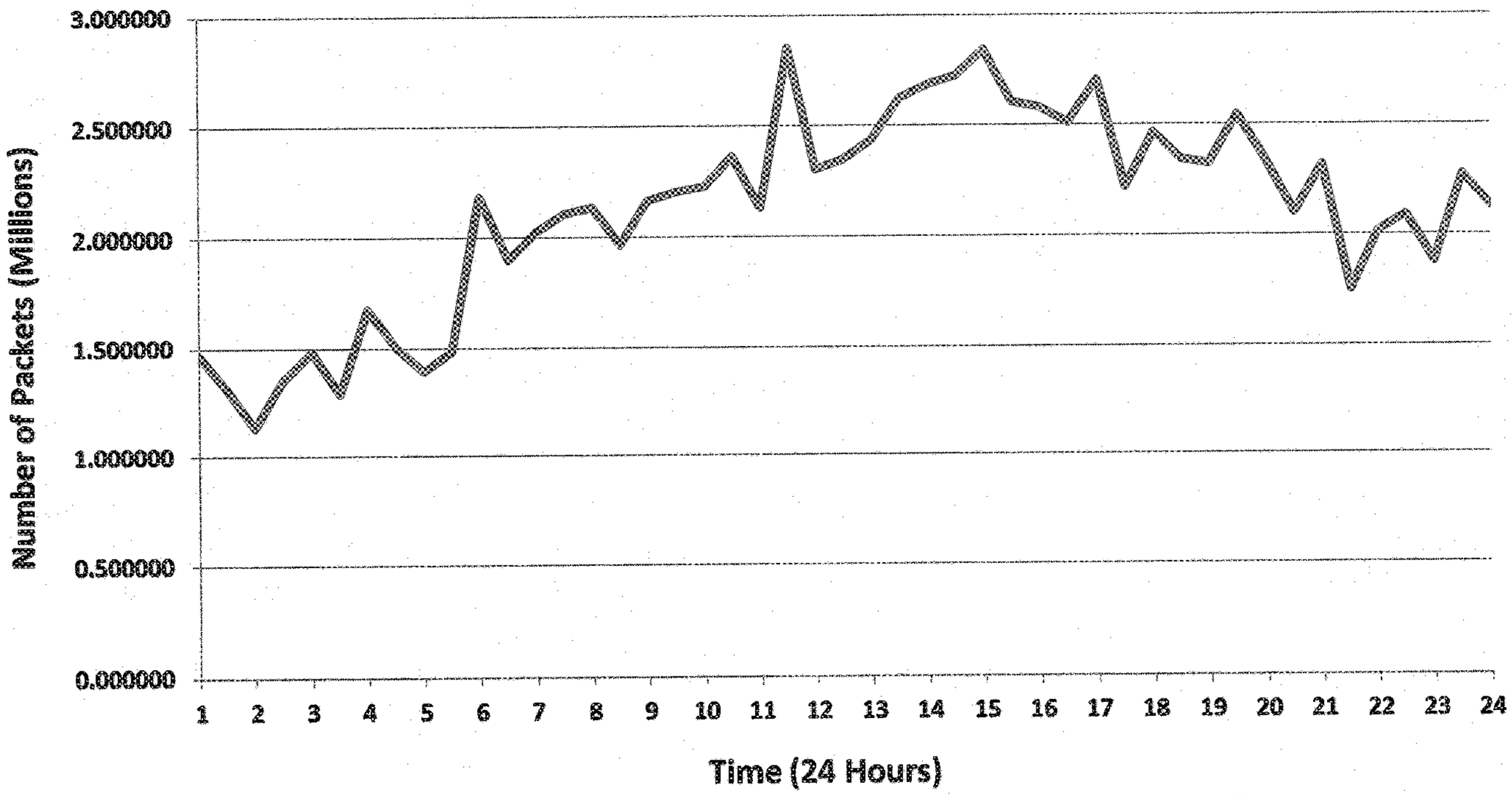


Figure 6: Darknet Packets Distribution

5.2 Feature-Based Analysis

This feature-based section represents the analysis by revealing the list of application protocols, the domain names, the distribution of top most protocols, the type of transport protocols and the geolocalization of the darknet communication in addition to various useful information such as ports, operating systems and connection types.

5.2.1 Protocol Profiling

The purpose of this section is to identify different protocols used on the darknet traffic. We focused on high-level protocols, mainly those existing on the transport and application layers.

Table 1 provides the distribution percentage of darknet protocols.

TCP	UDP	ICMP	Others
90.20%	5.94%	3.51%	0.35%

Table 1: Protocols Distribution

We further measured TCP, UDP and ICMP packets distributions.

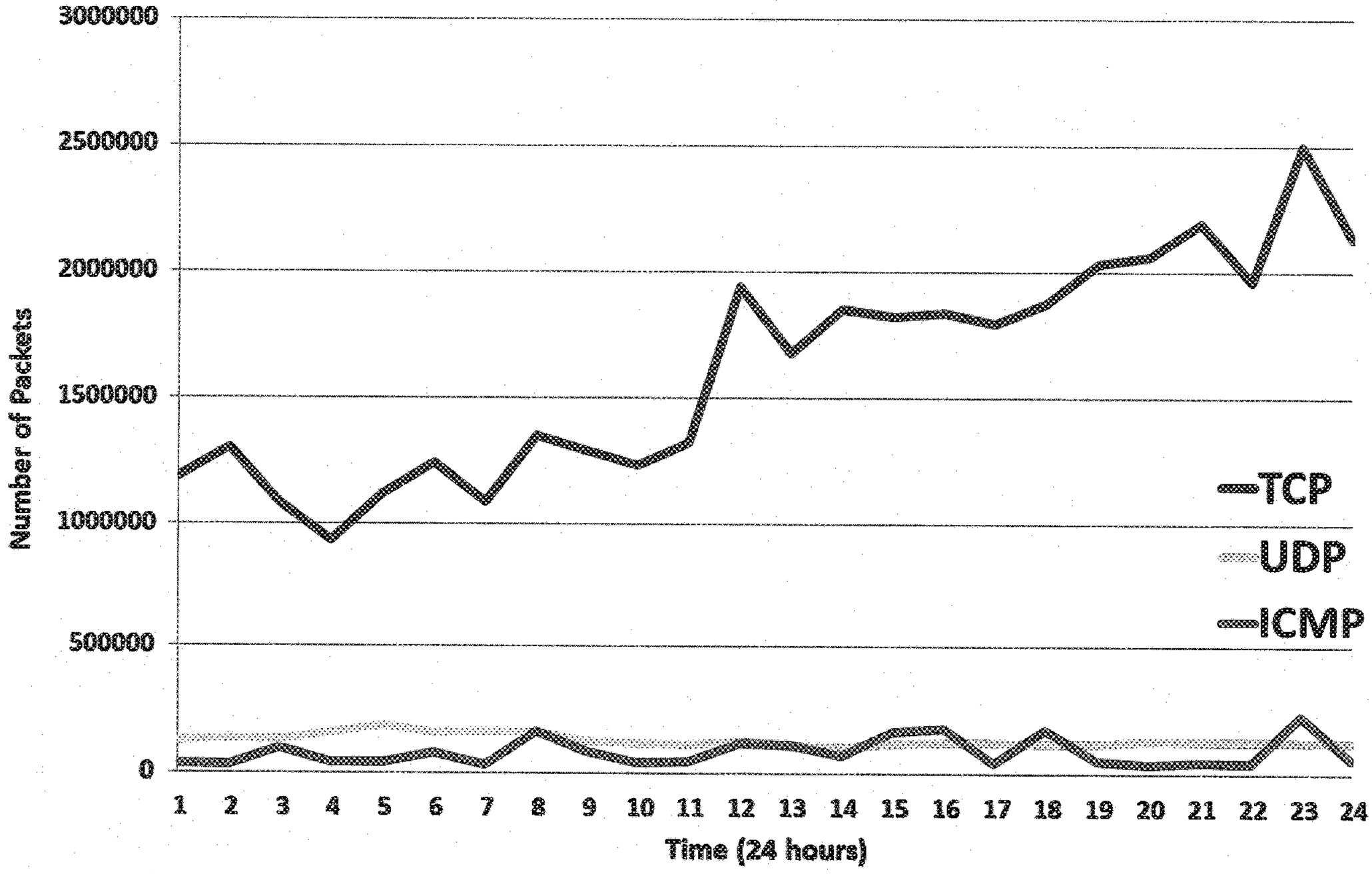


Figure 7: TCP, UDP, and ICMP Packets Distributions

It is observed that TCP plays the major role. Figure 7 fortifies this fact by plotting the packets distribution per protocol in a one day sample. TCP dominance can be explained by two facts; first, the majority of scanning attacks use TCP, and second, there exist known attacks that specifically target TCP ports as noted in [54].

The TCP distribution follows the same pattern of packets distribution illustrated in Figure 6 since this protocol represents the big proportion of packets.

Moreover, we differentiated darknet packets according to their nature and type following the subsequent method:

- TCP SYN packets are classified as scanning traffic.
- TCP SYN+ACK, RST, RST+ACK, and ACK packets are classified as backscattering as these packets are likely to be created by hosts trying to respond to connection from a suspicious source in the darknet.
- The remaining traffic packets are classified as misconfiguration.

Hence, Table 2 depicts the outcome distribution.

Scanning Traffic	Backscattering	Misconfiguration
66.58%	1.50%	31.92%

Table 2: Packets Distribution-Nature of Traffic

The results reveal that scanning or network probing constitute the majority of darknet traffic. Note that, such traffic could be interpreted as an indication of port scanning and/or vulnerability probing. Such attacks, in general, are preliminary triggered before launching a targeted attack towards a specific system.

Application Protocols: The next phase relies in profiling application protocols. For this purpose, we used different tools namely, App-ID[16] and Wireshark to fingerprint application protocols. Figure 8 shows the top 16 application protocols that have been found. The results demonstrate that the is leading while the is ranked second and is ranked third. It is worthy to note that the protocol is and thus its appearance as a top darknet application protocol is significant and maybe alarming. Moreover, the is ranked fourth. This protocol is used for remote procedure calls, which allow programmers to integrate distributed software in a common environment.

[16] http://www.paloaltonetworks.com/products/features/app-id.html

s.16(2)(c)

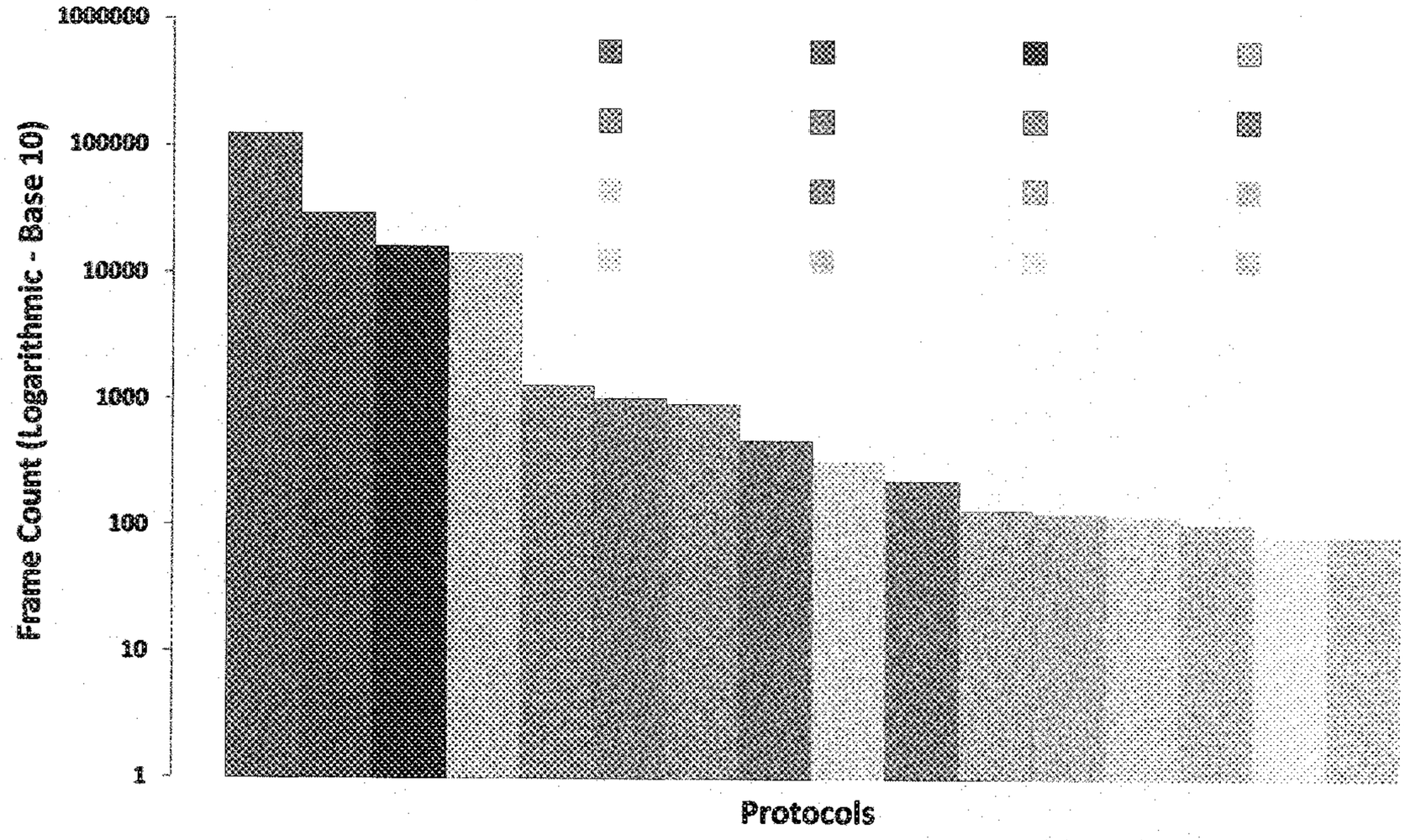


Figure 8: Top 16 Application Protocols

Nevertheless, during the analysis process, we have discovered approximately 160 protocols on different network layers. On the other hand, Figure 9 shows the distribution of the top 3 application protocols based on a logarithmic base 10 scale.

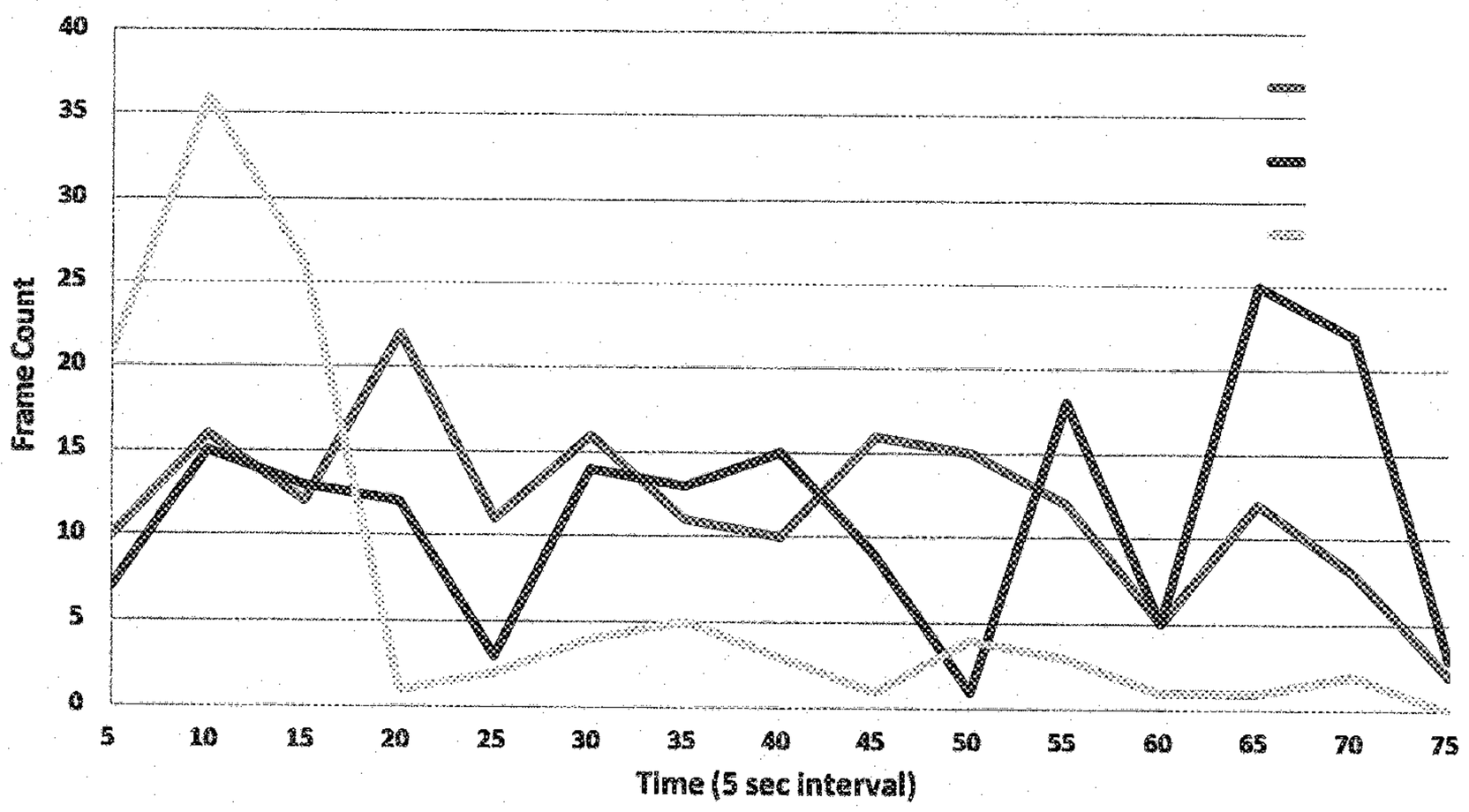


Figure 9: Top 3 Application Protocol Distributions

5.2.2 Domain Names

Figure 10 shows the top 5 resolved domain names on the darknet channel.

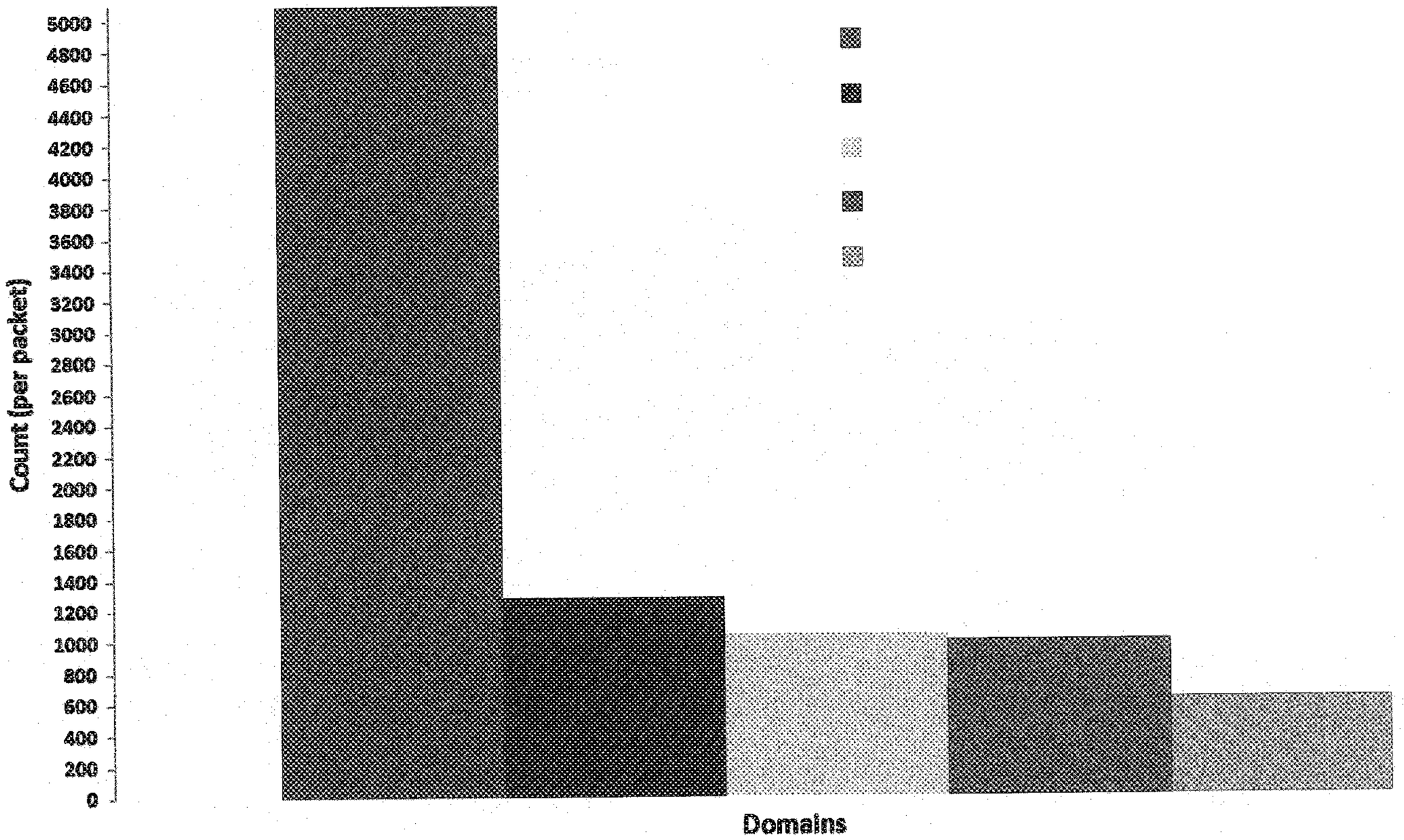


Figure 10: Top 5 Resolved Domain Names

Although the mentioned domains are not malicious, however such results could feed us, in general, with relevant information about unsolicited/malicious domains that could be used by attackers. Note that, for instance, the leading domain which is in fact a common name for the [55] [56], making it vulnerable to malicious use.

5.2.3 Countries

According to our analysis, the source countries reached 196 countries where the majority of source IPs are located in It is as well noticeable that represent the major portion of source IPs compared to other countries. Note that, when we reveal our threat-based severity analysis and geo-localize the sources behind those threats in the next section of this report, the three latter mentioned countries as well appear amongst the top contributed threat countries.

Figure 11 illustrates the distribution of destination countries on the world map. Figure 12 depicts the darknet frequency distribution of source countries on the world map. Moreover, Table 3 elaborates

s.16(2)(c)

Figure 11: Destination Countries - Severity GeoLocalization

on the top 5 source countries.

Figure 12: Source Countries - Severity GeoLocalization

30.50%	30.33%	7.97%	7.75%	7.20%

Table 3: Top 5 Source Countries

Class	Usage (in %)	
	Source	Destination
A	62.529	0.017
B	18.529	7.138
C	18.942	92.845

Table 4: IP Class Distribution

5.2.4 IP Classes

We have studied the distribution of source and destination IP classes in the collected darknet testbed. Table 4, provides the distribution percentage of sources and destination IPs among IP address classes. It reveals that the majority of source IPs belong to class 'A', whereas in the case of destination IPs, class 'C' is in abundance. Furthermore, Class 'A' proportion in the destination IPs is almost negligible, i.e., 0.017% whereas class 'B' contributes relatively more. It is substanial to mention that class 'C', being the most destined and smallest class, could be an indication that it is as well the most targeted class by cyber attacks and hence further investigation could yield very productive cyber intelligence results.

5.2.5 Ports

Another analysis has been performed on ports that are used in the collected darknet traffic. We noticed that a huge proportion of source and destination ports are unconventional ports. We compare collected ports number to the ones listed in IANA (Internet Assigned Numbers Authority)[17]. Table 5 illustrates the distribution of source and destination ports in terms of number of frames.

Port type	Ports Count (in frames)	
	Source	Destination
Unconventional	52758	23274
Conventional	253	131

Table 5: Port Distribution

Next, we analyzed the distribution of packets in terms of the source and destination ports of TCP and UDP protocols. For the sake of simplicity and readability, we plot the distribution of packets for the top 15 ports from each category using the Base-10 logarithmic function. The packet distribution of TCP source and destination ports are depicted in Figure 13 and Figure 14, respectively, while the ones of UDP source and destination ports are shown in Figure 15 and Figure 16, respectively.

[17] `http://www.iana.org/`

Figure 13 shows that more than 6 million TCP packets are originated from port 80 (http), which is used for the World Wide Web service.

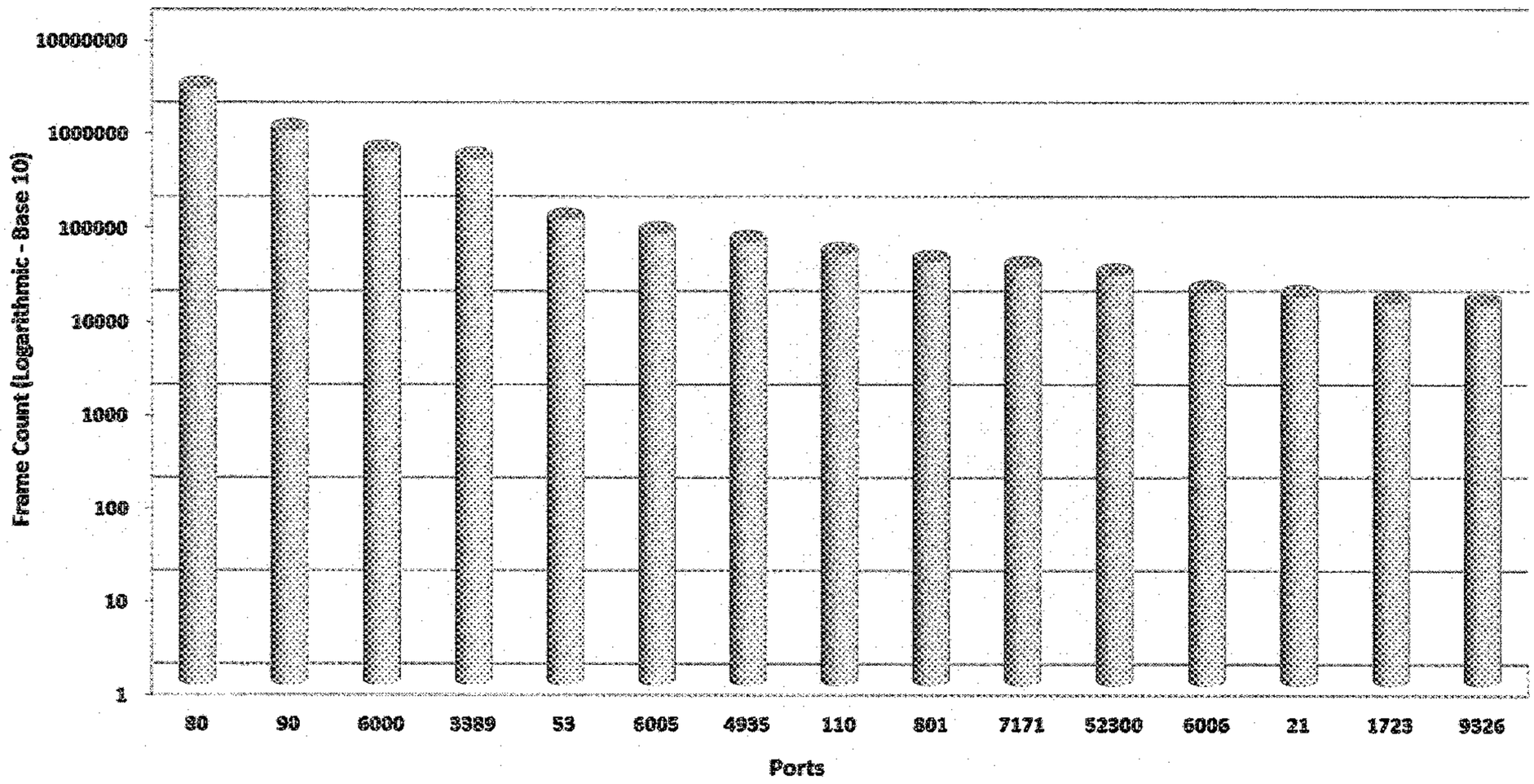


Figure 13: Top 15 TCP Source Ports Distribution

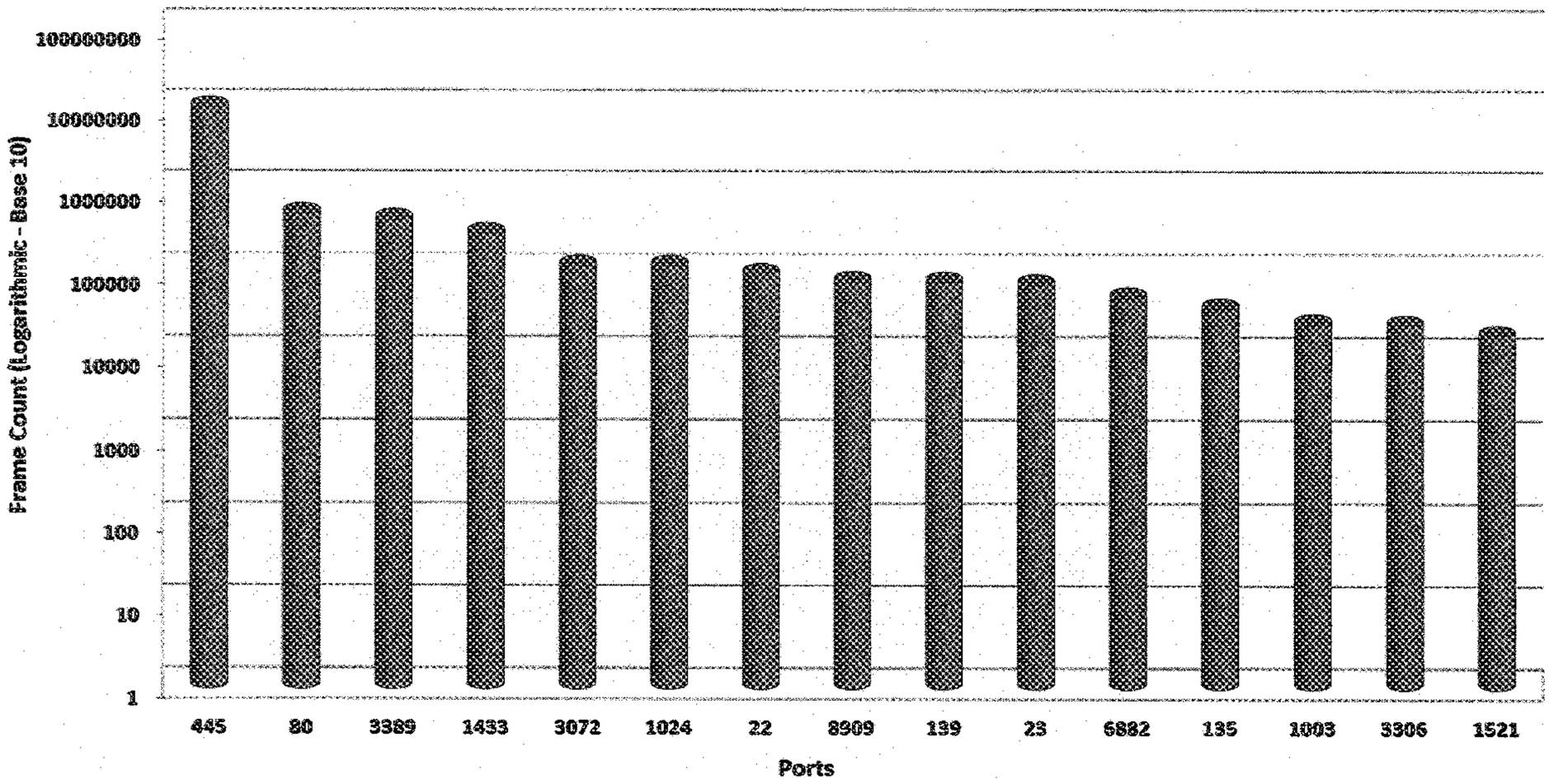


Figure 14: Top 15 TCP Destination Ports Distribution

On the other hand, Figure 15 demonstrates that the leading UDP source port is port 397, which is

the multi-protocol transport networking (MPTN) port.

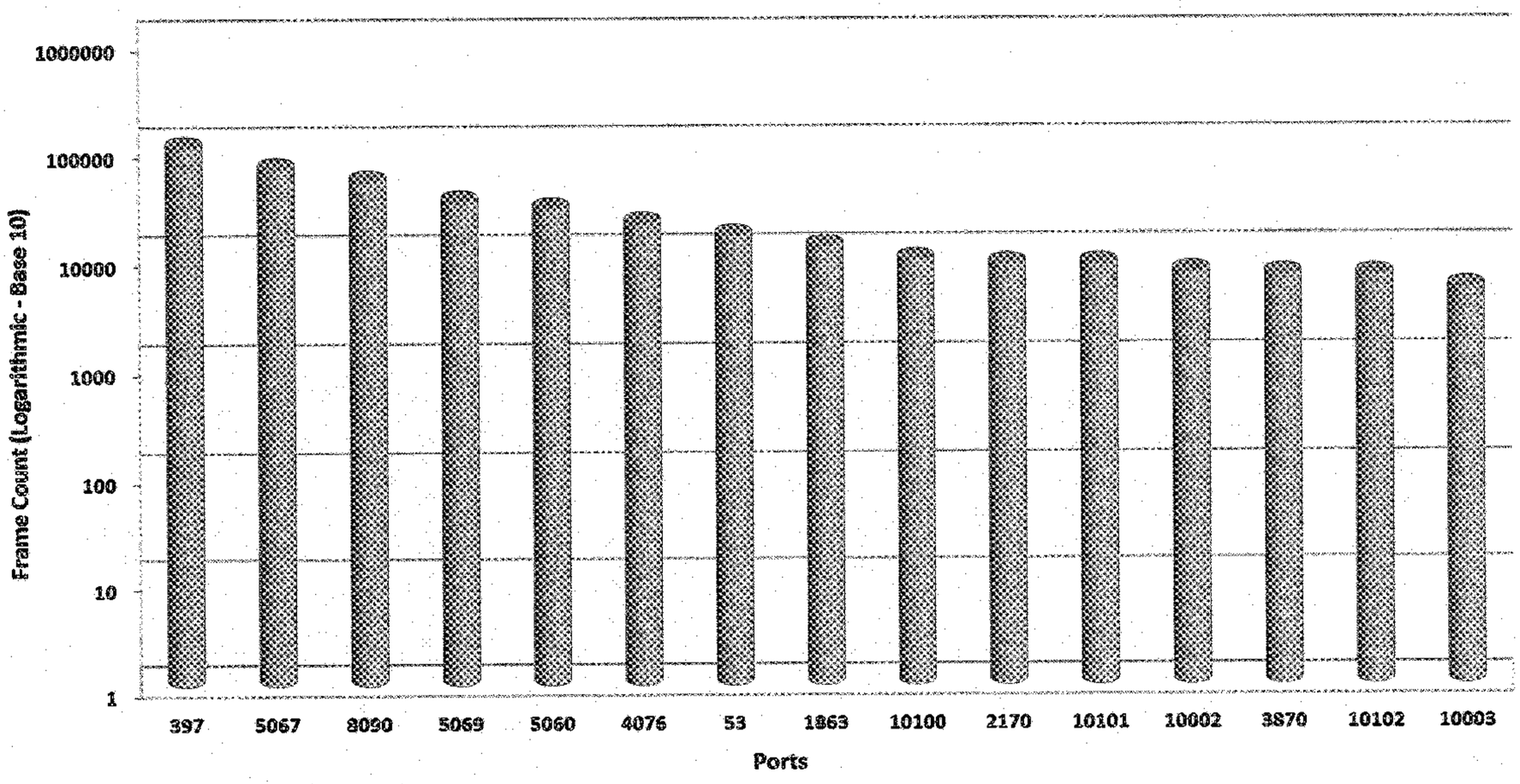


Figure 15: Top 15 UDP Source Ports Distribution

On the other hand, the distribution of UDP destination ports is depicted in Figure 16.

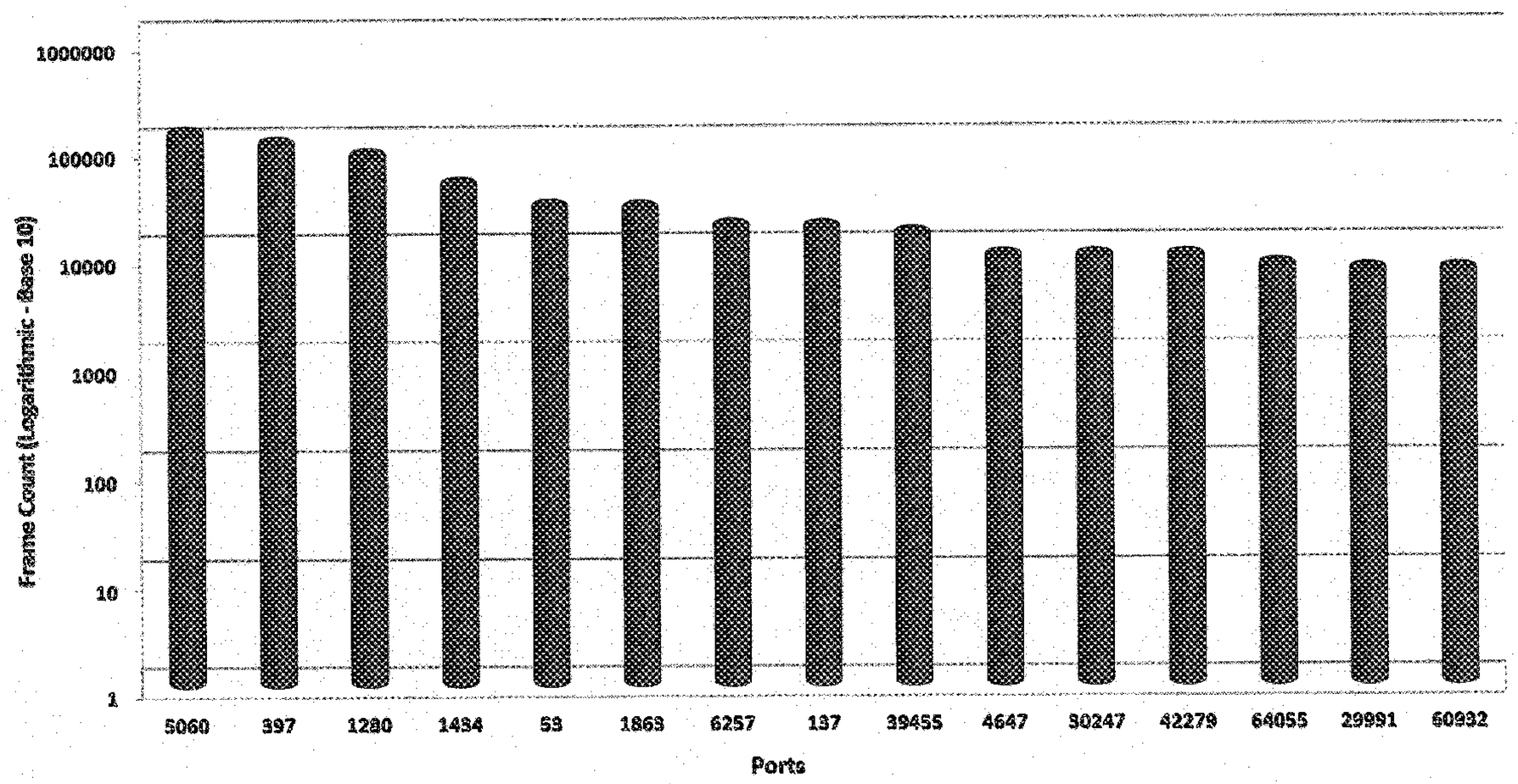


Figure 16: Top 15 UDP Destination Ports Distribution

Note that, the top three destination darknet TCP ports, namely, ports 445, 80, and 3389 are the Microsoft active directory service, the hypertext transfer protocol, and the Microsoft terminal server, respectively. These service ports have always suffered from security issues and vulnerabilities. A sample of the threats targeting such services are pinpointed in [57], [58] and [59], respectively. Hence, it is alarming that such ports appear as the top darknet destination TCP ports. On the other hand, the top three destination darknet UDP ports, namely, ports are the [60], respectively. The as mentioned in Section 5.2.1, is a significant target of attack. This result further validates the integrity of our results and insights. Moreover, the service are known to suffer from denial-of-service attacks when a malformed request is destined to them.

5.2.6 Operating Systems

In order to profile operating systems, we utilized P0f [18], which is a traffic fingerprinting tool. We managed to identify approximately the operating systems of 46% of the global traffic. Windows took the leadership with almost 45% of originating traffic. The operating systems statistics are as follows:

Windows	Linux	FreeBSD	Solaris	Novell	MacOS	Unknown
44.95%	0.14%	0.086%	0.063%	0.006%	0.004%	54%

Table 6: Operating Systems Used on Darknet

The above results may indicate that Windows is still the preferred platform for launching malicious traffic by cyber attackers.

[18] http://www.sans.org/security-resources/idfaq/p0f.php

5.2.7 Internet Service Providers

Figure 17 illustrates darknet source ISPs. The results reveal that
are leading. is ranked second
with 19.2% while ranks third with 16.2%.
is as well a major contributor to darknet traffic.

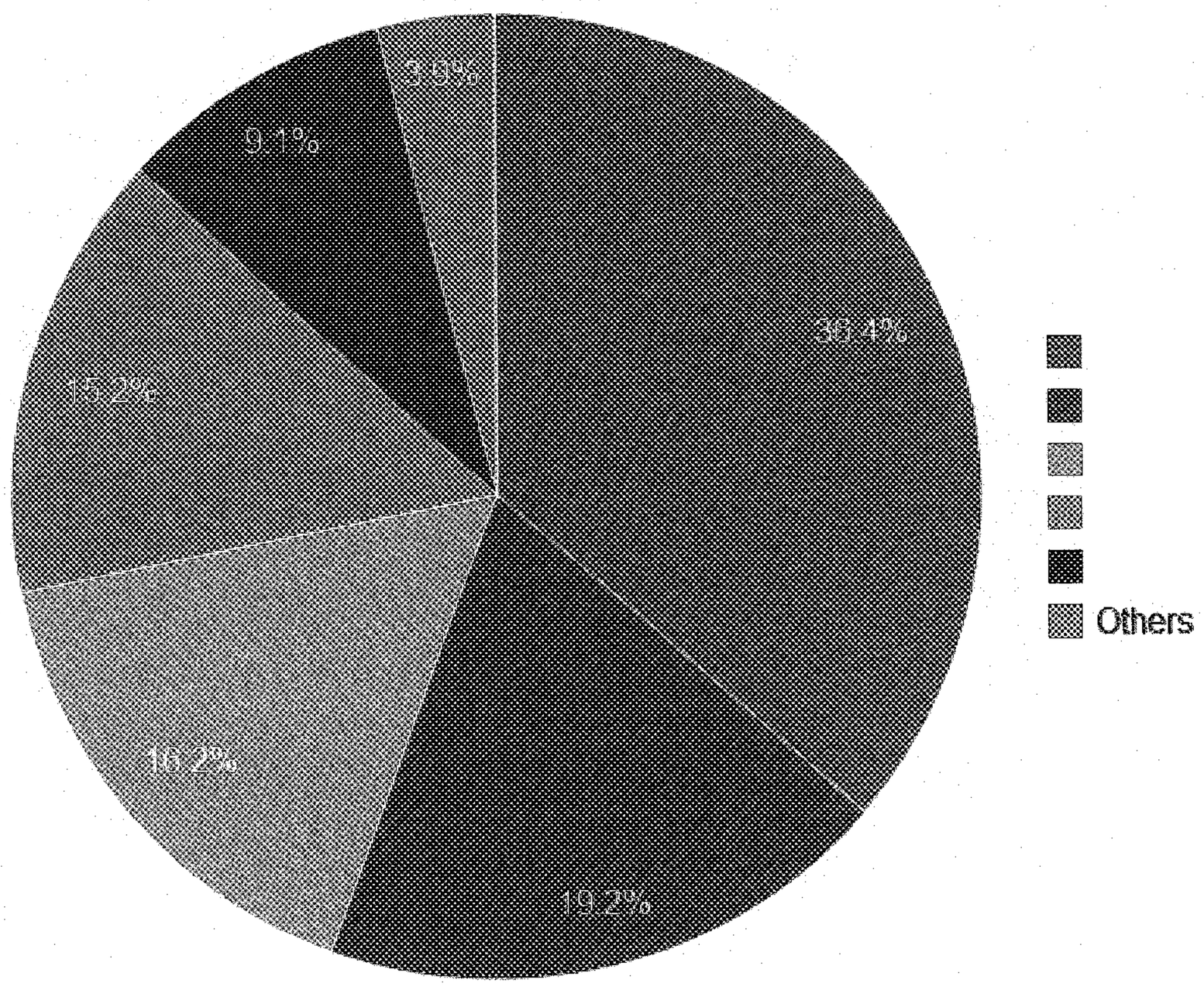


Figure 17: Darknet ISPs

5.3 Threat-Based Analysis

To inspect the existence of threats and attacks that aim at utilizing darknets as a transmission vector, we executed threat-based severity analysis on the darknet traffic. To accomplish that task, Snort [61] and Bro [62], two open source network intrusion prevention and detection systems (NIDSs), combining the benefits of signature, protocol and anomaly-based inspection, were investigated and implemented. Part of their content signature detection, Snort and Bro implement the Boyer-Moore (BM) exact string matching detection algorithm in addition to a non-deterministic finite automata regular expression (NFA RegEx) detection algorithm.

The Boyer-Moore algorithm, which is known to be very fast in practice, performs character comparisons between a character in the text and a character in the pattern from right to left. After a mismatch or a complete match of the entire pattern, it uses two shift heuristics to shift the pattern to the right. These two heuristics are called the occurrence heuristic and the match heuristic [63]. Note that the length of the shift is the maximum shift between the occurrence heuristic and the match heuristic. Additionally, these heuristics are pre-processed in $\Omega(m + |\Sigma|)$ time and space where m is the pattern length and Σ is the alphabet. Furthermore, the searching phase of the BM algorithm requires O(n×m) time in the worse case where n is the text length. Finally, the expected performance of the BM algorithm is sub-linear requiring about $\frac{n}{m}$ character comparisons on average [64].

On the other hand, the NFA RegEx algorithm is excessively utilized since it is known to be space efficient. A non-deterministic finite automaton is a mathematical model that consists of:

1. A set of states S
2. A set of input symbols Σ (the input alphabet)
3. A transition function that maps state symbol pairs to sets of states
4. A state s_0 that is distinguished as the start (or initial) state
5. A set of states F distinguished as accepting (or final) states

An NFA accepts an input string x if and only if there is some path in the transition graph from the start state to some accepting state, such that the edge labels along this path spells out x. A path can be represented by a sequence of state transitions called moves.

To obtain the threat-based results, we fed the darknet data to the NIDSs. The outcome of this procedure is summarized in Table 7.

Threat	Type	Priority
t_1		High
t_2		
t_3		
t_4		Medium
t_5		
t_6		
t_{7-30}		Low

Table 7: Darknet Threats and Corresponding Severities

The results reveal 30 distinct threats. According to the NIDSs, three threats are of high priority, the other three are of medium severity and the rest are of low priority. The first high priority threat (t_1) is in fact an attempt to

The other high priority threat (t_2) is rendered as an attempt to

The last high priority threat (t_3) is in reality

On the other hand, threats t_4, t_5 and t_6 are according the NIDSs of medium severity. Threat (t_4) represents an attempt to

Moreover, (t_5) is

The last medium priority threat (t_6) is an effort to

For the purpose of contributing to high-level attribution, we perform geo-localization of the threats sources. Figure 18 depicts the heat map. Note that, the threat count metric is of the order of thousands. The results reveal that lead in terms of number of triggered darknet threats with more than 40,000 threat counts. is as well major threat contributors.

Figure 18: Threats Sources - Heat Map (Order of Thousands)

In order to refine obtained results, we decided to expose the sources of threats.
Source of Threats-Countries: The source of threats per country is demonstrated in Figure 19.

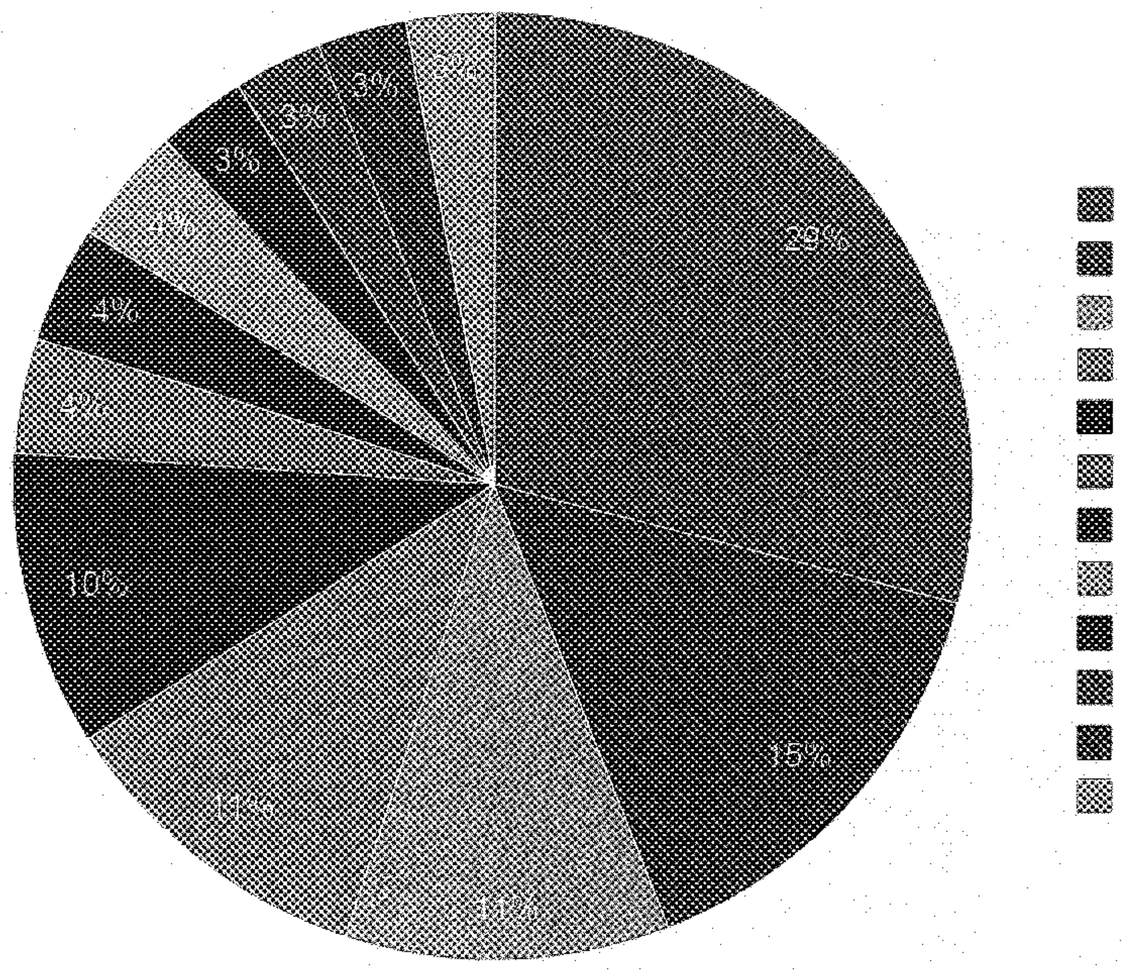


Figure 19: Source of Threats-Countries

Analysis of Darknet Space for Predictive Indicators of Cyber Threat Activity

Source of Threats-Domains: The graph of suspicious domains is revealed in Figure 20. net.br and com.cn are leading. primus.ca constituted 12.3% of the total threats of the analyzed traffic.

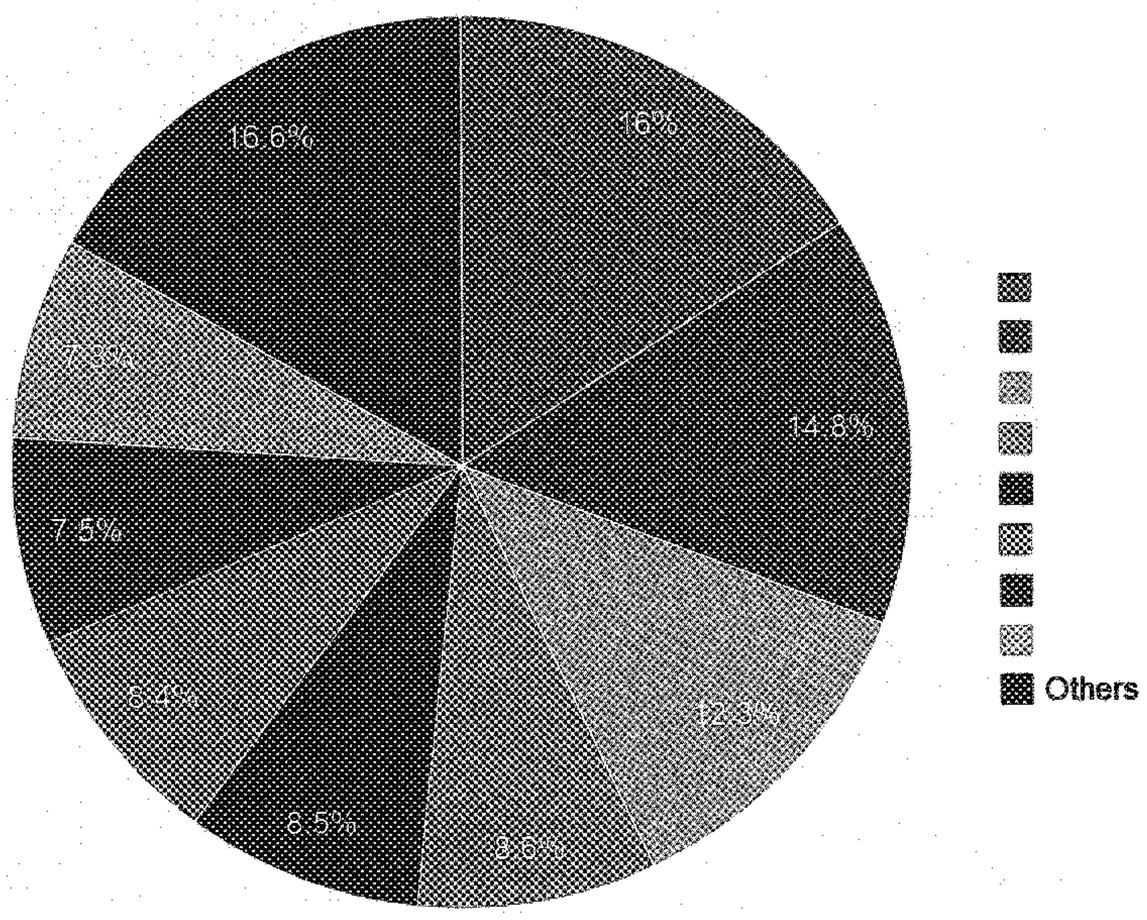


Figure 20: Source of Threats-Domains

Source of Threats-ISPs: The vulnerabilities pie chart based on ISPs is shown in Figure 21.

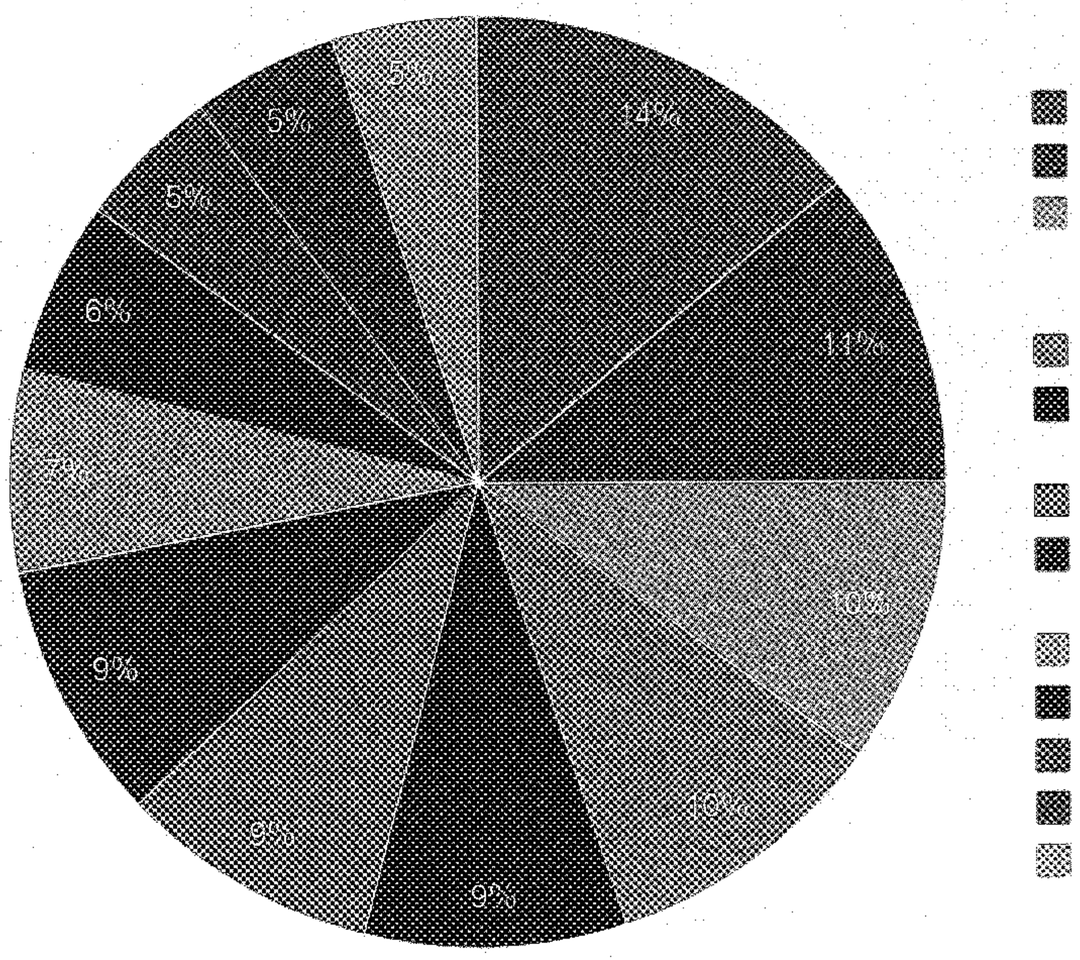


Figure 21: Source of Threats-ISPs

were among the top ISPs source of darknet threats.

Target of Threats-ISPs: We, as well, considered target of malicious activities on the dark channel. Hence, we have generated the target ISPs pie graph as shown in Figure 22. It was noticeable that currently owned by the and the constitutes more than 50% of threats' targeted ISPs.

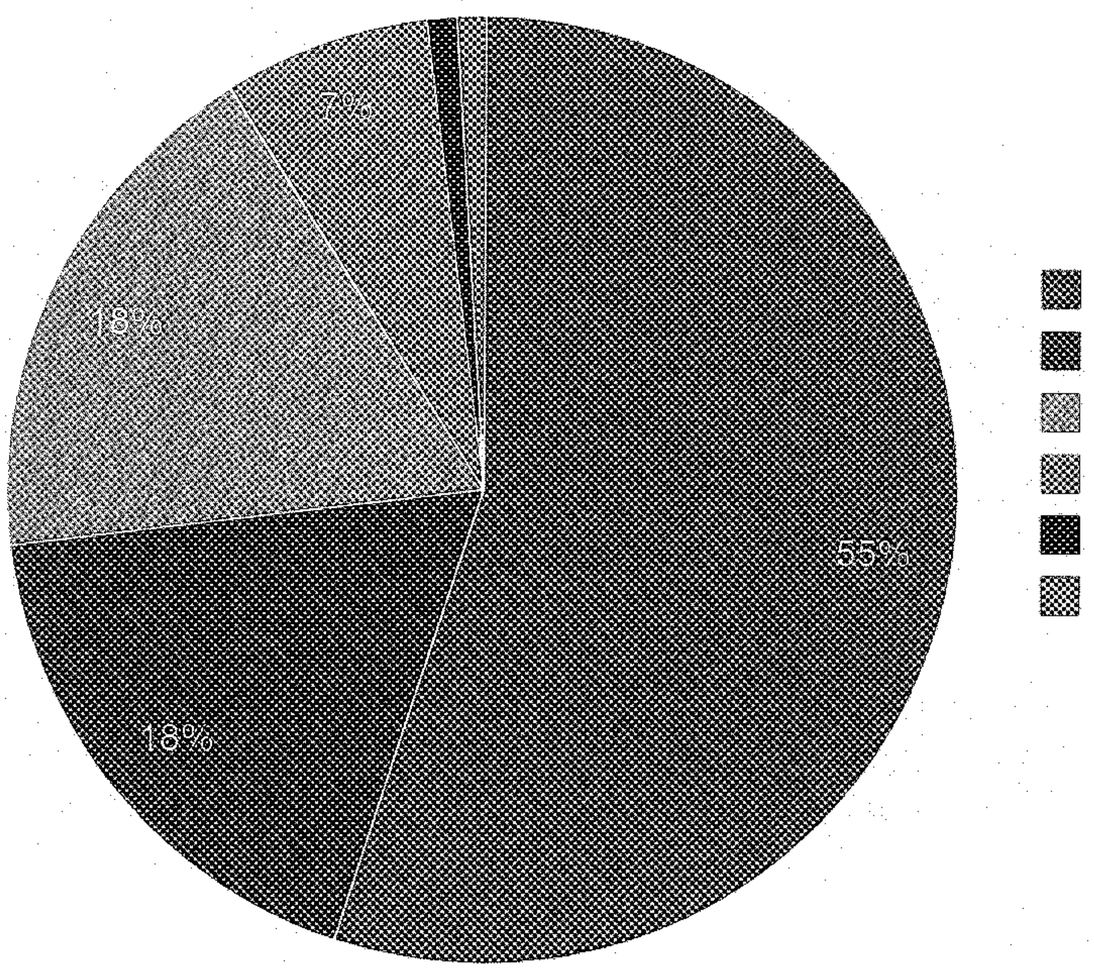


Figure 22: Target of Threats-ISPs

6 Detection of Malicious Domains and IPs

In this section, we introduce a cyber-intelligence framework that allows the detection of malicious domains and IPs. This framework is based on malware dynamic analysis and passive DNS data. We aim at detecting dark IPs from such intelligence in order to find suspicious fragments of data sent through darknet channel. In the sequel, we present an overview of our cyber-intelligence framework as well as the various obtained results and findings related to malicious IPs and domains.

6.1 Framework Overview

Mainly, our framework aims at generating relevant and important cyber-intelligence from automatic malware analysis and passive DNS data. It is composed of four components, namely,

Figure 23 illustrates the different components that constitute our framework. The framework has two relevant inputs: malware feeds and passive DNS data.

Figure 23: Cyber-Intelligence Framework Overview

6.1.1 Data Collection

The collected data is of two sorts: malware feeds and passive DNS data.

- Malware Feeds:

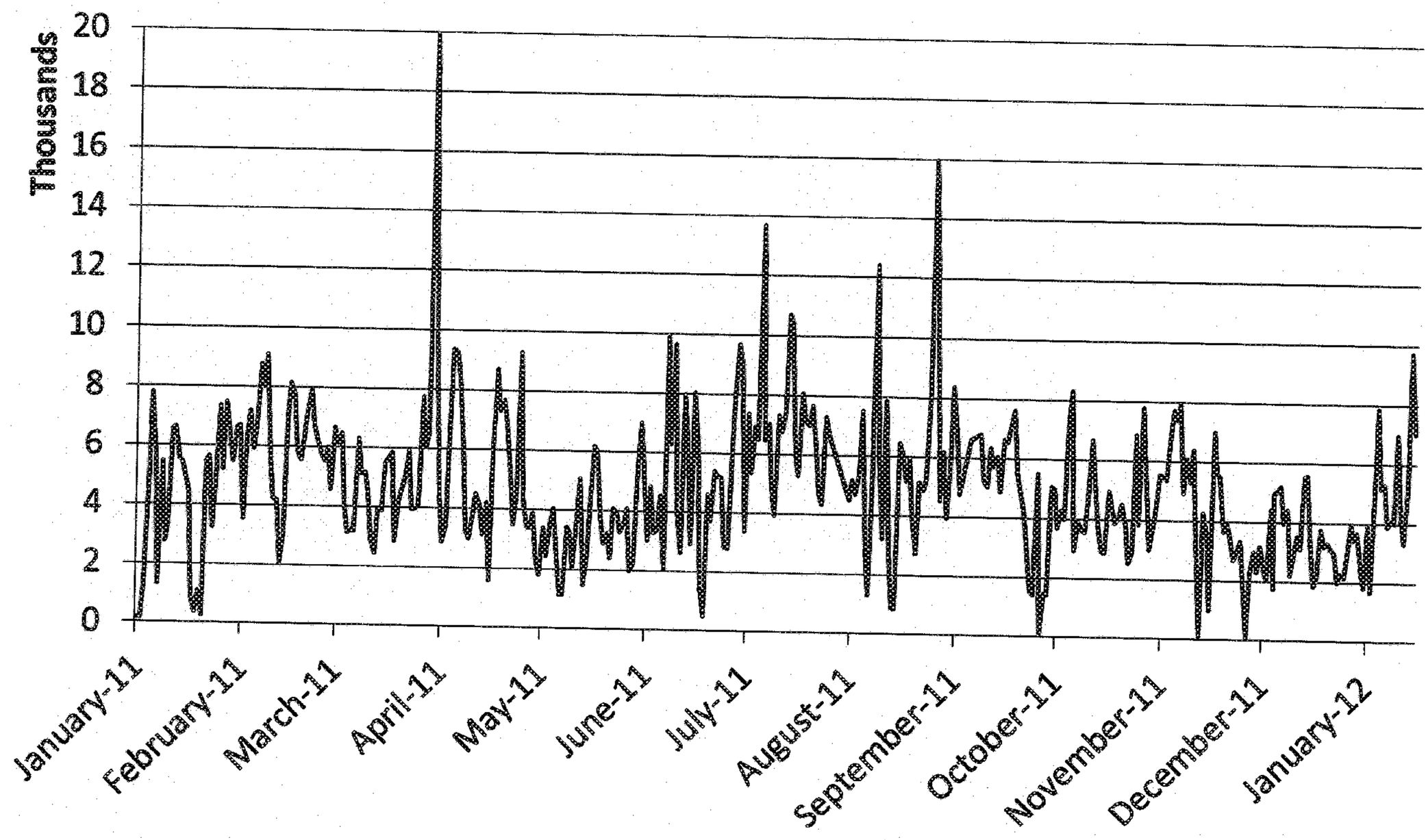


Figure 24: Malware Counts Graph

The malware data collection in our framework is based on GFI ThreatTrack feeds [68]. We receive malware samples on a daily basis. These malware information are indexed in files and

We observed the evolution of these feeds over time. We received 1794239 malware with an average of 3601.5 malware per day. Figure 24 illustrates the malware counts recorded during 368 days. We notice that the feeds represent a good source of intelligence since it reached a peak of 20143 malware. However, we observed during the malware collection period that some counts were less than 100. Such exceptions are insignificant since they occurred only three times during one year.

- Passive DNS Channel:

The domain name service (DNS) has a significant role in the Internet since it maps between domain names and their IP addresses. Given its essential role, it is more likely that malicious activities involve DNS requests. For example, bots can resolve names that points to their command-and-controls or proxies. Thus, it is a must to monitor DNS activities to detect suspicious actions that

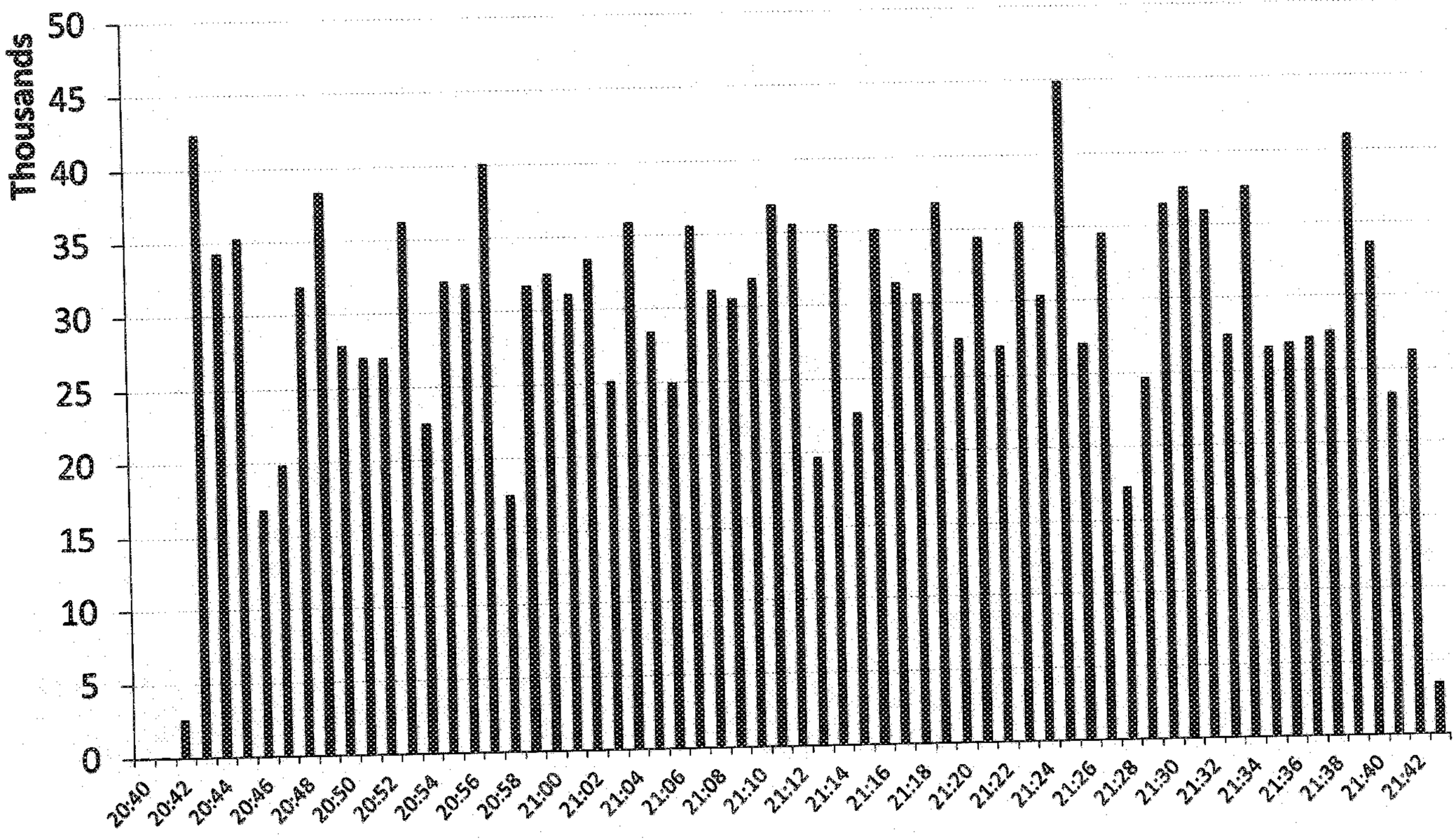


Figure 25: Passive DNS Records

are perpetuated by malware. Moreover, it is an essential artifact to corroborate malware dynamic analysis since it can be bound and correlated with it. In our framework, we used DNS feeds that are received from different sensors. An initial step was to create an offline capture of passive DNS raw data to get an insight about it. This capture represents one hour data, which constitute a size of 2.7 GB. Figure 25 represents the numbers of DNS requests that have been recorded. The captured data shows richness in terms of information. We observed 2168606 domains and IPs, where 998707 are unique domains or IPs. These feeds enclose Passive DNS records, namely, query responses, unanswered queries and unsolicited responses. The average number of query responses is approximatively 500 responses query per second whereas the average of unanswered queries is 76 queries per second. The unanswered queries ratio stands for 15% of the average query responses. The average of unsolicited reponses represents 9 unsolicited reponses per second, which stands for a ratio of 1.8% of query response.

6.1.2 Malware Dynamic Analysis

There exists a large variety of malware in the wild such as viruses, worms, spyware, rootkits, Trojan horses, bots, etc. All these malware share common characteristics that are confined mainly in performing unwanted malicious activities. An analysis of such malicious program is a must. It allows to understand the behavior of malware and their severity. In order to accomplish that, we decided to use a dynamic analysis of malware technology and integrate it in a cyber-intelligence framework

So far, we managed to obtain around 1.7 million reports from malware analysis during one year (January 2011 to December 2011). The identification and assessment of cyber-threats allow analysts to dissect potential attacks as well as adapting preemptive security strategies. On that basis, we elaborated on malware analysis reports to identify relevant information.

This module is still under development where an ongoing effort is done to enrich it with new features, namely,

6.2 Results

In this part, we introduce the different information and statistics that we managed to obtain from our cyber-intelligence framework. We built a front-end to show the different findings. In the sequel, we present the different findings, which fall into worldwide and Canadian statistics.

- Geo-localization: Our front-end has a geo-localization capability that allows geo-localizing malicious IPs and domains. Figure 26 illustrates a world map containing suspicious IPs and domains

Figure 26: Domains and IPs Geolocation

Figures 27 and 28 illustrates geo-localization information related to collected IPs and domains in Canada.

Figure 27: Domains and IPs Geolocation

s.16(2)(c)

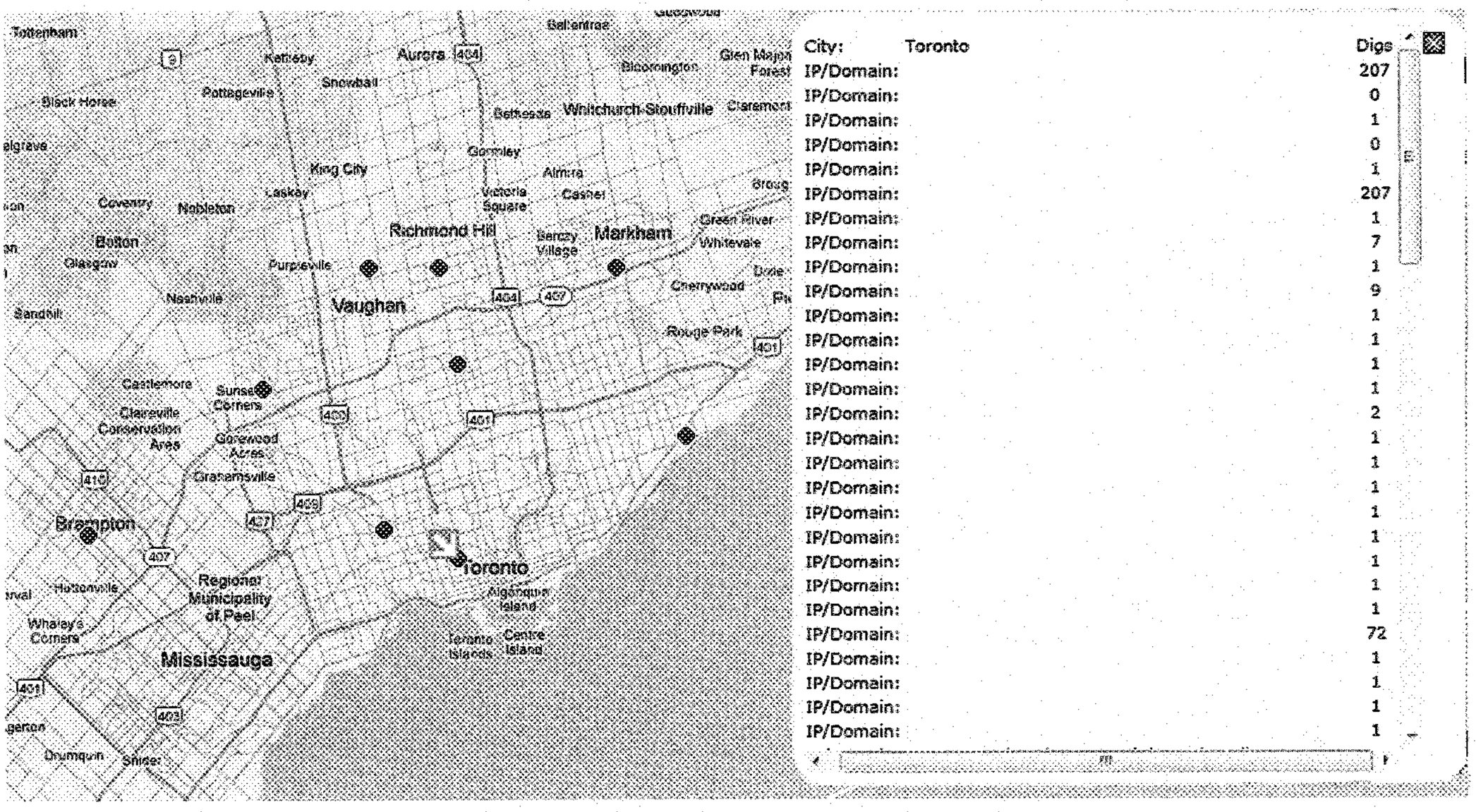


Figure 28: Domains and IPs Geolocation

- Domains and IPs: The total number of extracted suspicious IPs is 39,691 where the majority is located in (15,507), followed by (12,056) and (7,636). Regarding the domains, the total number is 41,140 where the majority is located in (12,903), followed by (12,972) and (10,331). Figure 29 shows the number of suspicious IPs and domains per continent.

s.16(2)(c)

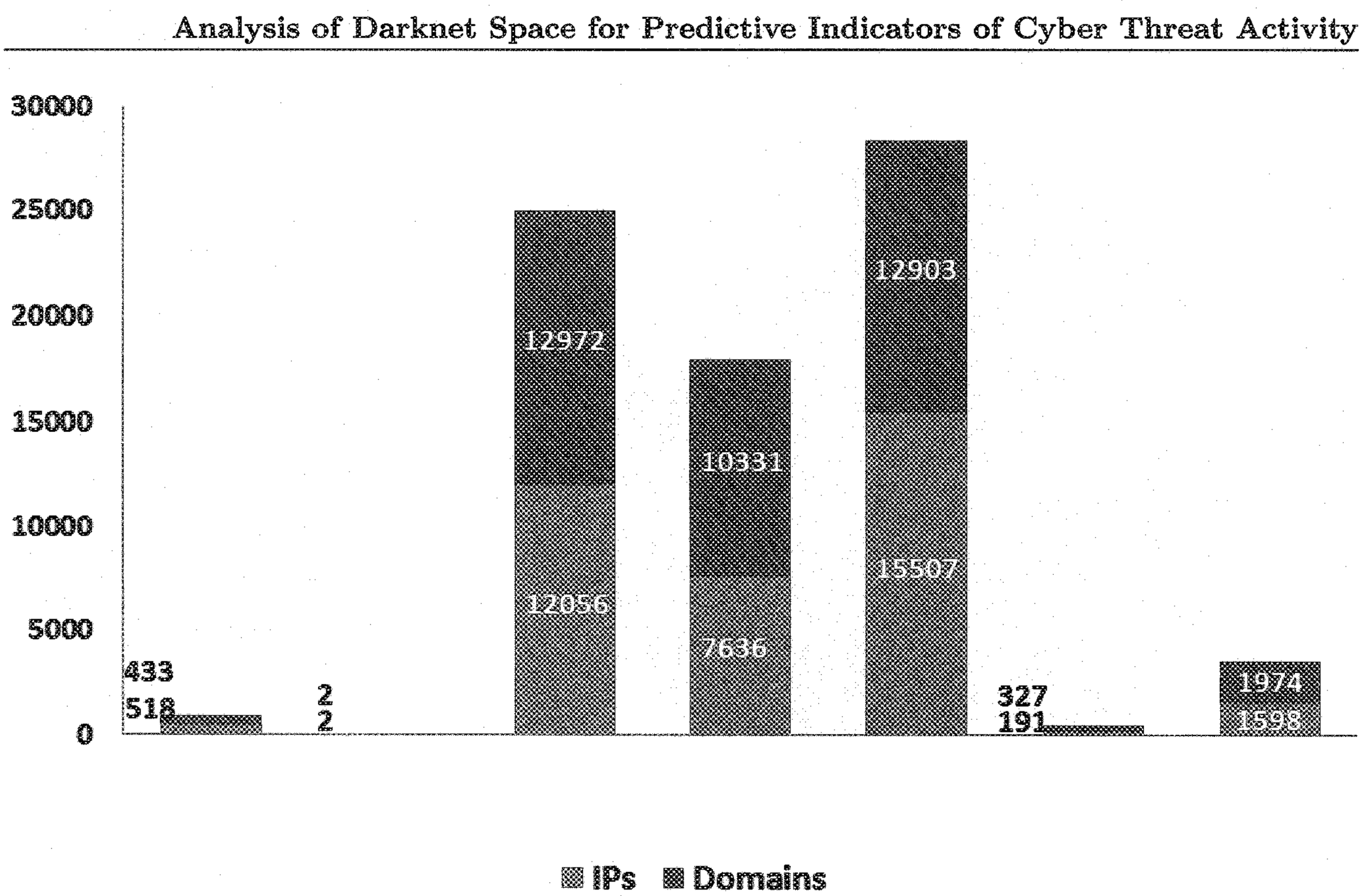


Figure 29: Domains and IPs Distribution per Continent

- Drop Locations: Regarding drop locations, we recorded 6,948 distinct FTP connection records. Among these connections, we enumerated 1,376 distinctive drop locations. Figure 30 illustrates the distribution of FTP drop locations throughout the continents.

s.16(2)(c)

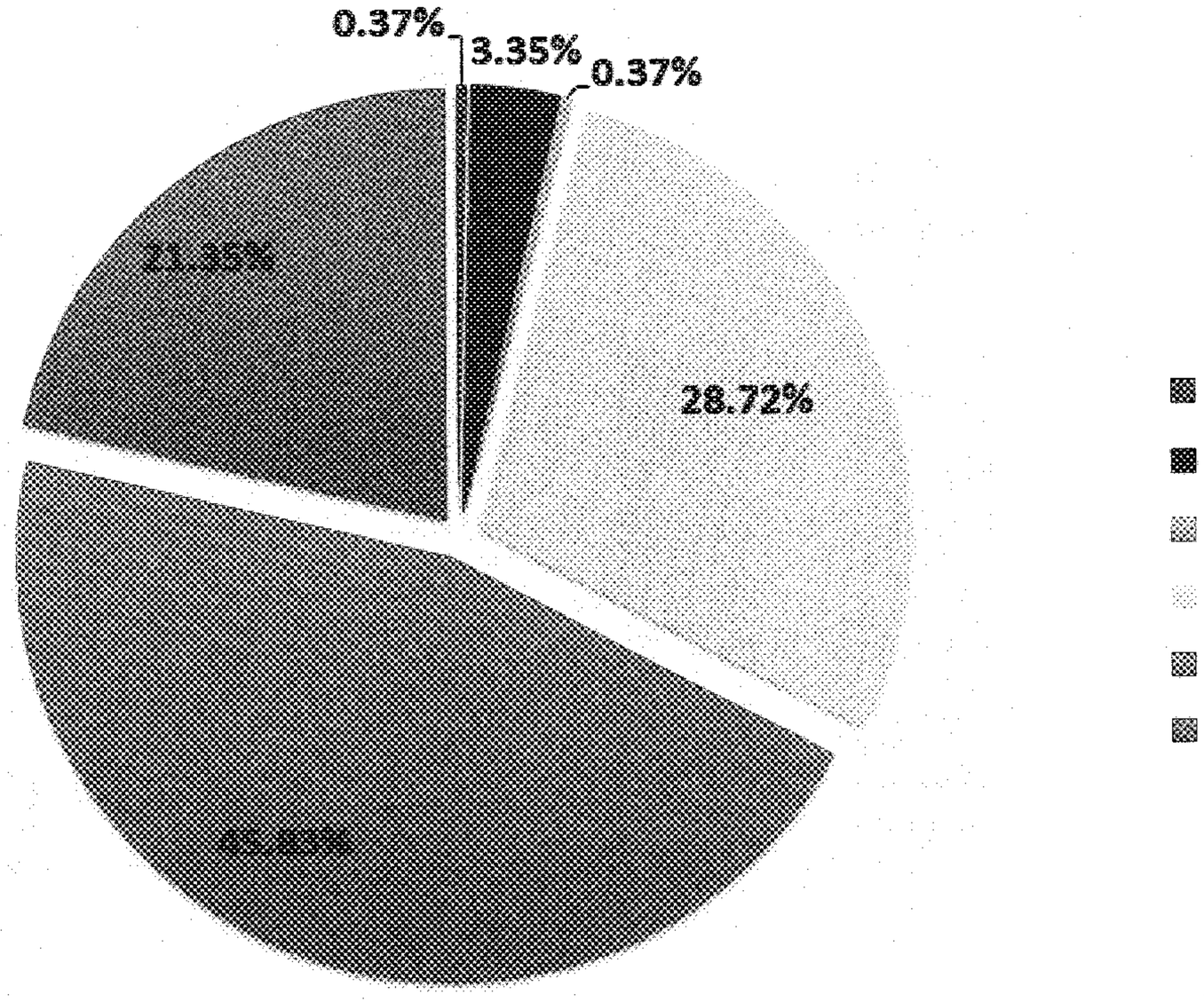


Figure 30: Drop Location Distribution per Continent

Canada has 46 distinctive drop locations. Figure 31 shows the distribution of drop locations in Canadian cities. constitute the main cities where drop locations are more active with more than 60% of all drop locations.

s.16(2)(c)

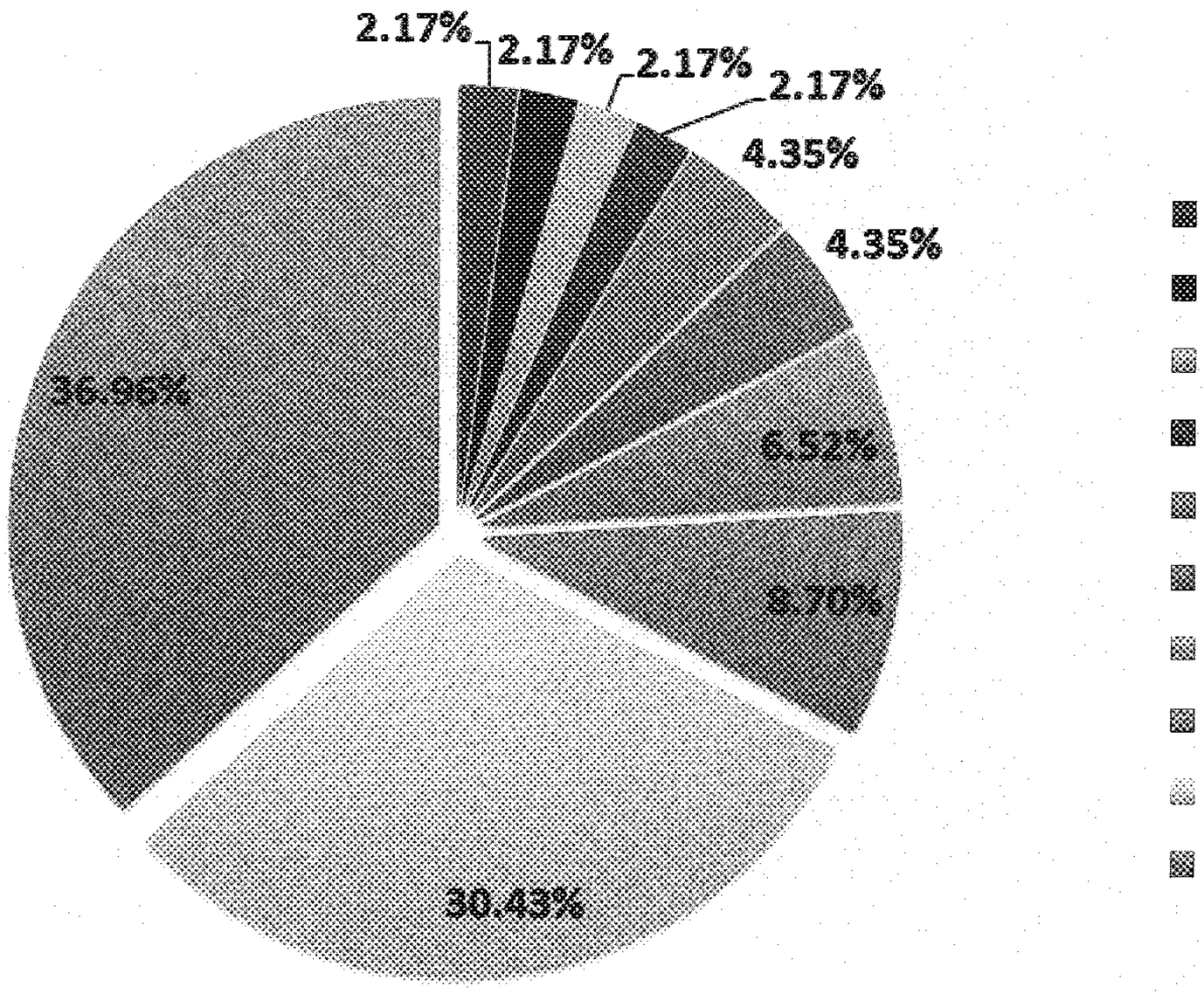


Figure 31: Drop Locations Distribution in Canada

- SMTP Connections: For SMTP connections, we enumerated 1316 connections where each connection corresponds to an attempt of email sending for spam campaigns or malware propagation. Figure 32 illustrates SMTP connections distribution per countries.
  constitutes top countries. constitutes 2.66 % with 35 SMTP connections.

s.16(2)(c)

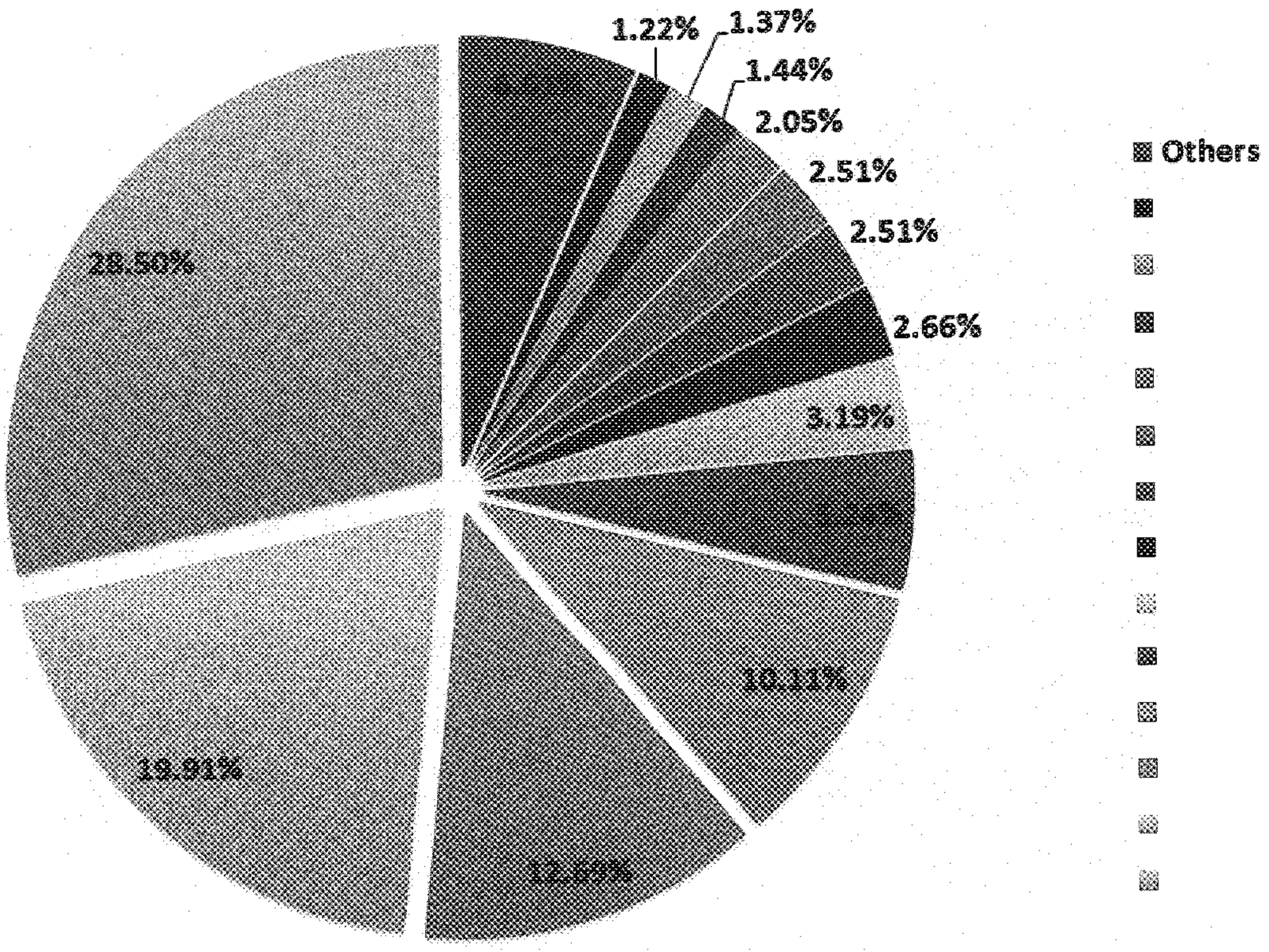


Figure 32: Malware SMTP Connections Distribution per Country

- IRC Channels: IRC channels are good media for botmasters to conduct targeted attacks by sending commands to bots. We obtained 8,179 IRC connections with their corresponding credentials. Figure 33 illustrates the distribution of IRC countries per countries.
  are the countries that have the majority of IRC channels. has 1.09% of the total IRC connections.

s.16(2)(c)

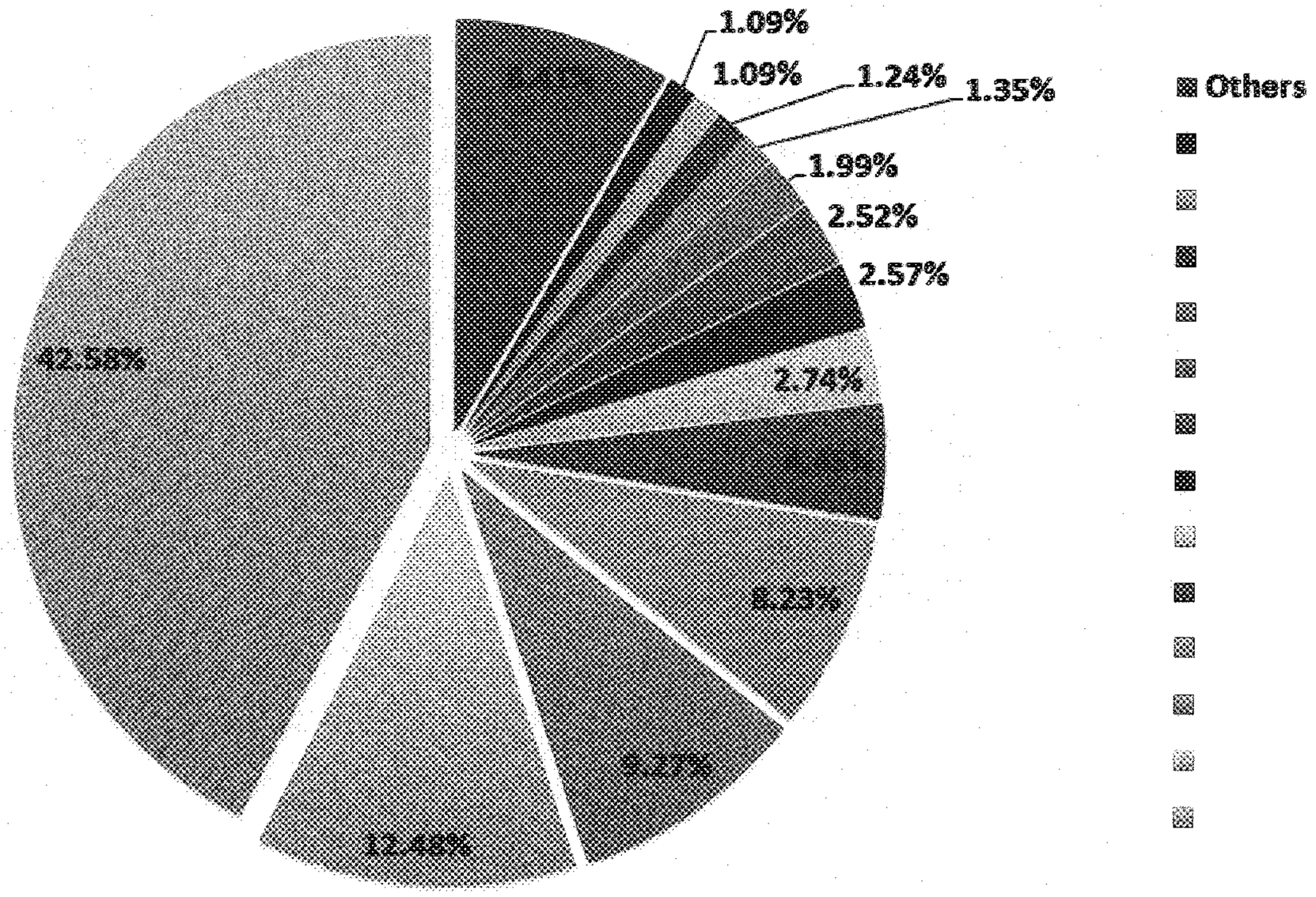


Figure 33: Malware IRC Connections Distribution per Country

6.3 Conclusion

Based on the ongoing malware-based analysis, we expect to finish building a dynamic passive DNS correlation module as well as a severity scoring mechanism for domains and IPs by applying data mining techniques. Meanwhile, we try to collect interactively continuous darknet data and extract IPs and domains. These latter would be fed to scripts to see whether these sources of connections are recorded within the malware dynamic analysis. If this is the case, we can grab suspicious connections into the darknet channel and analyze them.

7 Brand Protection Services

This section describes a system that provides security services in the area of brand protection. The main objective of this system is to correlate different pieces of information and generate relevant and actionable intelligence to assist financial institutions, Internet service providers (ISPs), and other corporations in mitigating online frauds and cyber crimes.

7.1 Background Information

In recent years, cyber crimes have received a lot of attention due to their impact on the safety and security of individuals, organizations as well as the critical infrastructure of nations. As for business organizations, the online reputation of their brand names and products had become an indispensable asset. Online threats such as cybersquatting, phishing, malware, piracy and others have caused significant damage to organizations in terms of money, reputation, trust and goodwill. Mark Monitor, which is considered as a leader in online brand protection, has reported more than 1.2 million sold unlicensed jerseys online by 6,000 suspects, that generated a revenue of more than $25 million per year through 56 million illicit visits. Mark Monitor has categorized different types of online threats into 17 classes, which are:

- Brandjacking
- Counterfeiting
- Cybersquatting
- Ecommerce Content
- False Association
- Grey Market Selling
- Malware
- Offensive Content
- PPC (Pay-Per-Click) Abuse
- Phishing
- Piracy
- Rock Phishing
- SEO Manipulation
- Search Engine Marketing Abuse
- Spam

- Traffic Diversion
- Typosquatting

Among those online brand abuses, phishing and malware are currently the two most serious threats targeting financial institutions (FIs).

Mark Monitor defines phishing as "Criminal use of email to divert traffic to websites in order to fraudulently acquire usernames, passwords, credit card details and other personal information. The email and websites used in these operations employ social engineering techniques to trick users into believing they are interacting with a business or organization that they trust." [70]. Phishing is one particular online threat to FIs where they depend heavily on the Internet to provide financial services to their customers. Due the convenience of using online services, FIs are forced to invest heavily on improving the security of their online services. Although organizations can ensure, to some extend, that their system is secure, however, they cannot assert the security of their customers' system. The latter is the weakest link in the whole process and it is effectively exploited in phishing attacks. Inexperienced Internet users tend to easily fall into such misdemeanors and provide all their online banking credentials to the attackers. The most obvious method to mitigate phishing attacks is to educate Internet users about the threat and pinpoint the various tricks attackers often use. However, even experienced users will fall into the most sophisticated attack methodologies. Therefore, FIs and ISPs must further react by shutting down as many phishing sites as they can detect. Security experts often set up fake email addresses, dubbed as spamtramps, and make them visible on the Internet to attract advertising and malicious emails (spam). These spamtraps will act as sink, collecting most emails with embedded phishing URLs. ISPs will then attempt to take those phishing pages down. In recent years, phishing attacks have attracted the attention of security experts and organizations. Anti-Phishing Working Group [71] and PhishTank [72] are currently the two most relevant and effective services working against phishing attacks. Well-known browsers like Microsoft Internet Explorer, Mozilla Firefox and Google Chrome also claim that they have built-in anti-phishing mechanisms based on blacklisting and heuristic methods [73] [74] [75]. Although they have made a noteworthy effort in an attempt to protect Internet users from phishing attacks, there is still a lag between the time when a phishing page is online and when it is taken down or is loaded into blacklists. Currently, there exist no systematic methods to measure how much damage a phishing site can caused or to detect a future phishing attack.

Malware, on the other hand, has always been a notable problem in information security. In the area of online brand protection, malware (malicious software) is defined as "the occurrences when an illicit site or email attachment installs malware, viruses, keyloggers or other software that automatically steals usernames, passwords, and additional private information without users' knowledge." [70]. Malware has received intensive attention from security experts and has been studied thoroughly over the years. However, malware has evolved from entertaining attempts of individuals into a serious business involving criminal organizations, underground online marketplaces and even governments. The remarkable benefit from creating, selling and using malware has motivated malware authors to create new and feature-rich malicious software which employ various complex techniques to evade detection and are often refined to target a small specific "niche" . An example of this would be Stuxnet, which targeted high-value Iranian assets [76] or Trojan bankers, which target FIs. [77]. Currently, FIs are extremely concerned

about malware that are precisely engineered to target their infrastructures, their online services or their customers. Consequently, the requirement to identify those specific types of malware in a swarm of new malware that appear everyday (over 50,000 new malware samples) is in fact a very challenging task.

7.2 System Overview

The system consists of three main components: Data acquirement, back-end data process and front-end data presentation. Figure 34 illustrates those three system sub-parts.

Figure 34: System Architecture

7.2.1 Data Acquirement

To detect phishing URLs, the system relies on two sources of data: spamtraps from SIE and ThreatTrack from GFI.

Spamtraps: Spamtrap is defined as an address or set of addresses that are used to capture spam in order to provide information on the source and content of sent spam. Spamtraps hence, do not belong to real users but instead are decoys set up to collect spam and montior and identify spammers. [78]. The spamtraps data, which the system uses, is retrieved from SIE [79]. It is received on the fly as a stream of network packets, where each packet represents a formatted spam email that includes its SMTP message header and corresponding embedded body URLs. In the testing stage, we have experimented using 50 GB of spamtraps data, which is collected in different time frames. Figure 35 represents the spamtraps data we collected from 15:00 on November 28th 2011 to 14:00 on November 29th 2011. During 23 hours, *25,384,564* spamtraps were recorded containing *51,452,112* URLs in which *18,028,303* consisted of unique URLs.

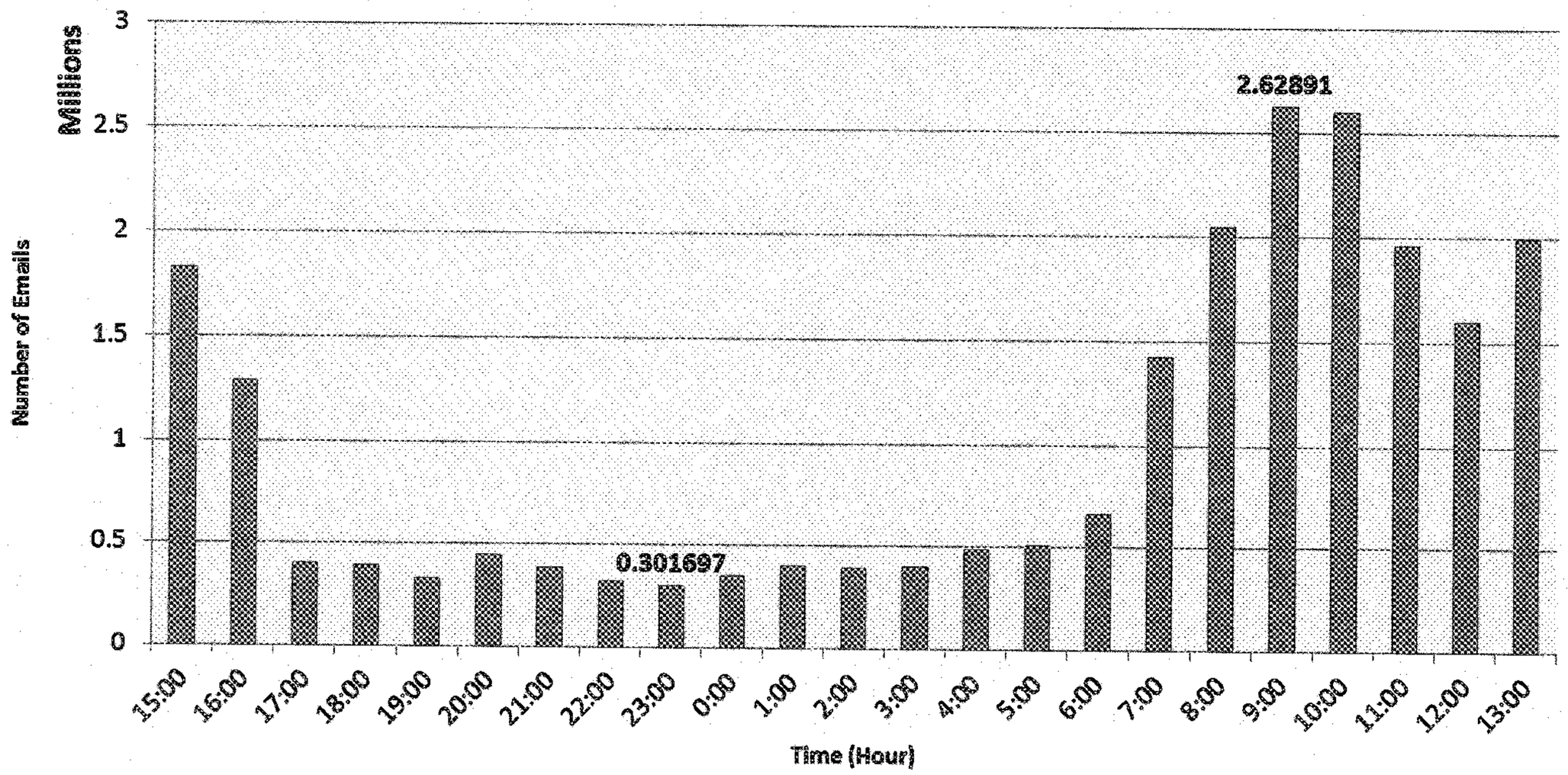


Figure 35: Spamtraps Data - 15:00 November 28th 2011 to 14:00 November 29th 2011

Figure 36 exhibits other spamtraps data that we recorded from 2:00 on January 31th 2012 to 2:00 on February 2nd 2012. Within 48 hours, *6,467,407* spamtraps were saved, consisting of *10,686,424* URLs (*2,598,932* unique URLs). We observed that spammers tend to distribute a lot of spam emails during working hours and slow down at the end of the day.

s.20(1)(c)

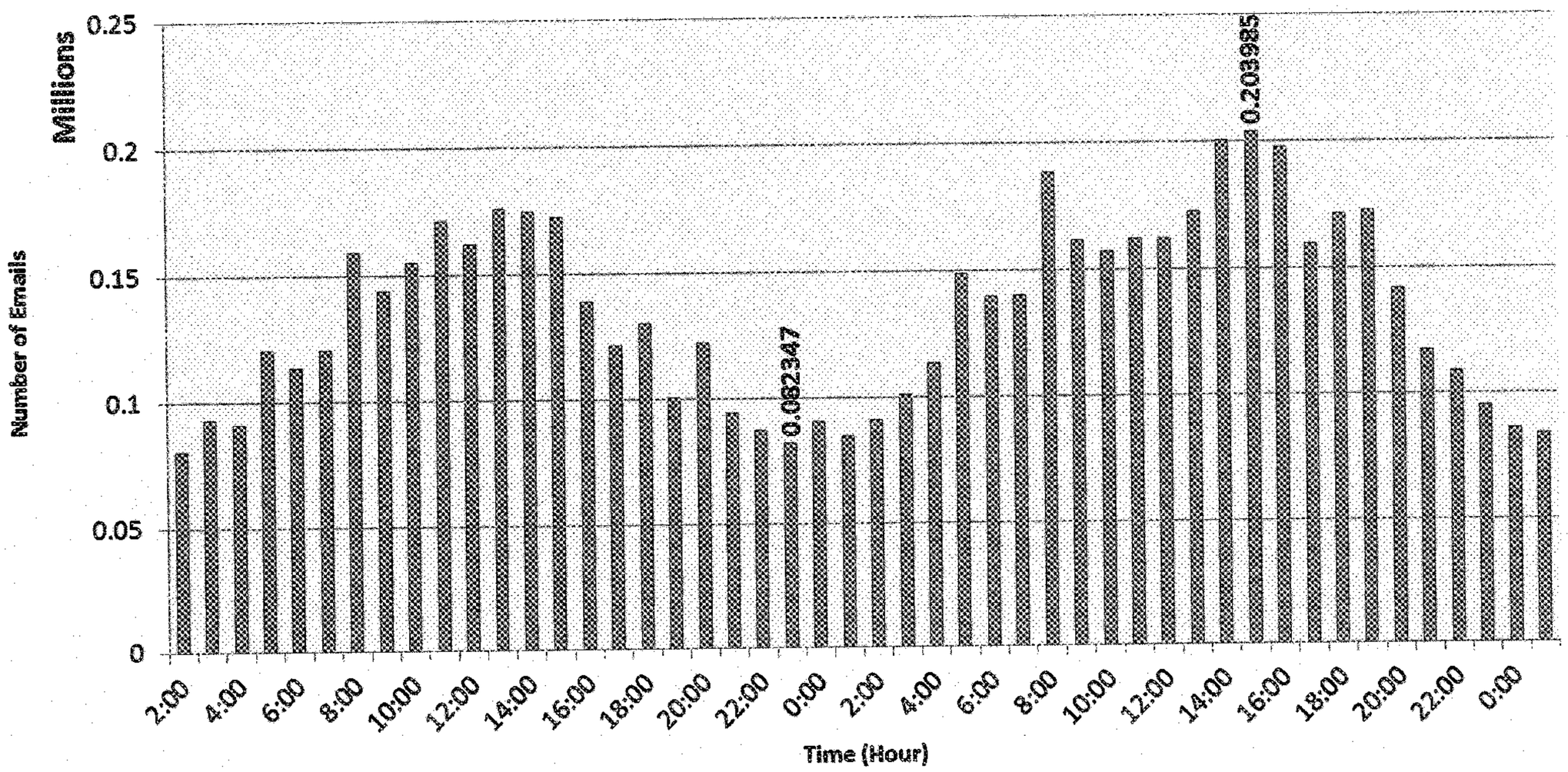


Figure 36: Spamtraps Data - 2:00 January 31th 2012 to 2:00 February 2nd 2012

ThreatTrack: ThreatTrack™ is a data sharing service from GFI composed of many security data feeds [80]. One of those feeds is the "Threat Track - categorized URLs" which, in summary, is a daily/hourly posting of malicious and unwanted URLs/IPs in 4 categories. The URLs in the "Phishing" category are characterized by GFI as "contain characteristics of phishing techniques: transposition, misspellings, common phishkit paths, and other phishing keywords". The "Phishing" category also includes reported phishing URLs from and other 3rd-party feeds. Figure 37 shows the number of phishing URLs in ThreatTrack per month in 2011, which has a total of *180,457* phishing URLs. On the other hand, Figure 38 displays the number of phishing URLs during the first three months in 2012, a total of *59,330* phishing. The system also uses two other data sources: passive DNS and malware samples. Those two data sources are described in detail in section 6.1.1.

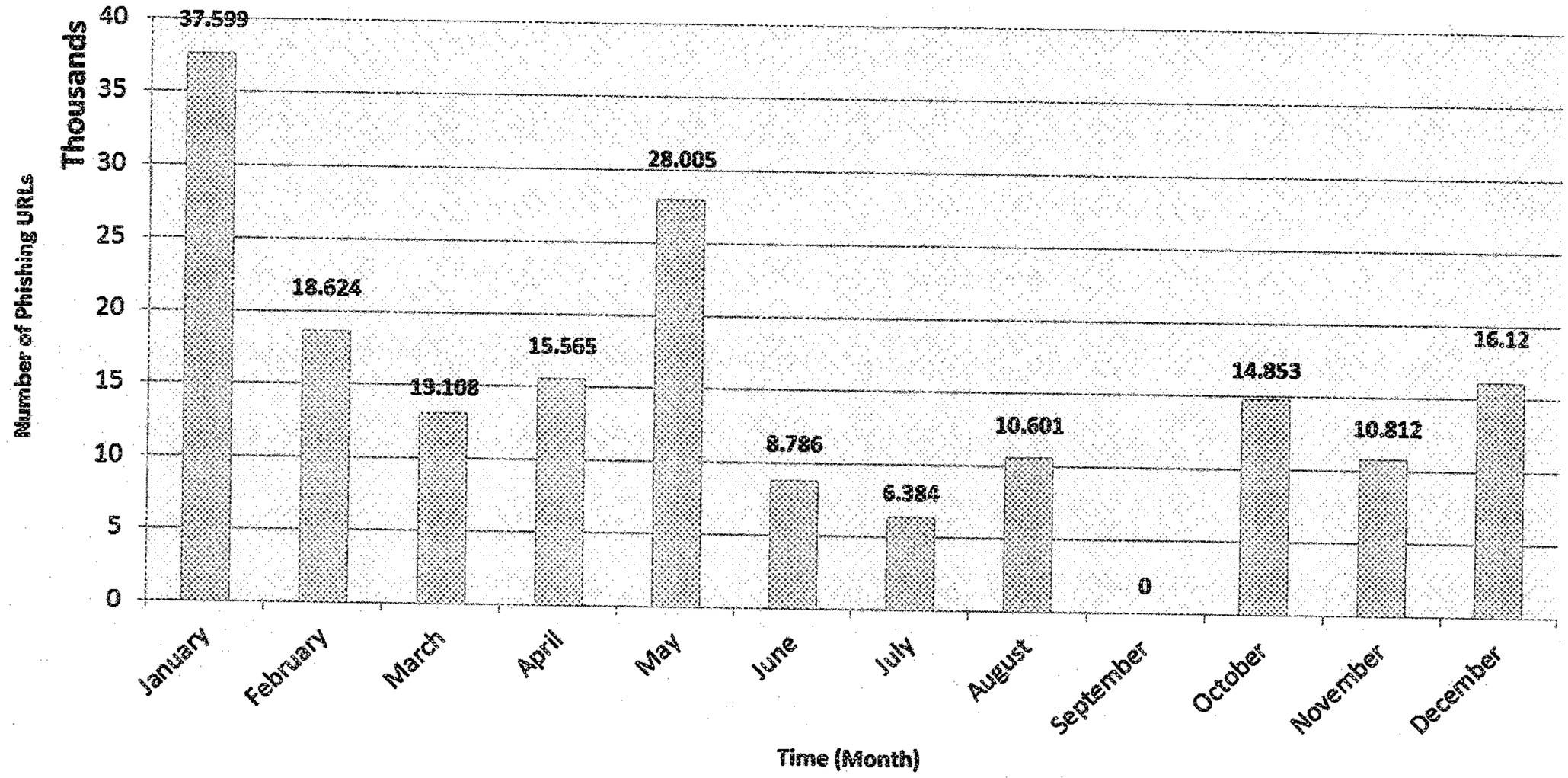


Figure 37: ThreatTrack™ Phishing URLs Feed in 2011

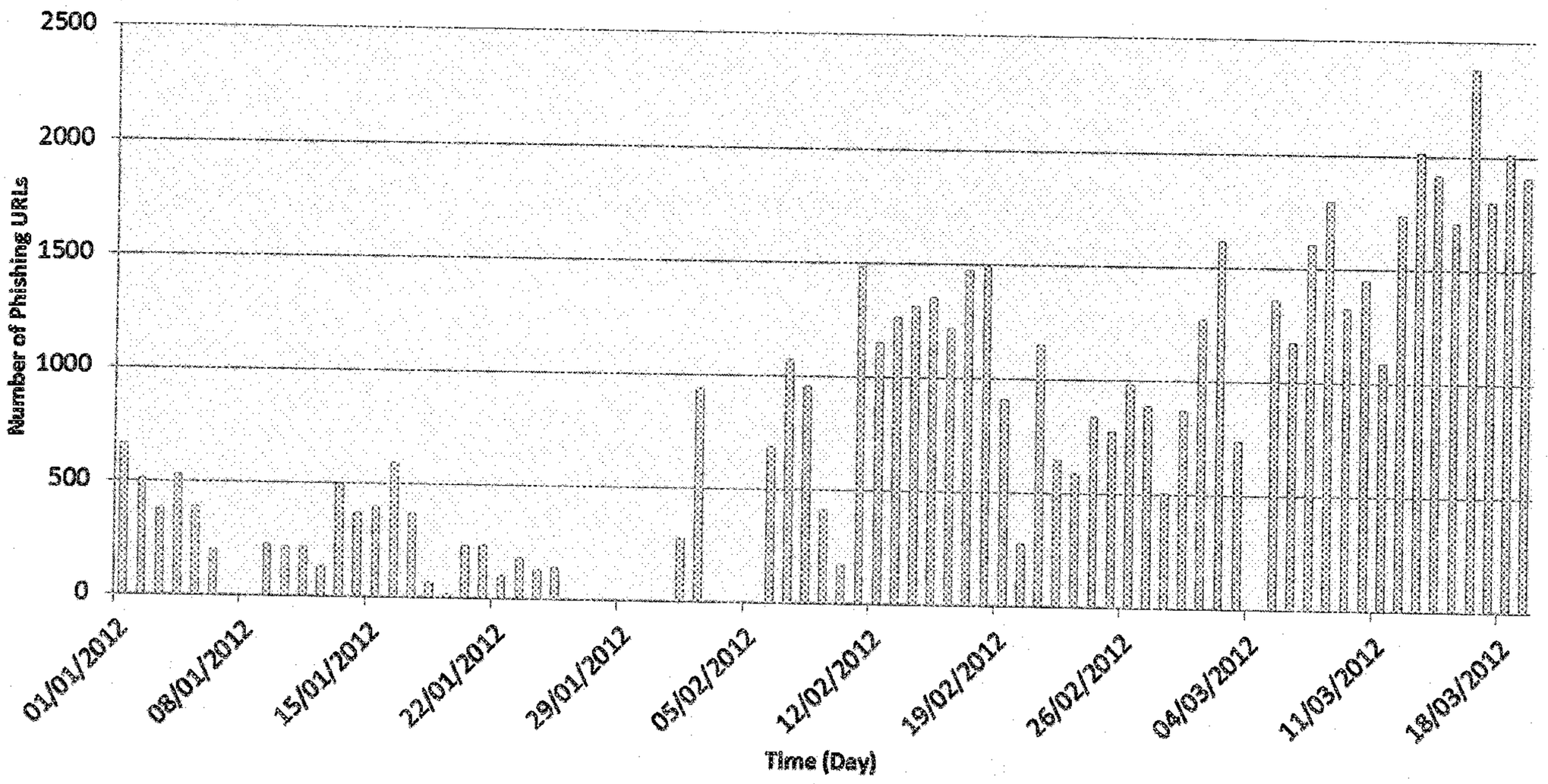


Figure 38: ThreatTrack™ Phishing URLs Feed in 2012

Pages 78 to / à 79

are withheld pursuant to section

sont retenues en vertu de l'article

16(2)(c)

of the Access to Information Act

de la Loi sur l'accès à l'information

In the Back-end Data Process, we are also in an early stage of having a Malware module, which can

7.2.3 Front-end Data Presentation

The Front-end of the system consists of a RDBMS (MySQL) and a web interface. It provides CFIs a platform where they can access actionable security insights including phishing, malicious domains/IP addresses/URLs, malware targeting them and others. This Front-end is in active development. Furthermore, a web service can be created to serve automatic request and can be integrated with our other platform.

7.3 Results

This section presents some screenshots of the web interface at the front-end. These screenshots depict implemented representations of our valuable security insights that we can offer to CFIs and ISPs. Figure 39 and Figure 40 are screenshots of the login page and the main page, which displays several statistics of the system.

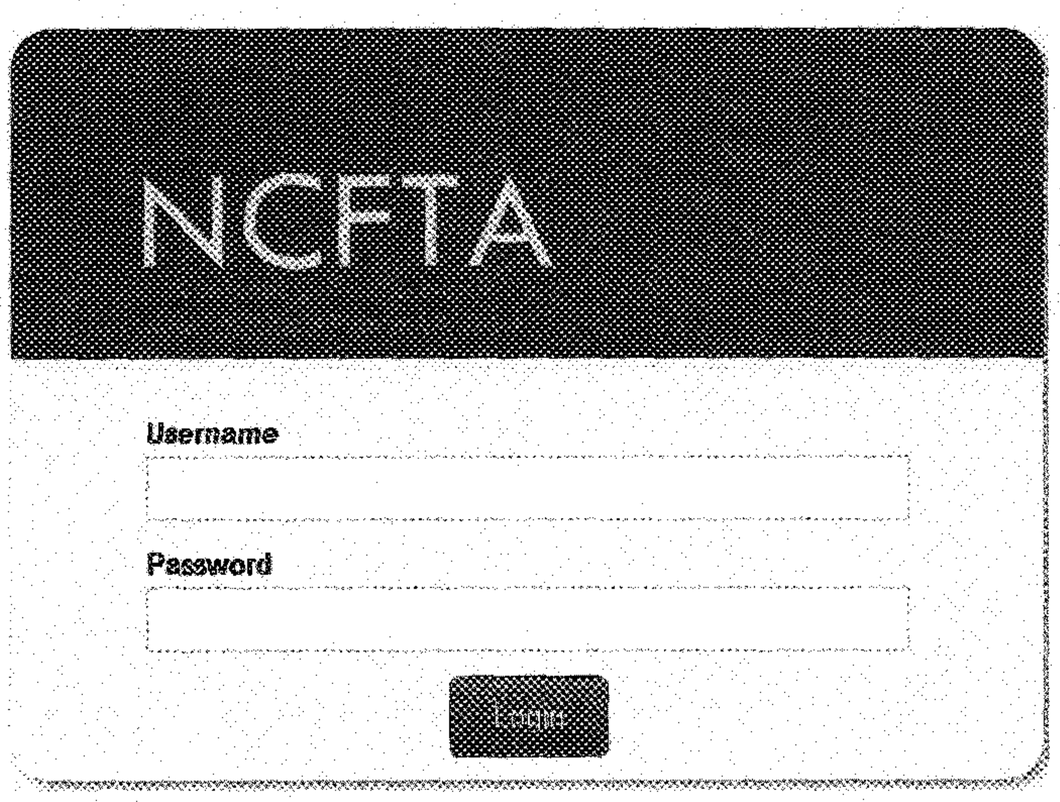


Figure 39: Login Page of the Portal

s.20(1)(c)

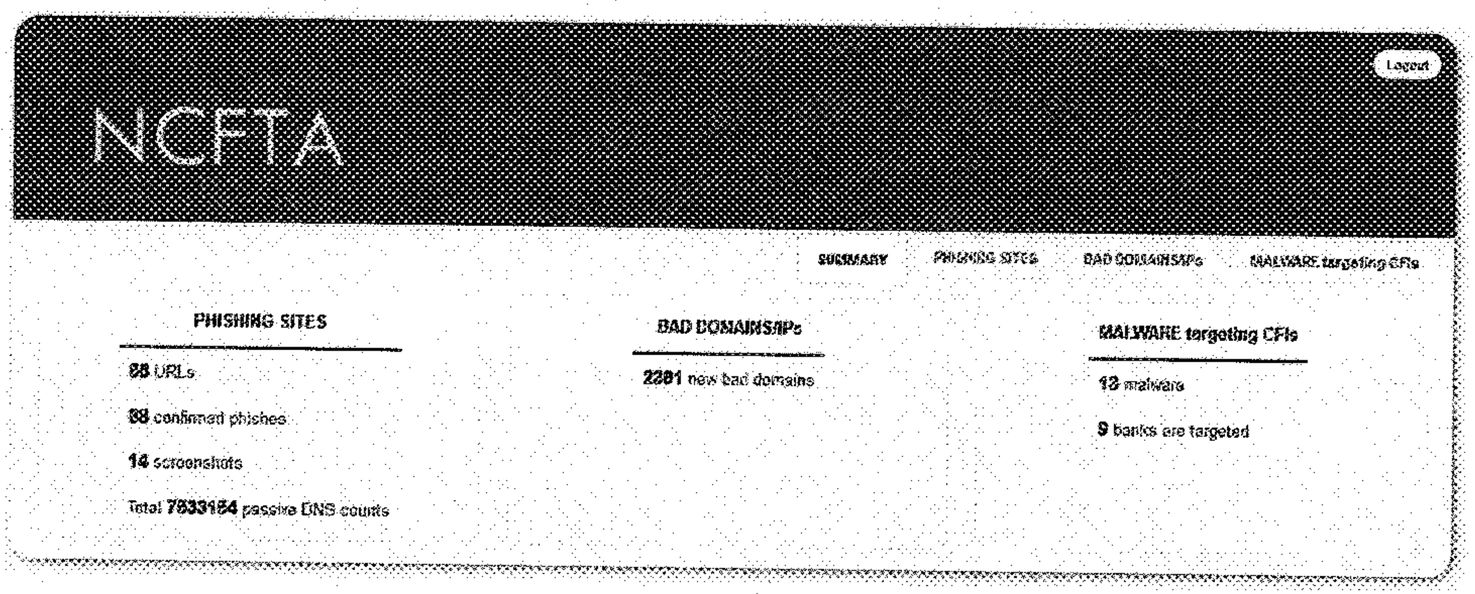


Figure 40: Main Page Displays Statistics of the System

During our testing stage, we managed to find 63 phishing URLs of

Although most of those phishing pages were injected into legitimate websites, some of them used their own bad domains with various deceiving sub-domains. An example of the phishing websites detected is shown in Figure 41.

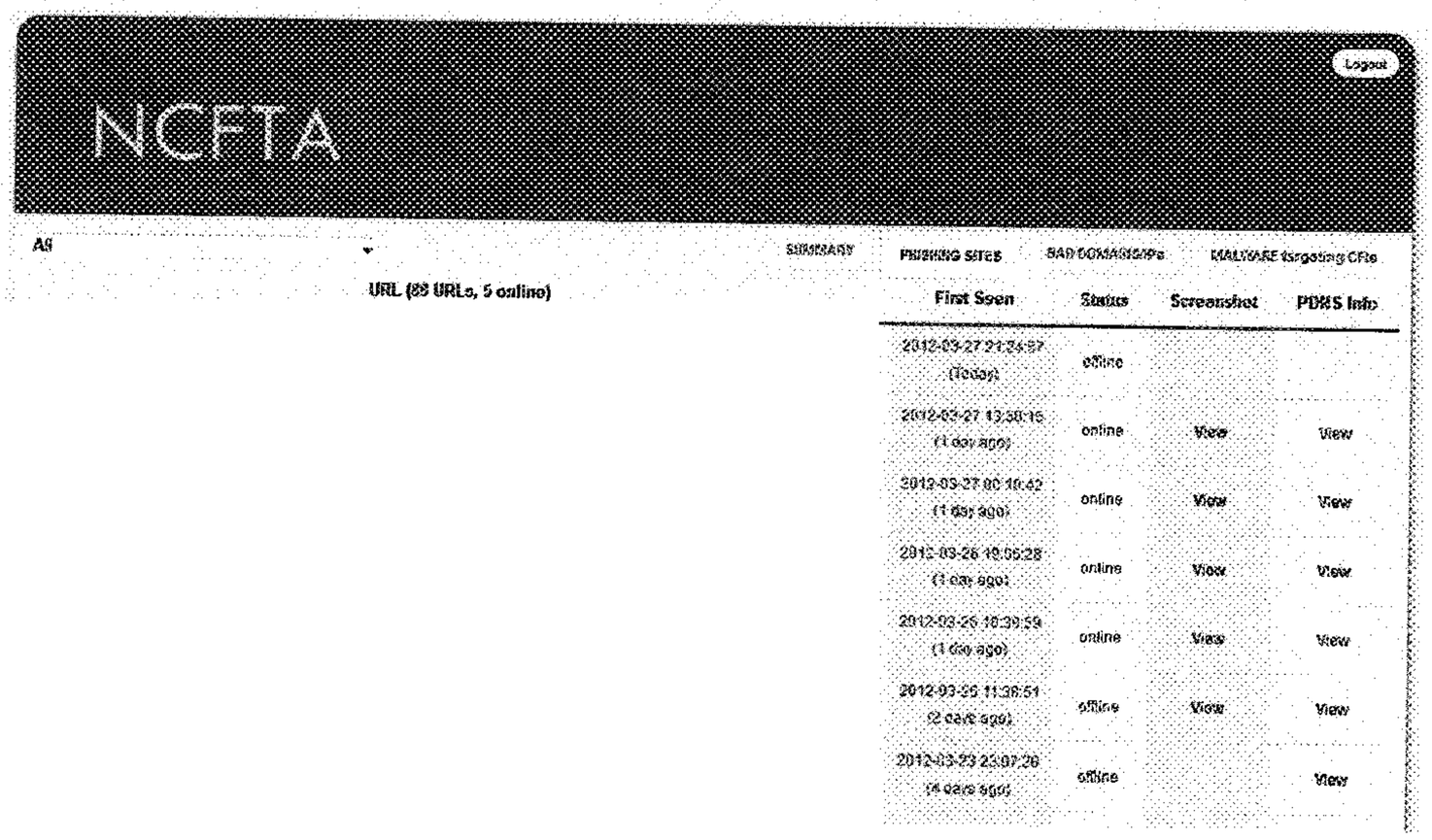


Figure 41: Phishing URLs

Moreover, Figure 42 reveals various sub-domains attacking

by using passive DNS data. We as well generate the number of DNS requests to those domains which can be used to estimate the damage of the phishing attack. Furthermore, we can also expose phishing-potential hostnames that have never been seen in spamtraps or phishing feeds. Those hostnames may be used for phishing attacks in the future.

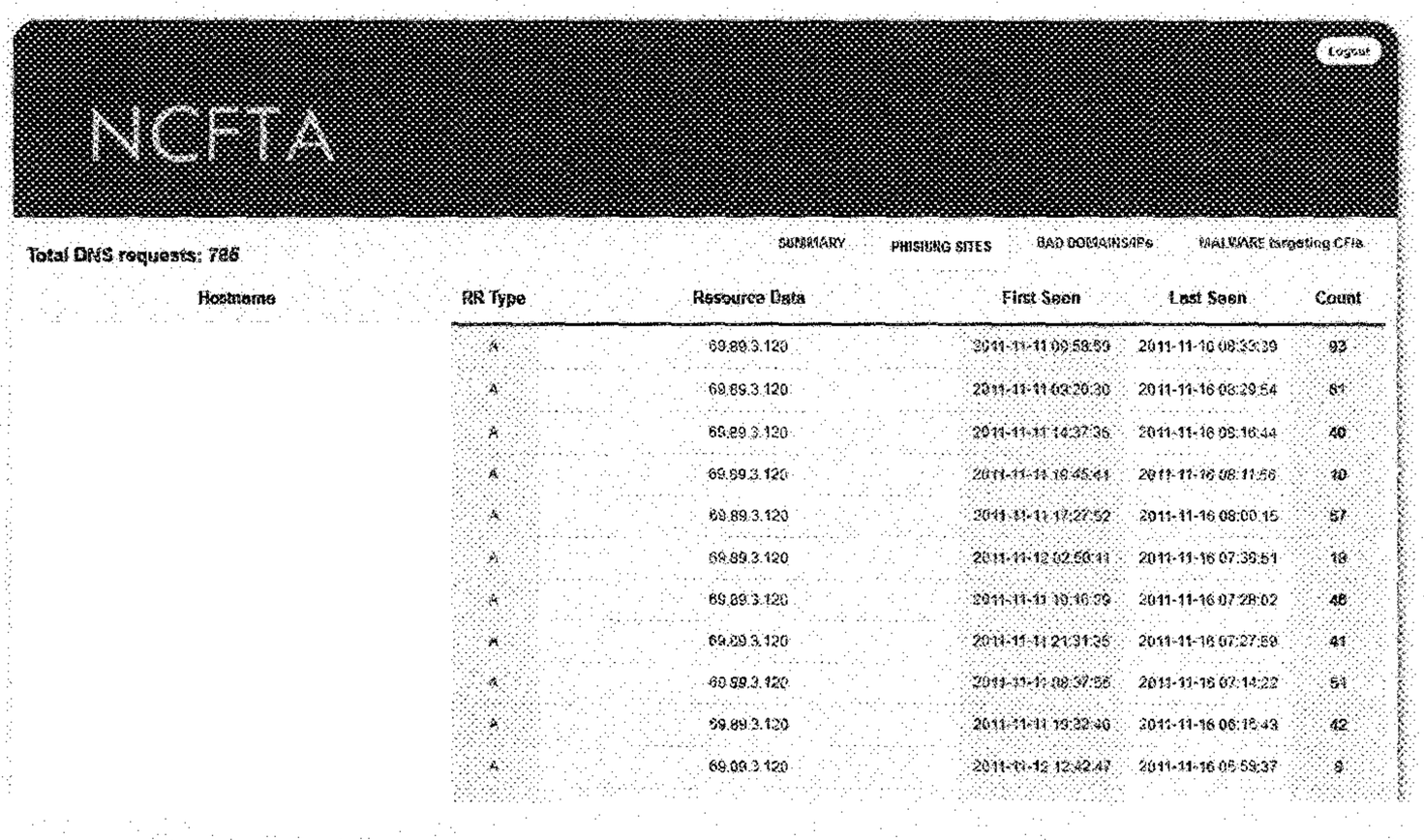


Figure 42: Passive DNS Information of Suspicious Phishing Domains

Figure 43 shows malware that are intentionally created to target CFIs. Most of them are classified as 'trojan bankers', which is a current major threat to FIs [77].

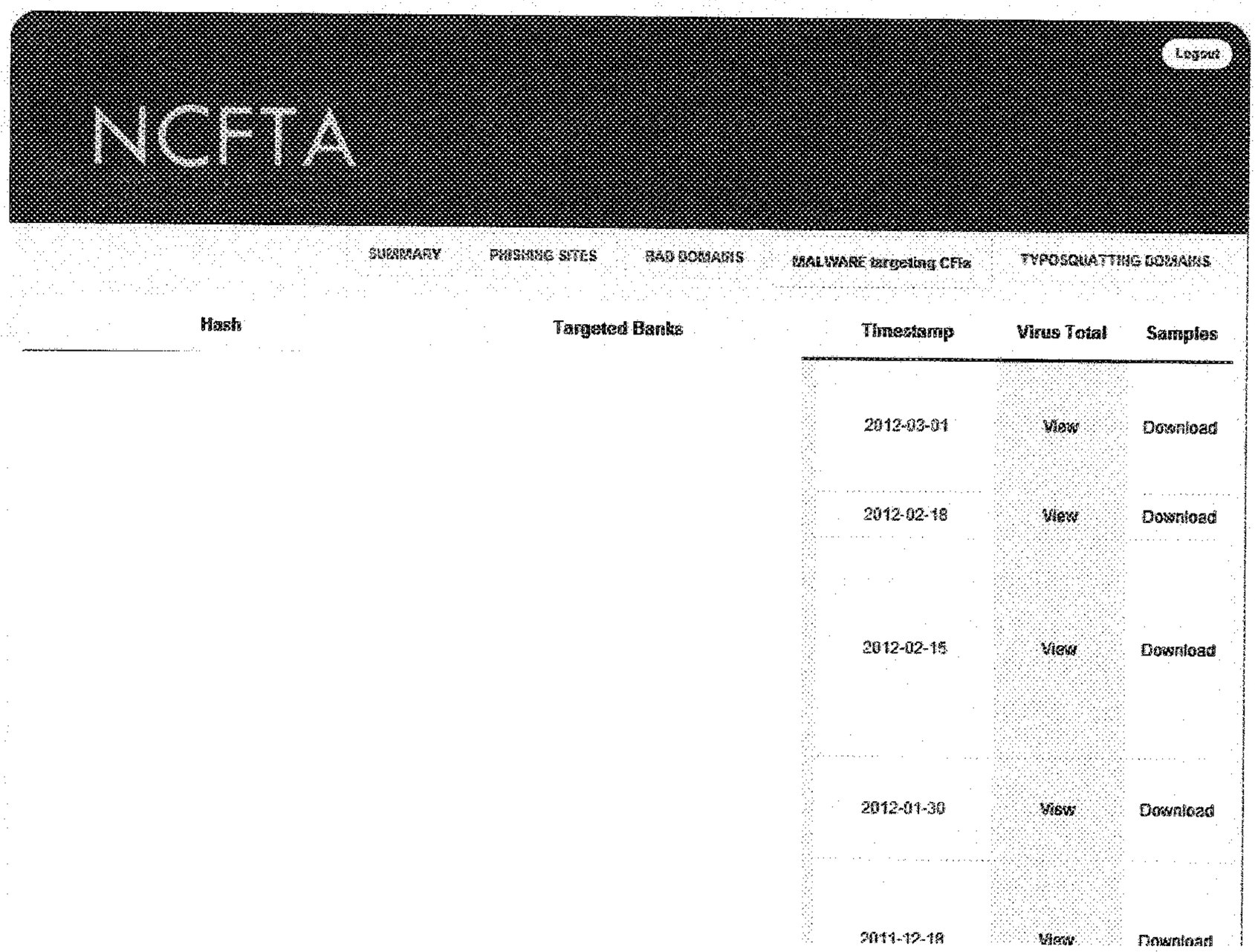


Figure 43: Screenshot of Malware Targeting CFIs

7.4 Conclusion

Nowadays, organizations, especially financial institutions, are greatly concerned about the security of the information system. They subscribe and receive many online security feeds to keep track of the security of their information, infrastructures, reputation and customers. Our brand protection system, even though still in an early stage, aims to provide financial institutions and Internet service providers valuable security data with minimized delay so that they can take action to protect their assets. In the future, newer and more advanced techniques such as online machine learning algorithms [86][87] can be utilized to improve the correctness and quality of our results.

8 Time Series Analysis

In this section, we perform time series analysis on the darknet data. The aim is to investigate and characterize the dynamics of self-similar behavior of the darknet streams. In other words, we intend to test whether such data (or a specific subpart of it) is correlated and hence can be predictable in the future. This insight is particularly of interest when there is a need to predict specific darknet threats, for the purpose of future mitigation.

This section is organized as follows: We initiate by defining the term time series coupled with its various types. Then we progress by pinpointing some of its objectives. Sequentially, we elaborate on our approach that utilizes the Detrended Fluctuation Analysis (DFA) to test for predictability and correlation within the darknet data. Finally, we discuss the implemented time series forecasting techniques and present the scenarios and their corresponding obtained results.

8.1 Definition and Types

A time series is a set of observations measured sequentially through time. These measurements can be taken continuously or at a discrete set of time points. By convention, these two types of series are called continuous and discrete time series, respectively, even though the measured variable may be of discrete or continuous nature in either case.

The usual method of analyzing a series is to sample (or digitize) it at equal intervals of time to provide a discrete-time series. Little or no information is lost by this process provided that the sampling interval is small enough. On the other hand, a discrete-time series may arise in three distinct ways, as follows:

1. by being sampled from a continuous series (e.g. temperature measured at hourly intervals. Such data may arise either by sampling a continuous trace, as noted above, or because measurements are only taken for example, once an hour);

2. by being aggregated over a period of time (e.g. total sales in successive months);

3. as an inherently discrete series (e.g. the dividend paid by a company in successive years).

For all the three types of discrete-time series, the data is typically recorded at equal intervals of time. Thus, the analysis of equally spaced discrete-time series constitutes the vast majority of time series applications. The analysis of the three types of discrete series is usually very similar, though one may occasionally wish to consider the effect of using different sampling intervals for continuous series or different periods of aggregation for aggregated data.

It is worth to note, that the data may be aggregated either across time or across the series. The first is called temporal aggregation and the second is called contemporaneous aggregation. For example, suppose we have sales figures for each of the various brand sizes of different products in successive weeks. Such data may be quite volatile and difficult to forecast without some form of aggregation, either across time (e.g. over successive 4-week periods) or across products (e.g. sum all brand sizes for

the same brand).

The possibility of aggregating data raises many questions such as how to choose the best level of aggregation for making forecasts and how to use monthly data to improve quarterly forecasts. For example, a common problem in inventory control is whether to develop a summary forecast for the aggregate of a particular group of items and then allocate this forecast to individual items based on their historical relative frequency, (the top-down approach), or perform individual forecasts for each item, (the bottom-up approach).

8.2 Time Series Objectives

A unique feature of time series data is that successive observations are usually dependent and hence the analysis must take into consideration the order in which the observations were collected. Effectively, each observation of the measured variable is a bivariate observation with time as the second variable.

The main objectives of time series analysis are:

- Description. To describe the data using summary statistics and/or graphical methods. A time plot of the data is particularly valuable.
- Modeling. To find a suitable statistical model to describe the data-generating process. A univariate model for a given variable is based only on past values of that variable, while a multivariate model for a given variable may be based, not only on past values of that variable, but as well on present and past values of other (predictor) variables. In the latter case, the variation in one series may aid to explain the variation in another series.
- Forecasting. To estimate the future values of the series. The literature uses the terms 'forecasting' and 'prediction' interchangeably and we follow this convention. There is a clear distinction between steady-state forecasting, where we expect the future to be much like the past, and what-if forecasting where a multivariate model is used to explore the effect of changing policy variables.
- Control. Strong reliable forecasts enable the analyst to take action for the purpose of controlling a given process.

8.3 Approach

Before we can apply time series analysis techniques for the purpose of forecasting, there is a need first to test for predictability. The aim is to pinpoint which darknet traffic time series is correlated; its future predicted values depend on current (or past) values. To perform such testing, we adopt the detrended fluctuation analysis method, which will be discussed next.

8.3.1 Detrended Fluctuation Analysis

The DFA method of characterizing a non-stationary time series is based on the root mean square analysis of a random walk. DFA is advantageous in comparison with other methods such as spectral

analysis and Hurst analysis since it permits the detection of long range correlations embedded in a seemingly non-stationary time series. It avoids as well the spurious detection of apparent long-range correlations that are an artifact of non-stationarity. Another advantage of DFA is that it produces results that are independent of the effect of the trend.

To utilize DFA, the following steps need to be applied:

- Integrate the time series; The time series of length N is integrated by applying $y(k) = \sum_{i=1}^{k}[B(i) - B_{ave}]$, where $B(i)$ is the ith interval and B_{ave} is the average interval.
- Divide the time series into 'boxes' of length n.
- In each box, perform a least-squares polynomial fit of order p. The y coordinate of the straight line segments is denote by $y_n(k)$.
- In each box, detrend the integrated time series, $y(k)$, by subtracting the local trend, $y_n(k)$. The root-mean-square fluctuation of this integrated and detrended time series is calculated by

$$F(n) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}[y(k) - y_n(k)]^2}. \tag{1}$$

- Repeat this procedure for different box sizes (i.e, time scales) n.

The output of the above procedure is a relationship $F(n)$, the average fluctuation as a function of box size, and the box size n. Typically, $F(n)$ will increase with box size n. A linear relationship on a log-log graph indicates the presence of scaling; statistical self-affinity expressed as $F(n) \sim n^{\alpha}$. Under such conditions, the fluctuations can be characterized by a scaling exponent α, which is the slope of the line relating $logF(n)$ to $log(n)$.

The scaling exponent α can take the following values, disclosing the correlation status of the time series.

- $\alpha \prec 0.5$: anti-correlated.
- $\alpha \approx 0.5$: uncorrelated or white noise.
- $\alpha \succ 0.5$: correlated.
- $\alpha \approx 1$: $1/f - noise$ or pink noise.
- $\alpha \succ 1$: non-stationary, random walk like, unbounded
- $\alpha \approx 1.5$: Brownian noise.

8.3.2 DFA Results

To implement DFA, we have utilized the MATLAB code found in [88]. Specifically, we have used DFA to test the predictability/correlation of 1) various darknet traffic (i.e., TCP, UPD, ICMP), 2) various darknet traffic destined to specific destinations and 3) specific darknet threats targeting specific destinations. The first point presents a high-level view of the dynamics of the darknet traffic. The second point demonstrates a closer look at such traffic to better understand its statistical properties. The third point provides specific insights about the self-similarity and correlation of some of the darknet threats. The outcome of those procedures will 1) reveal the temporal fluctuation of the darknet data and 2) inform us whether or not specific darknet threats are correlated and hence whether or not those could be further analyzed using time series techniques for the purpose of forecasting/prediction.

Scenario I: DFA applied to Global Darknet Traffic: The aim of this scenario is to characterize the dynamics of the darknet traffic. It presents a high-level view of the self-similarity of the darknet streams. To accomplish the mentioned objective, we have applied DFA on the three major darknet traffic sources, namely, TCP, UDP and ICMP traffic using four days of darknet streams. The distribution of those darknet traffic sources in a one day sample is presented in Figure 44. Consequently, the results of the DFA of those sources for the four days is depicted in Figure 45. The results disclose that, on a global darknet scale, ICMP time series is correlated, where the scaling exponent ranges from 0.56 to 0.74 with an average of 0.66. Moreover, the correlation status of UDP time series fluctuates per day between being correlated and being $1/f - noise$. Such results trigger the need for further, more specific investigation of this time series, which we perform in Scenario II. On the other hand, TCP, on a global darknet scale, seems to be a random walk where in few cases, it tends as well to get closer to Brownian noise.

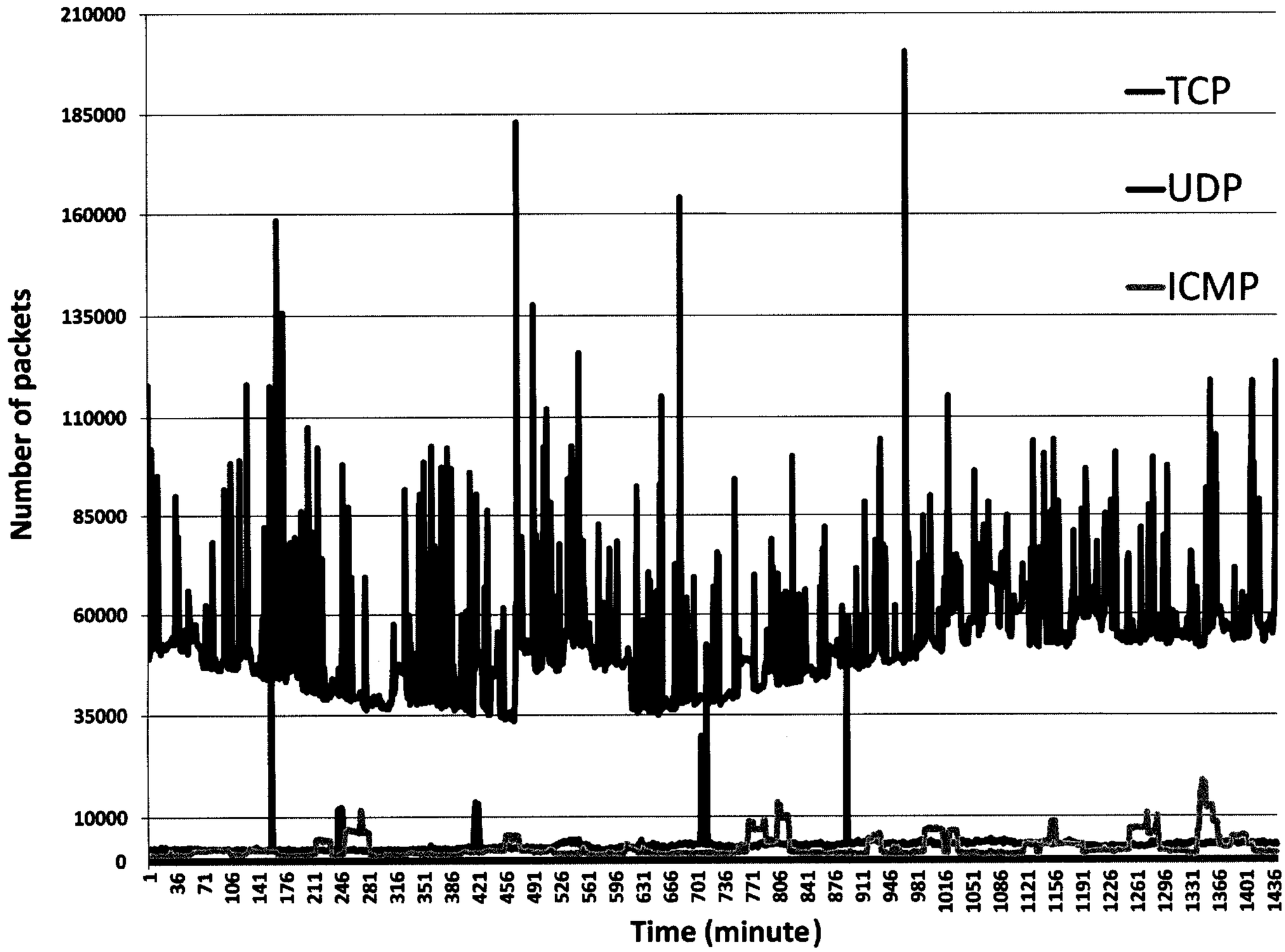


Figure 44: TCP, UDP and ICMP Distributions

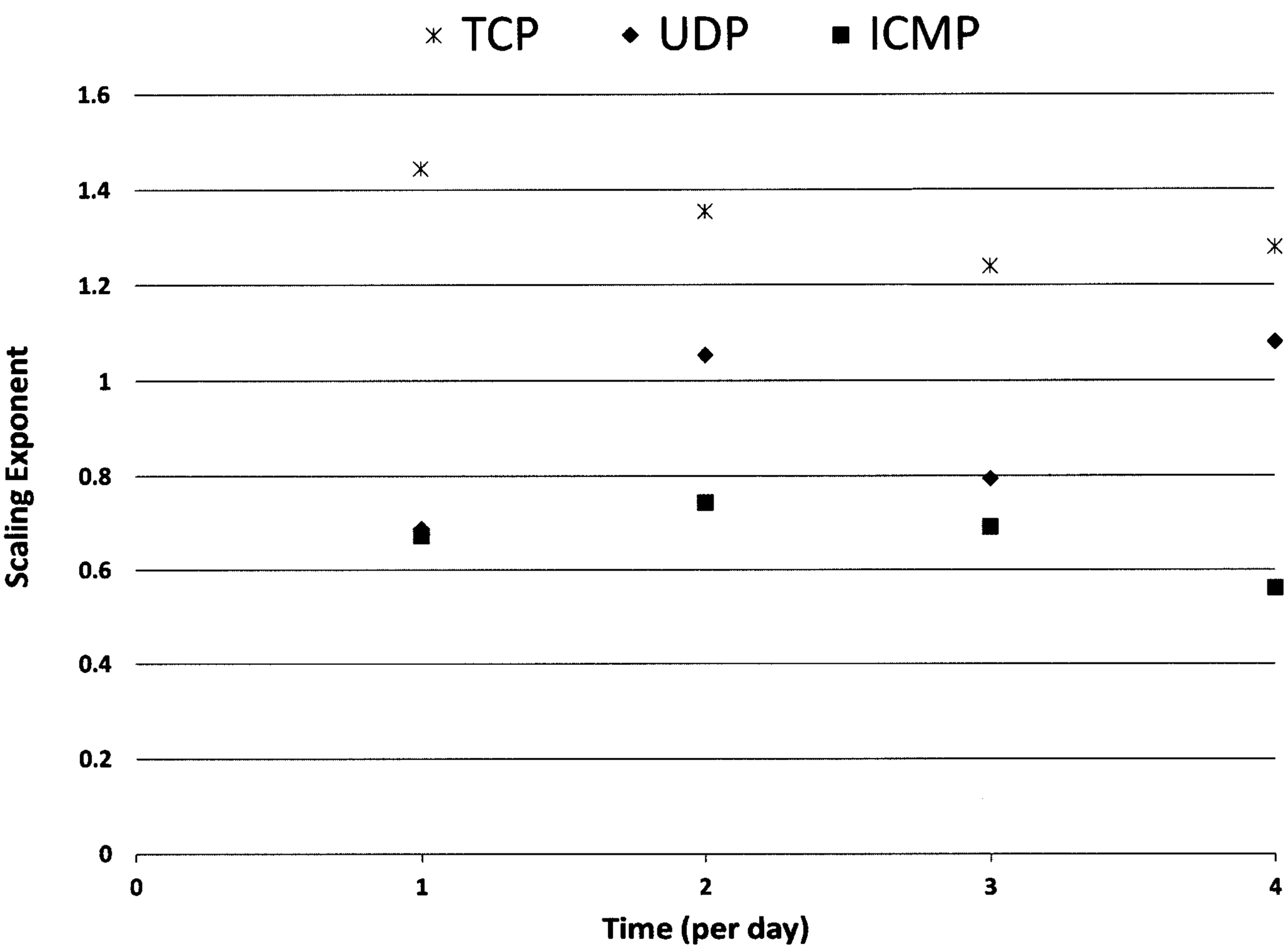


Figure 45: DFA Exponents: TCP, UDP and ICMP

Scenario II: DFA applied to Darknet Traffic destined to Specific Destinations: The aim of this scenario is to provide a better understanding of the dynamics of the darknet traffic. We accomplish this by selecting five darknet providers as our target destinations. Moreover, in addition to performing DFA analysis on TCP, UDP and ICMP traffic as we did in Scenario I, we extend the analyses to scanning and backscattering traffic destined to those providers. A daily sample of those distributions is depicted in Figure 46.

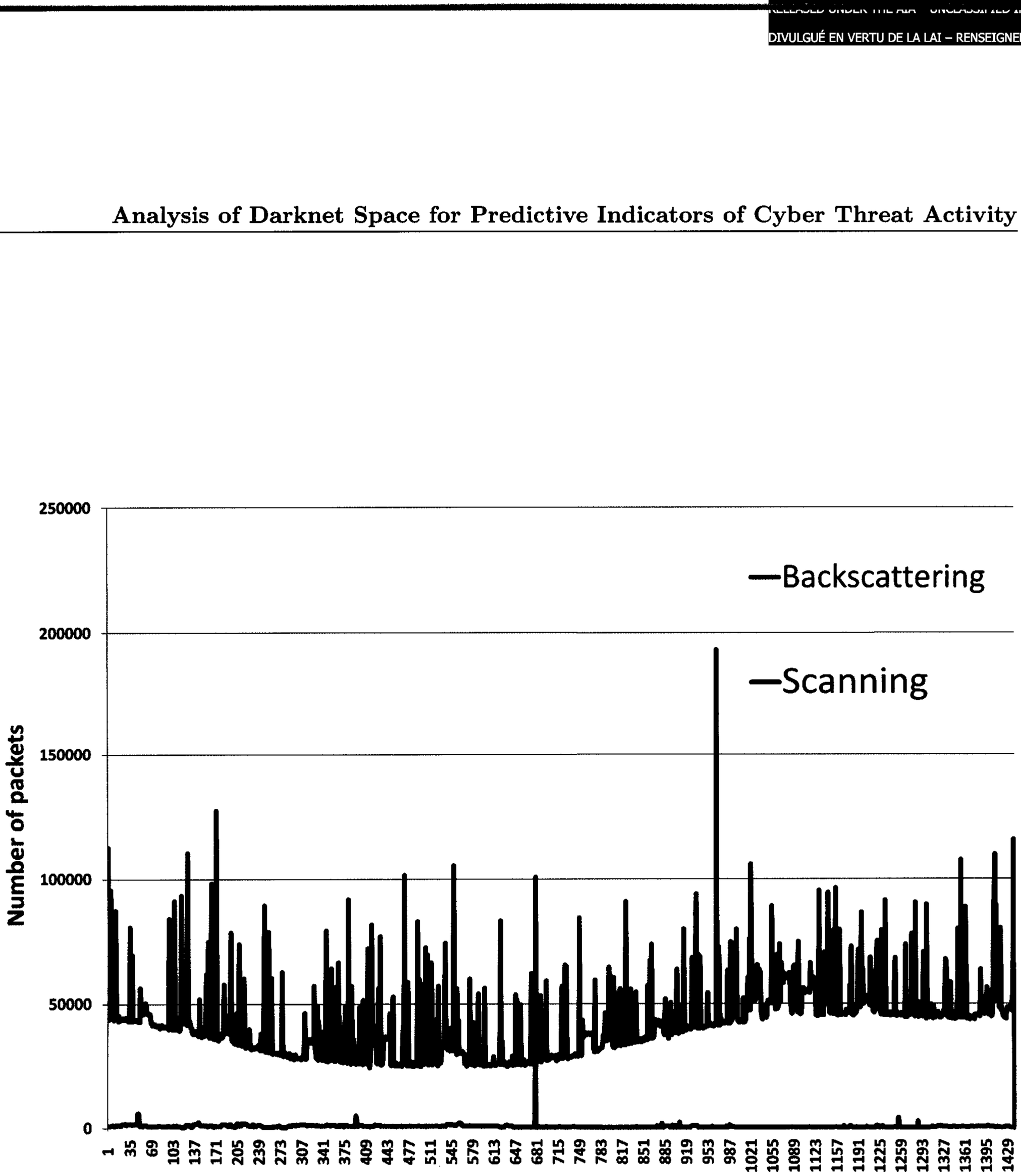


Figure 46: Scanning and Backscattering Distributions

s.20(1)(c)

Moreover, Figures 47, 48, and 49 show the DFA results of TCP, UDP and ICMP respectively.

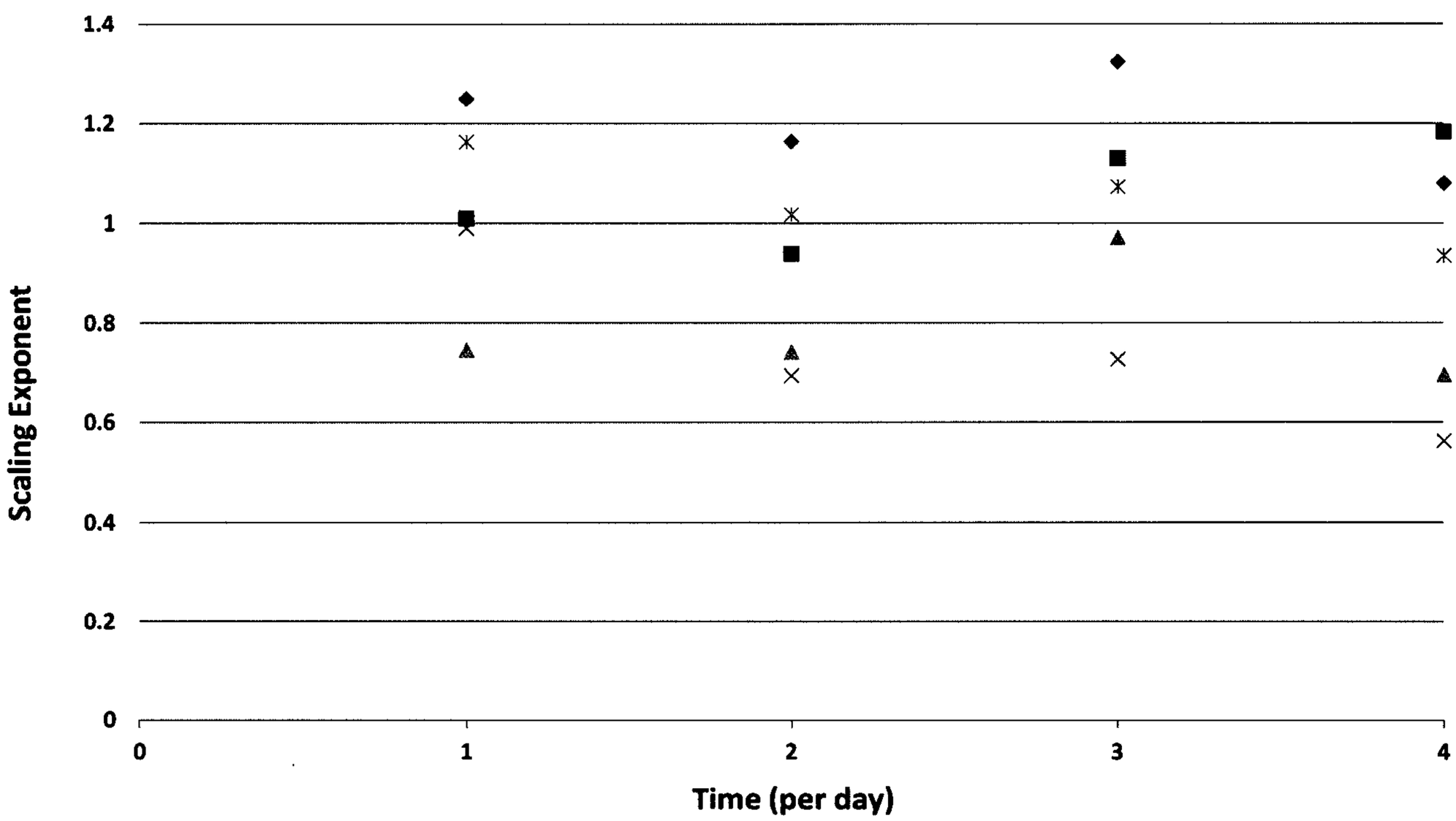


Figure 47: DFA Exponents: TCP

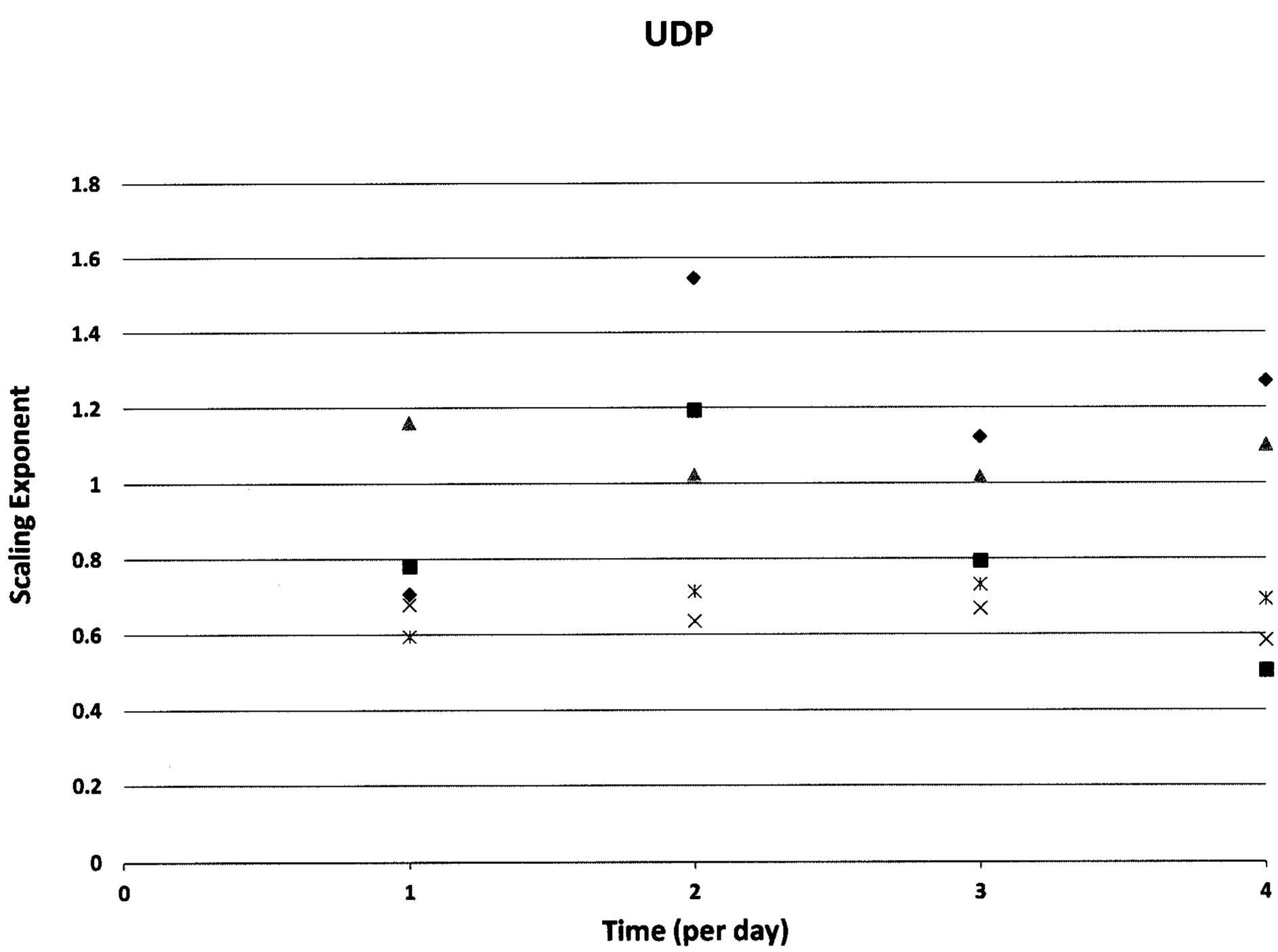


Figure 48: DFA Exponents: UDP

s.20(1)(c)

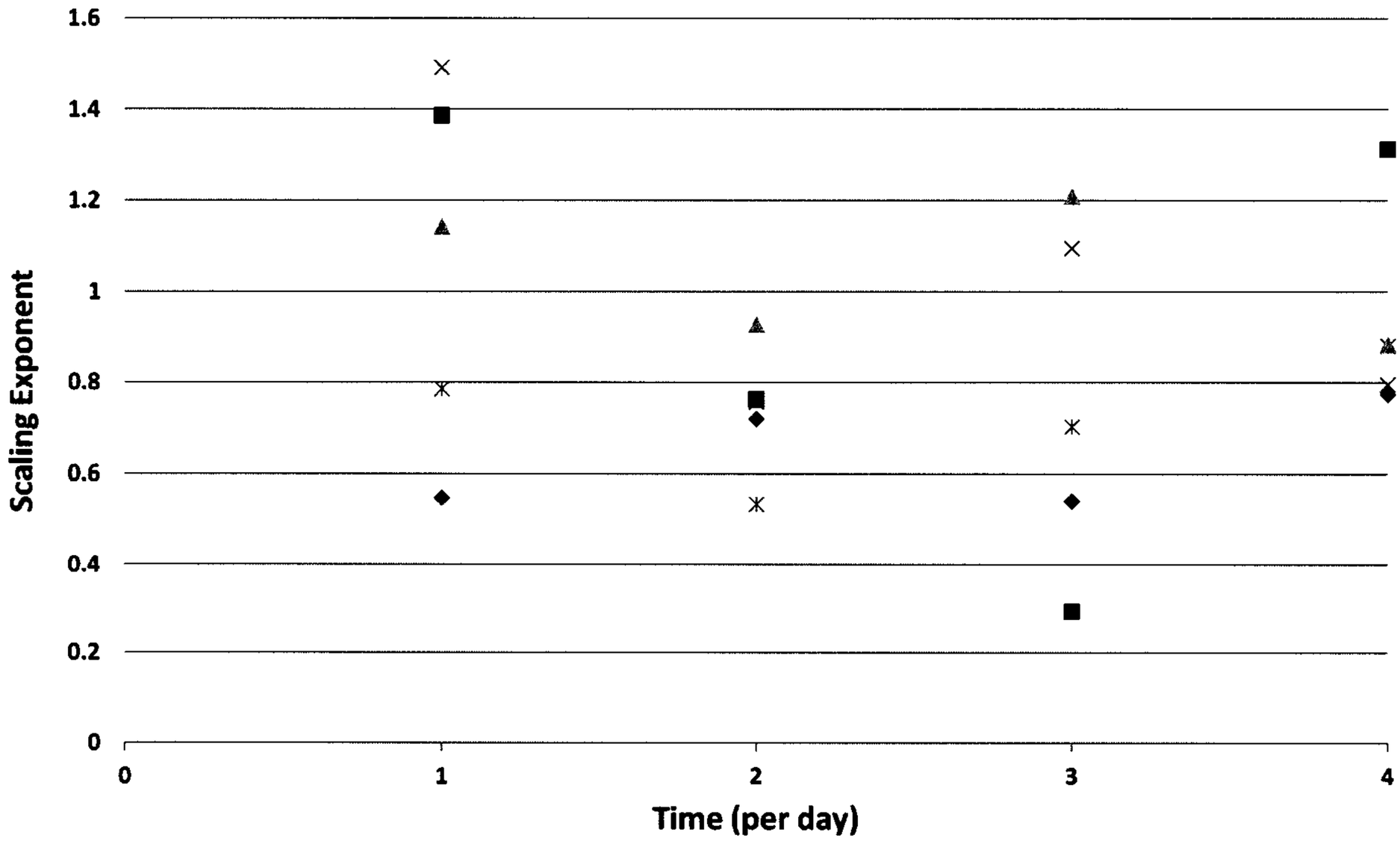


Figure 49: DFA Exponents: ICMP

Furthermore, the results of the DFA applied to scanning and backscattering traffics for four darknet days destined to the five providers are illustrated in Figures 50 and 51 respectively.

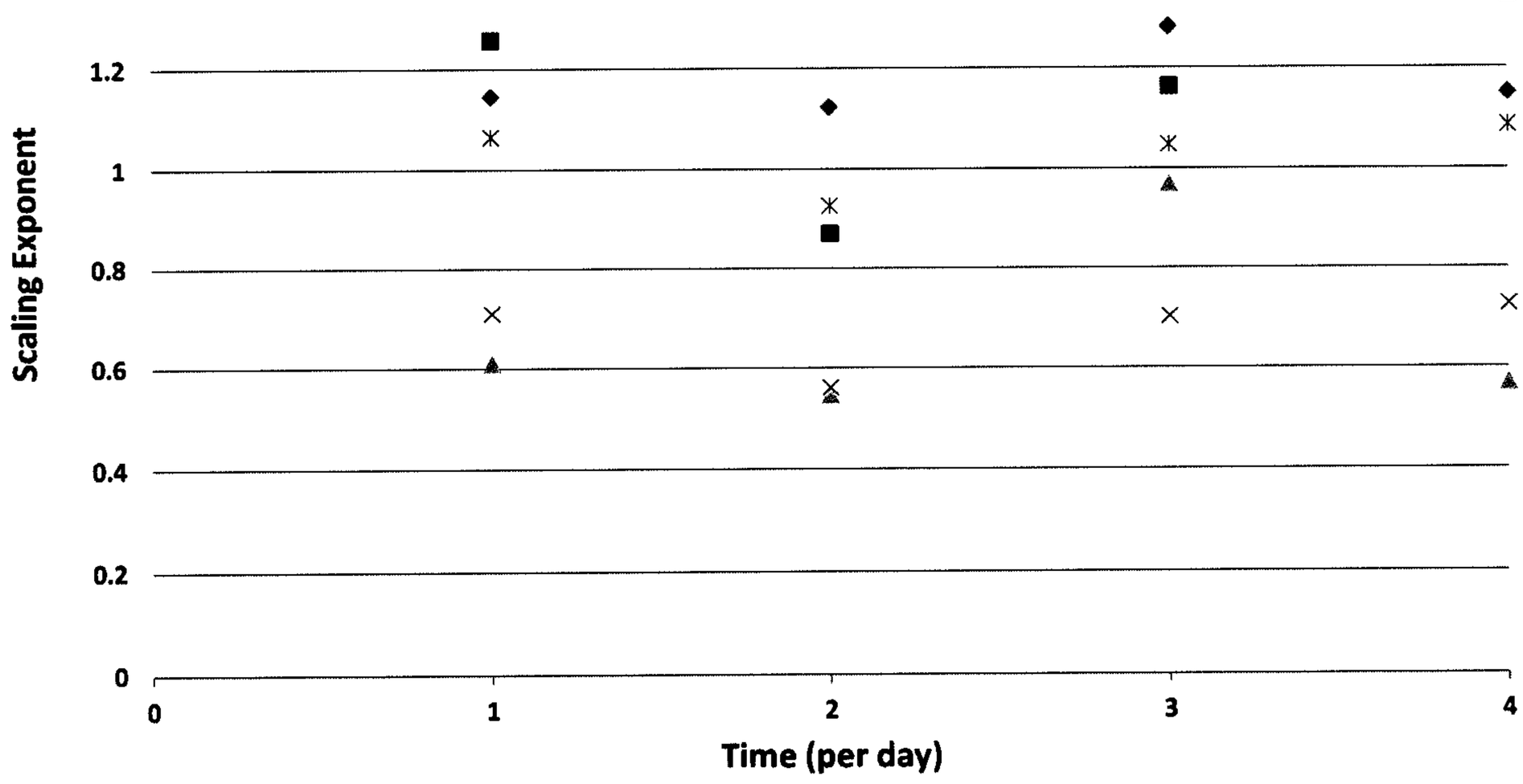


Figure 50: DFA Exponents: Scanning

s.20(1)(c)

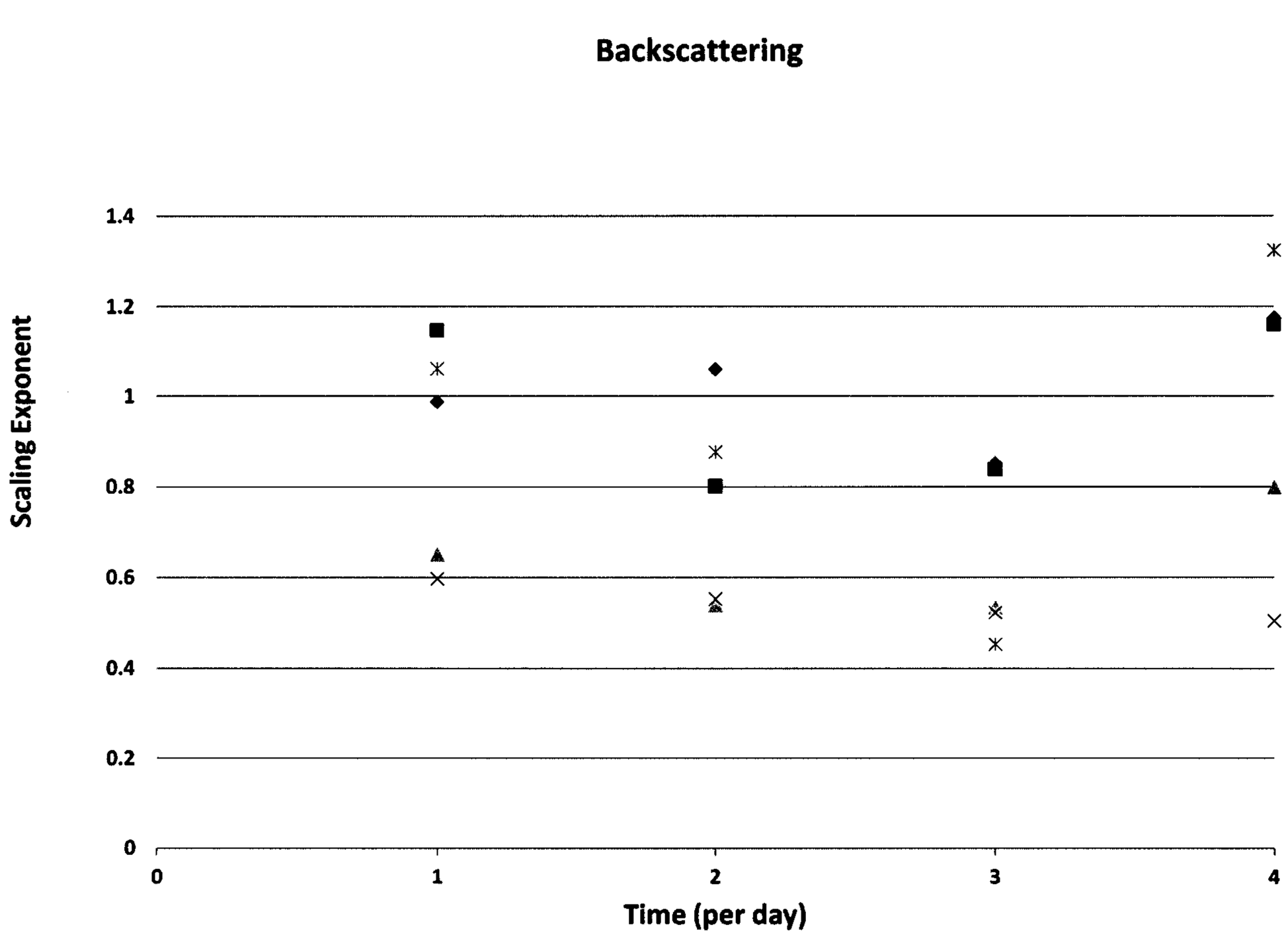


Figure 51: DFA Exponents: Backscattering

The results can be summarized as follows:

- TCP traffic destined to and Performance Systems International is correlated.
- TCP traffic destined to other providers is either $1/f - noise$, Brownian noise or a random walk.
- UDP traffic destined to is correlated.
- UDP traffic destined to is highly unstable.
- ICMP traffic destined to is correlated.
- ICMP traffic destined to is highly unstable.
- Backscattered traffic destined to is correlated.
- Backscattered traffic destined to is close to $1/f - noise$.
- Scanning traffic destined to is correlated.
- Scanning traffic destined to is close to $1/f - noise$.

The correlation of TCP, UDP and ICMP traffic destined to specific destinations implies that such traffic can be modeled using the unique value of the scaling exponent α and hence their self-similarity or dynamics can be rendered in such way. On the other hand, the correlation of backscattered and scanning traffic destined to specific destinations implies that such data is predictable and hence those could be further analyzed using time series techniques for the purpose of mitigation.

Scenario III: DFA applied to Specific Darknet Threats: The aim of this scenario is to analyze the correlation of specific darknet threats targeting specific destinations. To achieve this task, we selected a traceroute threat (medium priority) and a buffer overflow (high priority) threat targeting To test for predictability of those threats, we applied DFA on those threats time series for four days of darknet data streams. The results are represented in Table 8. Note that, the threat notation is consistent to that mentioned in Section 5.3 of this report. The results demonstrate that the time series of those threats are correlated in which the scaling exponent of the DFA fluctuated from 0.59 to 0.84. Such results imply that the threats could be further analyzed using time series techniques for the purpose of forecasting and hence mitigation.

		Scaling Exponent			
Service Provider	Threat	Day 1	Day 2	Day 3	Day 4
	t_1	0.7584	0.7134	0.874	0.7634
	t_4	0.7806	0.7948	0.5939	0.6853

Table 8: DFA Exponents on Specific Threats

8.4 Forecasting Techniques

Although there exits various time series techniques and algorithms, in this work, we adopt three methods, namely, the exponential smoothing, the weighted moving average, and the moving average. Subsequently, we discuss the adopted three time series techniques.

8.4.1 Exponential Smoothing

Exponential smoothing has become very popular as a forecasting method for a wide variety of time series data. A simple and pragmatic model for a time series would be to consider each observation as consisting of a constant and an error component, that is: The constant (b) is relatively stable in each segment of the series but may change slowly over time. If appropriate, then one way to isolate the true value of (b) and thus the systematic or predictable part of the series is to compute a moving average, where the current and immediately preceding ('younger') observations are assigned greater weight than the respective older observations. Simple exponential smoothing accomplishes exactly such weighting, where exponentially smaller weights are assigned to older observations. The specific formula for simple exponential smoothing is:

$$St = a * X_t + (1 - a) * S_{t-1}$$

When applied recursively to each successive observation in the series, each new smoothed value (forecast) is computed as the weighted average of the current observation and the previous smoothed observation; the previous smoothed observation was computed in turn from the previous observed value and the smoothed value before the previous observation, and so on. Thus, in effect, each smoothed value is the weighted average of the previous observations, where the weights decrease exponentially depending on the value of parameter (a). If (a) is equal to 1 then the previous observations are ignored entirely; if it is equal to 0, then the current observation is ignored entirely, and the smoothed value consists entirely of the previous smoothed value.

8.4.2 Moving Average

Moving average techniques forecast demand by calculating an average of actual demands from a specified number of prior periods. Hence, each new forecast drops the demand in the oldest period and replaces it with the demand in the most recent period. Thus, the data in the calculation "moves" over time. Simple moving average is expressed by:

$$A_t = ((D_t + D_{t-1} + D_{t-2} + ... + D_{t-N+1})/N)$$

Where:
N = total number of periods in the average;
Forecast for period t+1: $F_{t+1} = A_t$;

A key decision in the above is N; how many periods should be considered in the forecast. The trade-off resides in the following reasoning. On one hand, if we increase the value of N, we will obtain greater smoothing but lower responsiveness. On the other hand, if we decrease the value of N, we will obtain less smoothing but more responsiveness. Hence, the more periods (N) over which the moving average

is calculated, the less susceptible the forecast is to random variations, but the less responsive it is to changes. By default, a large value of N is appropriate if the underlying pattern of demand is stable whereas a smaller value of N is appropriate if the underlying pattern is changing or if it is important to identify short-term fluctuations.

8.4.3 Weighted Moving Average

A weighted moving average is a moving average where each historical demand may be weighted differently. The average can be expressed as:

$$A_t = W_1 * D_t + W_2 * D_{t-1} + W_3 * D_{t-2} + ... + W_N * D_{t-N+1}$$

Where:
N = total number of periods in the average;
W_t= weight applied to period t's demand;
$\Sigma w = 1$;
Forecast: $F_{t+1} = A_t$ which is the forecast for period T_{t+1}.

8.5 Experimental Results

In this section, we apply the forecasting techniques on the darknet traffic and threats that *were shown to be predictable using DFA*. Specifically, we forecast backscattering and scanning traffic on the medium priority (traceroute) threat t_4 and high priority (buffer overflow) threat t_1 targeting

Figure 52 demonstrates the application of the forecasting techniques on the backscattering traffic from The one day count distribution of such traffic is plotted in black. Using the forecasting techniques, we are able to predict that traffic for the next (future) 90 minutes. It is observable that at the end of 90 minutes, the techniques predicted an average of approximately 60,000 packets.

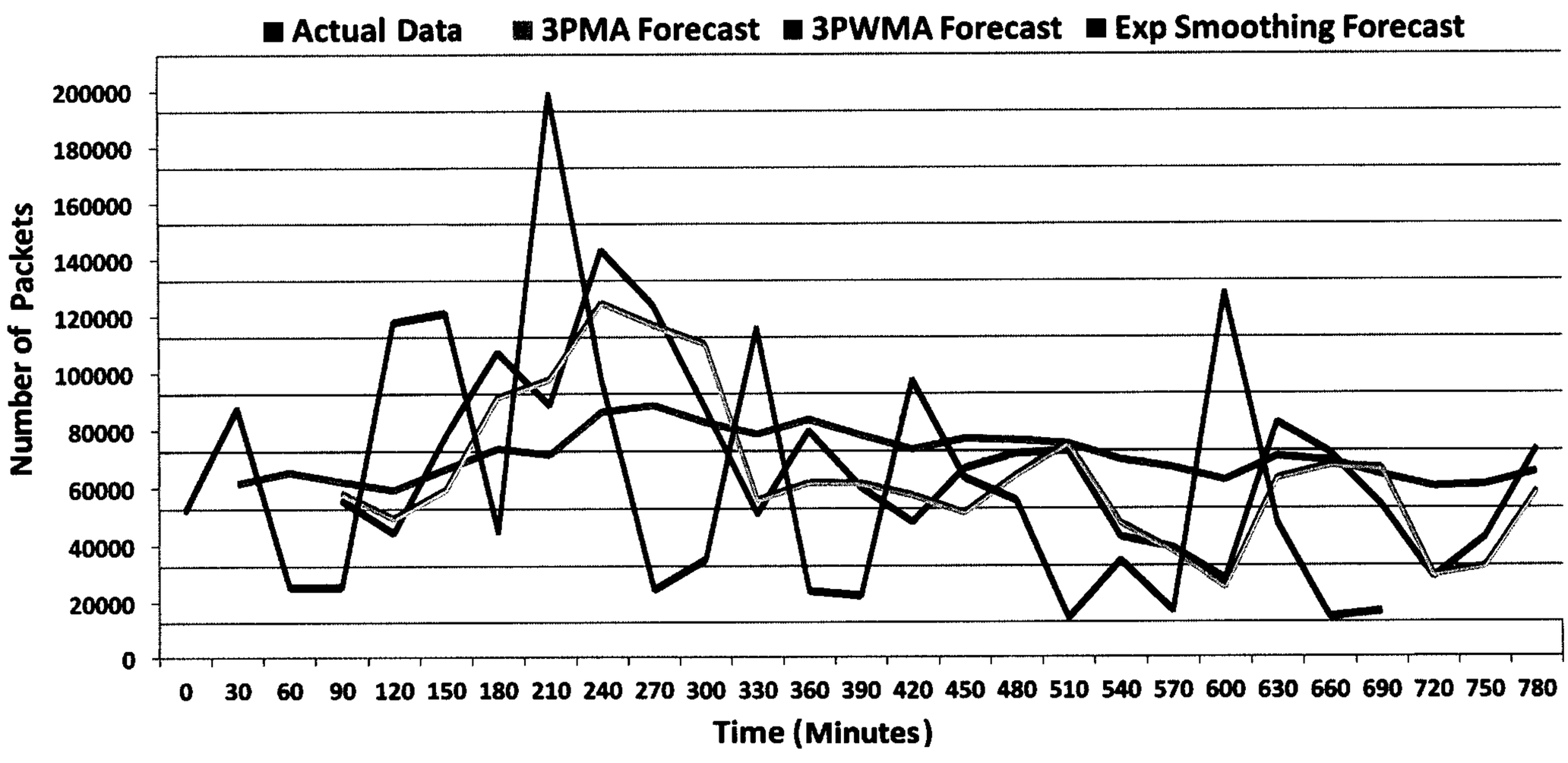


Figure 52: Forecasting of Backscattering Traffic

Figure 53 demonstrates the application of the forecasting techniques on the scanning traffic on The one day count distribution of such traffic is plotted in black. Using the forecasting techniques, we are able to predict that traffic for the next (future) 120 minutes. It is observable that at the end of 120 minutes, the exponential smoothing forecasting technique predicted an average of approximately 100,000 scanning packets, the three point moving average technique forecasted approximately 170,000 scanning packets and the three point weighted moving average technique recorded less than 50,000 scanning packets targeting

Moreover, Figure 54 depicts the distribution of the medium severity (traceroute) threat t_4. It is observable that such threat distribution is extremely periodic and systemic. Hence, its forecasting and hence mitigation is trivial and thus there is exist no need to apply the forecasting approaches on such distribution. Note that in this case, DFA results of this time series that were presented in section 8.3.2

s.20(1)(c)

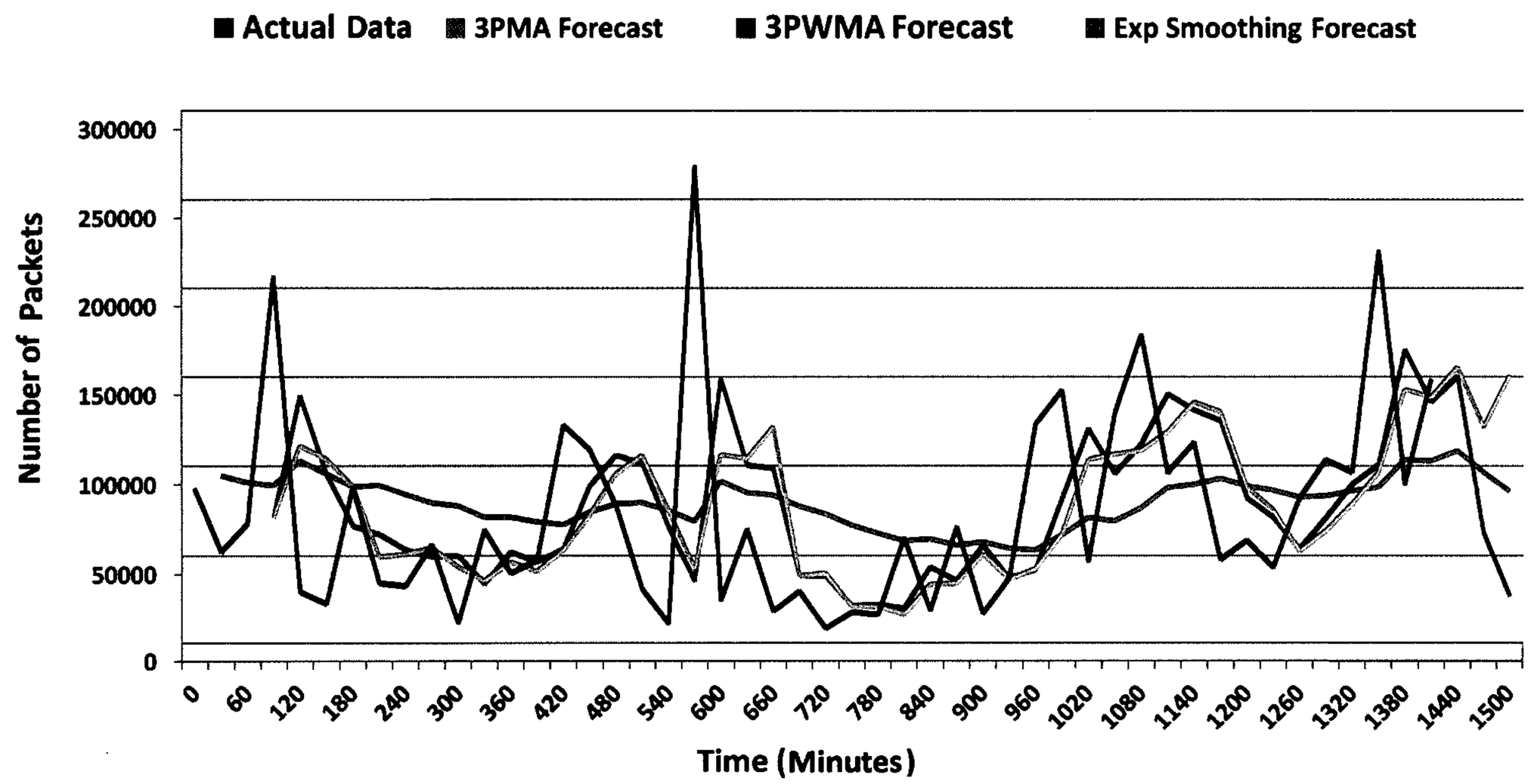


Figure 53: Forecasting of Scanning Traffic-

were very significant and led us to discover such periodic distribution.

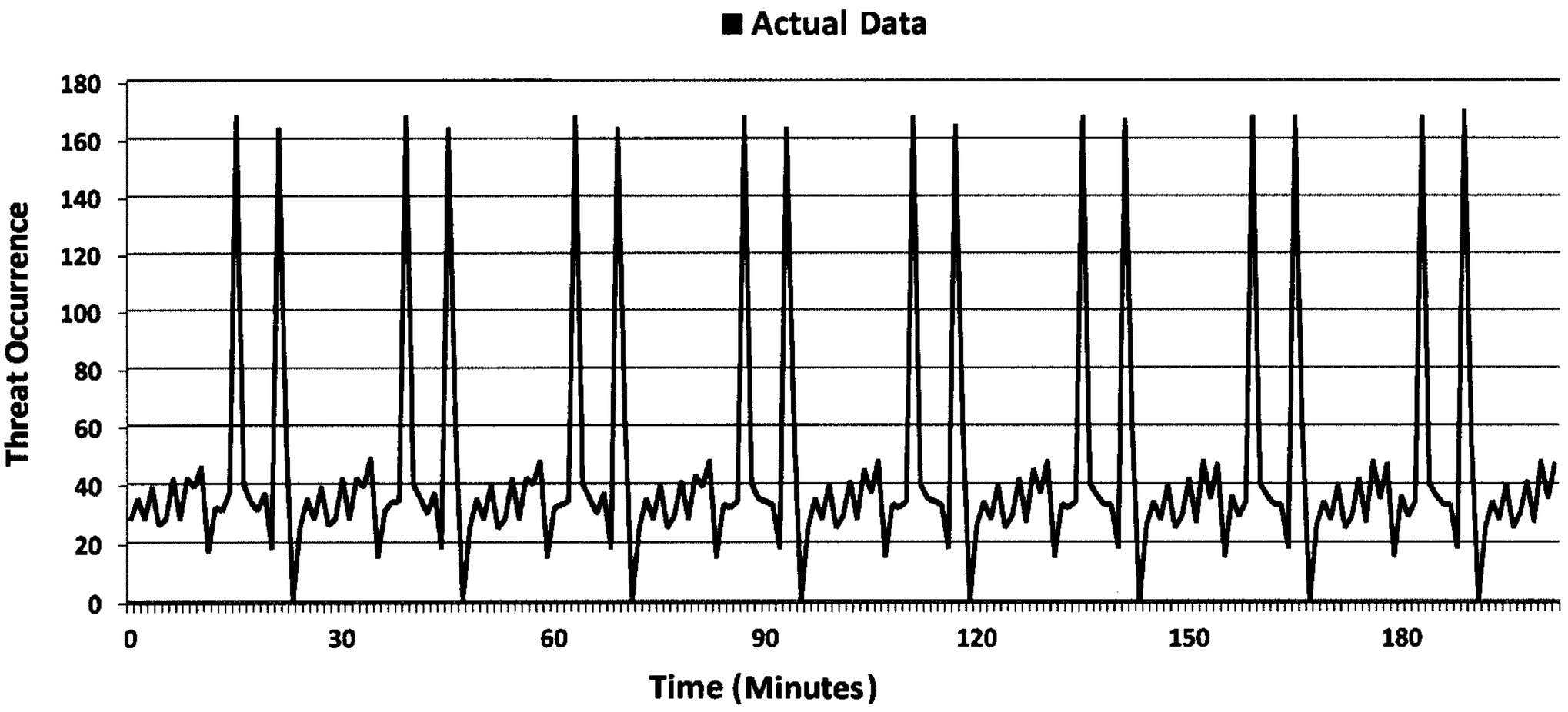


Figure 54: Forecasting of the Medium Severity (traceroute) Threat

Finally, Figure 55 demonstrates the application of the forecasting techniques on the high severity

(buffer overflow) threat t_1. The one day count distribution of such traffic is plotted in black. Using the forecasting techniques, we are able to predict the occurrences for that threat for the next (future) 180 minutes. It is observable that at the end of 180 minutes, the exponential smoothing forecasting technique predicted an average of approximately 3 threat occurrences, the three point moving average technique forecasted approximately 7 attacks and the three point weighted moving average technique predicted 2 occurrences targeting

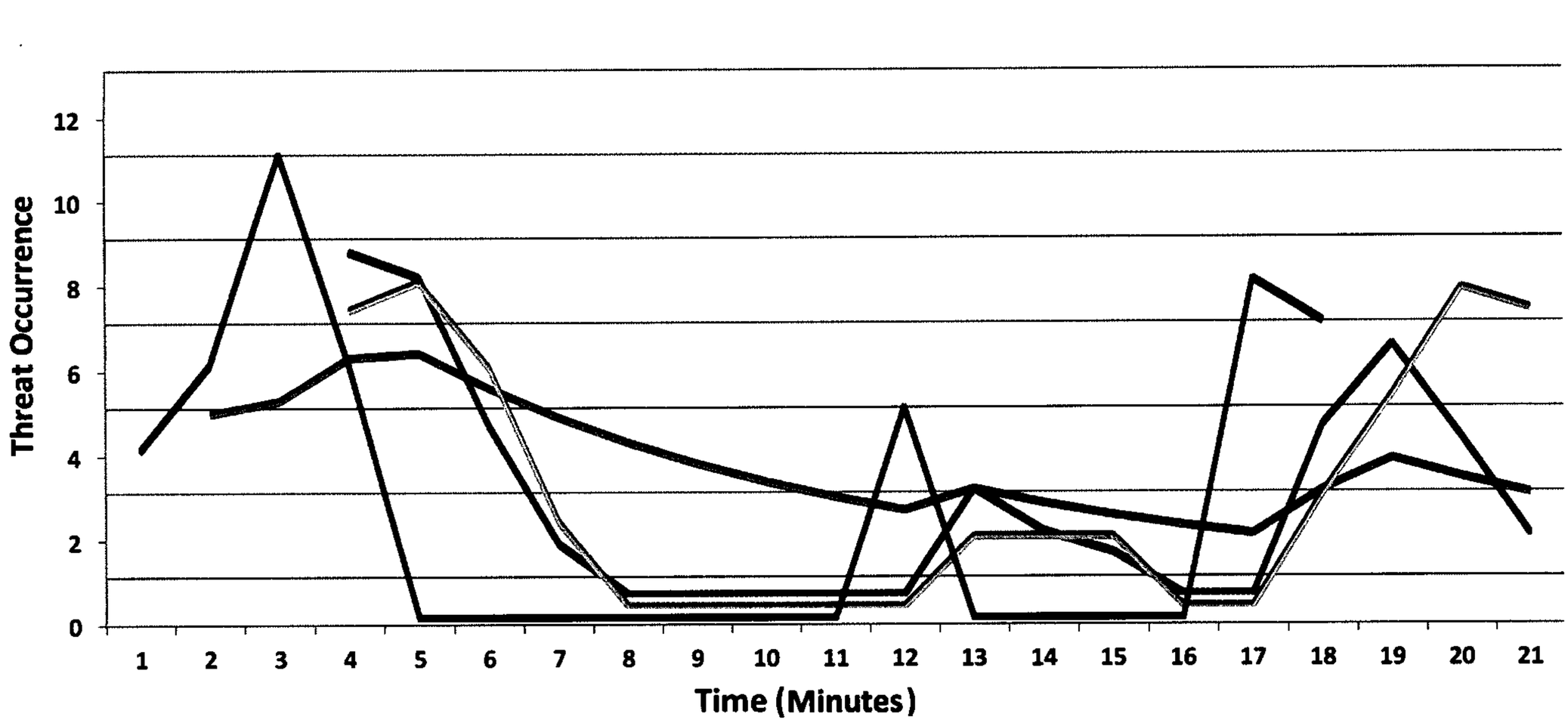


Figure 55: Forecasting of the High Severity (Buffer Overflow) Threat

8.6 Conclusion

In this section, we have performed time series analysis on the darknet data. We initiated by defining time series by pinpointing its types and objectives. We progressed by analyzing the correlation and the dynamics of the darknet data by using detrended fluctuation analysis. We further extended our work by testing for predictability for certain darknet traffic and threats targeting specific destinations. After achieving that, we utilized three forecasting techniques to predict the values (packet counts and occurrences) for backscattering and scanning traffic as well as a medium and a high priority darknet threat. Such prediction would aid in mitigation of future threat occurrences.

9 Threat Correlation

As accessibility to and dependability on the cyberspace continues to surge, cyber incidents continue to flourish and target more victims. According to the 2009 UK cyber-crime report [89], more than 3 millions cyber incidents, with an increase of 4.2% from 2008, were recorded targeting approximately 50% of UK businesses, averaging a loss of £30,000 per business and a general loss in excess of £500,000. Another research revealed that 90% of United States companies have been the target of a cyber attack, with 80% suffering a significant financial loss [90]. As a result, the U.S. Department of Defense has officially announced that it now considers the cyberspace to be the fifth dimension of warfare. In addition, the Canadian cyber security strategy report [91] eleborated that in a recent one year period, 86% of large Canadian organizations had suffered a cyber attack where the loss of intellectual property as a result of these attacks doubled between 2008 and 2010.

There is a crucial need to further analyze the threats detected and described previously. This section aims to investigate and generate threat patterns or clusters of darknet threats that co-occur targeting a specific victim. We believe that such cyber threat intelligence would aid in better understanding of threat patterns and their consequences by providing interpretations of the threat scenarios.

9.1 Approach

The goal is to investigate the interdependence and inter-correlation of darknet threats. Particularly, we aim to answer the following questions: Are there any threats targeting a specific victim that follow a certain pattern? Moreover, if some of the co-occurring threats appear in a darknet traffic, how confidently one can predict the existence of other threats? To investigate this, we employed the technique of frequent pattern mining (frequent item-set) and association rule mining [92]. The outcome of these methods are threat association rules that could be used as an input to a classification model that is able to predict and hence mitigate future threat occurrences. These techniques have been proven to be very successful for identifying hidden patterns in DNA sequences, customer purchasing habits, text categorization, and many other applications of pattern recognition. The proposed threat correlation approach is a three-step process, namely, frequent pattern mining, association rule generation from each frequent threat-set, and rule analysis by applying various correlation techniques. Each of these steps is detailed below.

9.2 Frequent Pattern Mining

An item-set or a pattern is group of two or more objects that appear together. An item-set is a *frequent* pattern if its members appear together for some minimum number of times. In the context of threat analysis, an item or an object is a threat and an item-set is the threat-set. Table 9, which is used for illustration and explanation purposes, shows 10 threat-sets, one threat-set per row.
Let $T = \{t_1, \cdots, t_m\}$ denote the universe of all threats detected from the given darknet feeds F. Suppose a threat-set $T_i \subseteq T$ detected at a time interval τ_i represents a row or an instance in the threat Table 9. This table shows ten threat-sets captured at time intervals $\{\tau_1, \cdots, \tau_{10}\}$.

Time Intervals	Identified Threats
τ_1	$\{t_2, t_5, t_7, t_9\}$
τ_2	$\{t_2, t_5, t_7\}$
τ_3	$\{t_2, t_5\}$
τ_4	$\{t_1, t_5, t_7\}$
τ_5	$\{t_4, t_5, t_7\}$
τ_6	$\{t_3, t_6, t_8\}$
τ_7	$\{t_4, t_5, t_8\}$
τ_8	$\{t_3, t_6, t_8\}$
τ_9	$\{t_2, t_5, t_8\}$
τ_{10}	$\{t_1, t_5, t_7, t_8, t_9\}$

Table 9: Vectors of Darknet Threats

Let $T_i \subseteq T$ be a threat-set or a pattern in the threat table. A pattern that contains k threats is a $k-pattern$. For instance, $\tau_1 = \{t_2, t_5, t_7, t_9\}$ is a $4-pattern$. Similarly, the support of a pattern T_i is the percentage of the instances in the threat table containing T_i. A pattern T_i is a *frequent pattern* if the support of T_i is greater than or equal to some user specified minimum support threshold, which is a real number in an interval of $[0, 1]$. Further explanation of these terms is given in Example 9.1.

Example 9.1 *Consider Table 9. Suppose the user-specified threshold* $min\_sup = 0.3$*, which means that a pattern* $T_i = \{t_1, \cdots, t_k\}$ *is frequent if at least 3 out of the 10 rows contain all threat-items in* T_i*. For instance,* $\{t_2, t_5, t_7, t_9\}$ *is not a frequent pattern because it has support 1/10 = 0.1. Similarly,* $\{t_2, t_5\}$ *is a frequent 2-pattern because it has support 4/10 = 0.4 and contains two threats. Likewise,* $\{t_5, t_8\}$ *is a frequent 2-pattern with support* $3/10 = 0.3$.

There are various data mining algorithms for extracting frequent patterns, such as the Apriori [92], FP-growth [93], and ECLAT [94]. In this work, we employ the Apriori algorithm as it is easy to comprehend and it has been validated in several text mining studies [95, 96]. Below, we provide an overview of the Apriori algorithm.

Apriori is a level-wise iterative search algorithm that uses frequent k-patterns to explore the frequent $(k+1)$-patterns. First, the set of frequent 1-patterns is found by scanning the threat table, accumulating the support count of each threat-set, and collecting the threat patterns T_i that has $support(\{T_i\}|T) \geq min\_sup$. The resulting frequent 1-patterns are then used to find frequent 2-patterns, which are then used to find frequent 3-patterns, and so on, until no more frequent k-patterns can be found. The generation of frequent $(k+1)$-pattern from frequent k-patterns is based on the following Apriori property.

Property 9.1 (Apriori property) *All nonempty subsets of a frequent pattern must be frequent.*

By definition, a pattern T_i' is not frequent if *support* $(T_i'|T) < min\_sup$. The above property implies that adding a threat t to a non-frequent pattern T_i' will not make it frequent. Thus, if a k-pattern T_i' is not frequent, then there is no need to generate $(k+1)$-pattern $T_i' \cup t$ because $T_i' \cup t$ is also not frequent. The following example shows how the Apriori algorithm exploits this property to efficiently extract all frequent patterns or threat-sets. For a formal description, we refer the reader to [92].

Example 9.2 *Consider Table 9 with* $min\_sup = 0.3$. *First, identify all frequent 1-patterns by scanning the threat table once to obtain the support of every threat-set. The items having support* ≥ 0.3 *are frequent 1-patterns, denoted by* $L_1 = \{\{t_2\}, \{t_5\}, \{t_7\}, \{t_8\}\}$. *Then, join* L_1 *with itself, i.e.,* $L_1 \bowtie L_1$, *to generate the candidate set* $C_2 = \{\{t_2, t_5\}, \{t_2, t_7\}, \{t_2, t_8\}, \{t_5, t_7\}, \{t_5, t_8\}, \{t_7, t_8\}\}$ *and scan the threat table once to obtain the support of every pattern in* C_2. *Identify the frequent 2-patterns, denoted by* $L_2 = \{\{t_2, t_5\}, \{t_5, t_7\}, \{t_5, t_8\}\}$. *Similarly, perform* $L_2 \bowtie L_2$ *to generate* $C_3 = \{t_5, t_7, t_8\}$. *By scanning the threat table once, we found that* $\{t_5, t_7, t_8\}$ *is not frequent, i.e., 3-pattern* L_3 *is empty. The finding of each set of frequent k-patterns requires one full scan of the rows in Table 9.*

9.3 Association Rule Mining

The selected frequent patterns or frequent threat-sets are used to investigate the correlation and inter-dependence of the subsets of each frequent threat-set. This can be achieved by applying association rule mining techniques [97]. For this, all 1-patterns are deleted as they contain only one threat and thus can not be associated with any other threat. The 2-patterns threat-sets are used to extract single-dimensional association rules while the 3-patterns and higher patterns are used to construct multi-dimensional association rules. To construct an association rule of threats, we need to calculate the confidence for each frequent threat-set. Assume we have a threat-set $\{t_a, t_b\}$, for which the association rule would be $\{t_a\} \Rightarrow t_b$. An association rule has a confidence c in the threat table T, where c is the percentage of threat-sets in T containing t_a that also contains t_b. This statement is mathematically expressed in Equation 2.

$$\text{confidence}(\{t_a\} \Rightarrow t_b) = \rho(t_b|t_a) = \frac{support\{t_a \cup t_b\}}{support\{t_a\}} \tag{2}$$

Having support-count of $(t_a \cup t_b)$ and t_a, we can calculate confidence$(\{t_a\} \Rightarrow t_b)$ using Equation 2. Once the frequent threat-sets are extracted, the related association rule of a frequent threat-set T_i can be constructed as follows:

- Generate all non-empty subsets of T_i
- For every non-empty subset S, construct a rule $(S \Rightarrow (T_i - S)$, provided the $\frac{support(T_i)}{support(S)} \geq min\_conf$

9.4 Correlation Analysis

In order to investigate the interdependency of the threats, various correlation techniques including χ^2, *cosine* measure, and *lift* [97] can be used. In the current study, we use *lift*, which is easy to understand as it is based on probabilities and its results are interpretable by non-technical domain experts without the help of data mining experts. The correlation technique *lift* measures how many times more often t_a

and t_b occur together than expected if they are statistically independent. It is mathematically expressed as follows:

$$lift(t_a, t_b) = \frac{\rho(t_a \cup t_b)}{\rho(t_a)\rho(t_b)} \tag{3}$$

If the value of Equation 3 is equal to 1 then threats t_a and t_b are independent and therefore have no correlation; otherwise they are either negatively correlated (i.e., $lift < 1$) or positively correlated (i.e., $lift > 1$).

9.5 Experimental Results

In this section, we implement our correlation approach on the threat data. For that purpose, we apply the Apriori algorithm implemented in WEKA data mining toolkit [98]. In summary, the Apriori takes the threat table in ARFF or CSV file types as input along with the user-defined parameters including minimum support min_sup and confidence c, and generate association rules.

To assess our approach, we experimented with different threats that are detected and mentioned in Table 7. Specifically, we targeted studying the correlation and inter-dependency among those identified threats to build threat association rules that target specific threats. The experimental results are validated by employing sequential rule mining techniques for correlating same set of threats discussed in Section 9.7. Consequently, the generated rules can be used to build associative classification model for predicting the occurrences of specific threats in real-time darknet traffic.

In general, the threat-rules generated by the Apriori, provided the threshold is kept low, is usually very large. However, we can tune and filter the results to bring the rules to a manageable level by applying the following steps:

- Choosing a suitable value for the minimum support[19] based on the occurrence count of the targeted threat.
- Taking into consideration the size of the association rules by specifying the number of items per threat-set as input to the algorithm.
- Removing threats, before the analysis, that do not contribute in information gain (i.e., a threat that is absent during the analyzed period).

We were interested to study the clusters of threats that co-occur targeting a specific victim. By accomplishing this, we may reveal threat patterns or sequences and corresponding consequences that target a specific organization. In the current work, we selected a portion of darknet providers network blocks as the target of attacks. Specifically, we restricted the target of the attacks to a network block consisting of 253 hosts.

Table 10 represents our data mining results from the top 5 darknet destination networks. For confidentiality and privacy matters, we anonymize some sensitive information. This table discloses the

[19]The choice of selecting a minimum support threshold is inversely proportional to the number of generated threat-sets. Eg: By increasing the minimum support value, the number of generated association rules will decrease.

s.16(2)(c)
s.20(1)(c)

analyzed IP blocks, their corresponding identified threat patterns or association rules, coupled with their lift[20], confidence[21] and occurrence per day[22].

Darknet Feed Providers	Analyzed Address Blocks	Association Rules	Confidence	Lift	Count
		1.$\{t_7, t_8, t_9\} \Rightarrow t_{10}$	0.63	3.64	282
		2.$\{t_{10}, t_{14}, t_{13}\} \Rightarrow t_{11}$	0.56	7.06	306
		3.$\{t_{10}, t_{15}, t_4\} \Rightarrow t_1$	0.76	1.54	193
		4.$\{t_{12}, t_{11}, t_{13}\} \Rightarrow t_{10}$	0.92	3.81	359
		5.$\{t_{10}, t_7, t_8, t_9, t_{13}\} \Rightarrow t_4$	0.55	10.75	218
		6.$\{t_{10}, t_8, t_9, t_{13}\} \Rightarrow t_{12}$	0.26	3.68	348
		7.$\{t_7, t_8, t_9\} \Rightarrow t_{10}$	0.43	4.12	113
		8.$\{t_4, t_8, t_9\} \Rightarrow t_{10}$	0.98	6.6	102
		9.$\{t_{10}, t_7, t_8, t_9, t_{13}\} \Rightarrow t_{11}$	0.41	3.56	260
		10.$\{t_7, t_8, t_9, t_{11}, t_{13}\} \Rightarrow t_{10}$	0.82	3.65	131

Table 10: Darknet Threat Patterns

In the sequel, we provide an interpretation to the above identified threat patterns.

Threat Patterns Interpretation:
Please refer to the numbered association rules in Table 10 as a reference to the below interpretations.

1. This rule discloses the following information.

2. This rule reveals the subsequent information.

[20]The lift indicates whether the identified threat patterns are correlated together. A value above 1 means that they frequently co-occur when they target a specific victim.
[21]Looking at an association rule, the confidence is the likelihood to obtain a threat on the right-hand side of the $\Rightarrow$ by inspecting those that appear on the left-hand side of it.
[22]This is a strong indication that the identified threat pattern is valid since it frequently occurs per unit of time (a day in our current analysis).

3. This rule can be interpreted as the following.

4. This rule, in fact, presents

5. This rule discloses the following information.

6. This rule can be interpreted as the following.

7. This rule is in fact a series of

8. This rule unveils the following information.

9. Although this rule and rule 10 are syntactically different, however contextually, they can be interpreted similarly. They disclose that

9.6 Sequential Pattern Mining

In this section, we refine the frequent patterns that were generated in the previous step by taking into consideration the sequence of occurrence of the identified threats. The purpose is to investigate whether or not the order of the threats is significant and how it can affect the results of the threat correlation method. Moreover, the frequent sequence mining of this section can be used for cross-validation of the results of the frequent threat-set mining, discussed in Section 9.2.
The research problem of sequential pattern mining is defined in [99] as follows: given a set of sequences and a user-defined minimum support threshold *min_sup*, sequential pattern mining finds all frequent subsequences; the subsequences whose occurrence frequency in the set of sequences is not less than *min_sup*. In the context of threat correlation, a sequential pattern is a set of threat-sets and each threat-set is a sequence of threats.
Example 9.3 provides a more explicit explanation of sequential patterns by extending the concept of frequent patterns as described in Section 9.2. Let $U = \{t_1, \cdots, t_m\}$ be the universe of all items or threats detected in the darknet data. An item-set or a threat-set is a subset of the detected threats U. A threat-sequence T is an ordered list of threat-sets and is denoted by $< T_1 T_2 \cdots T_n >$ where each ordered threat-set T_j is called a threat-element. Each threat-element T_j is a sequence of individual threats denoted by $T_j =< t_1 \cdots t_l >$.

Time Intervals	Identified Threats
τ_1	$< t_1 < t_1 t_2 t_3 >< t_1 t_3 > t_4 < t_3 t_6 >>$
τ_2	$<< t_1 t_4 > t_3 < t_2 t_3 >< t_1 t_5 >>$
τ_3	$<< t_5 t_6 >< t_1 t_2 >< t_4 t_6 > t_3 t_2 >$
τ_4	$< t_5 t_7 < t_1 t_6 > t_3 t_2 t_3 >$

Table 11: Vectors of Sequential Darknet Threats

Example 9.3 *Consider Table 11. Suppose the set of threats detected is* $\{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ *and let the user-specified threshold min_sup* $= 0.2$. *A threat-sequence* $< t_5 t_7 < t_1 t_6 > t_3 t_2 t_3 >$ *has three elements, namely,* $(t_5 t_7)$,$(t_1 t_6)$, *and* $(t_3 t_2 t_3)$. *Similarly,* $< t_5 t_7 < t_1 t_6 > t_3 t_2 t_3 >$, *having a length of 7, is a 7-threat-pattern.* t_1 *contributes one to the length and* t_3 *contributes two to the length of the sequence while the support-count of both is equal to one. Threat-sequence* $< (t_1 t_2) t_3 >$ *is a subsequence of* τ_1 *and* τ_3, *having a support of 2 and dubbed as a 3-threat-pattern since its length is equal to 3.*

There are several algorithms proposed for sequential pattern mining including Generalized Sequential Patterns (GSP) [100], FreeSpan [101], Sequential PAttern Discovery using Equivalent classes (SPADE) [102], and PrefixSpan [102]. GSP, FreeSpan, and SPADE are based on the Apriori algorithm discussed in Section 9.2, and thus suffer from scalability and efficiency problems. However, PrefixSpan is based on Pattern-Growth [93], which means that it does not generate a candidate set at each step and hence it is one of the most efficient methods among all the sequential pattern mining algorithms [103]. In the current study we have employed PrefixSpan. The algorithm takes vectors of threats ordered by the time of their occurrence, along with a minimum support threshold and a maximum length of the output patterns. It produces a list of threat-patterns and the occurrence frequency of each pattern.

In our experiments, we have used the same dataset that has been used in the frequent pattern mining section in order to compare the results of the two techniques.

The experimental results of PrefixSpan show that more than 80% of the frequent sequence patterns are overlapping with those listed in Table 10. This can be considered as a validation step of the previous results and their corresponding scenario interpretations that were given before. Note that, the frequencies of the new patterns are relatively less than the unordered frequent patterns, which is usual in an ordered set. All of the threat-sets listed in Table 10 for the two ISPs, namely, and are detected by PrefixSpan with a matching minimum support. In case of the remaining three ISPs, the number of extracted frequent sequence patterns is less than the unordered patterns, which could be interpreted as a false positive case for the former technique. In addition, we identified some interesting sequences with high occurring frequencies that went undetected by the Apriori algorithm, which can be explained as a false negative case of the previous technique.

The results of the frequent sequence mining are summarized in Table 12. Moreover, the description of the selected additional sequential threat patterns is given below.

Darknet Feed Providers	Analyzed Address Blocks	Frequent Sequential Patterns
		1. $< t_7 t_8 t_9 t_{10} >$ 2. $< t_{10} t_{14} t_{13}\ t_{11} >$ 3. $< t_{10} t_{11} t_{12}\ t_{13} >$ 4. $< t_{10} t_{12} t_{14}\ t_4 >$
		5. $< t_{12} t_{11} t_{13} t_{10} >$ 6. $< t_{11} t_{13} t_8 >$
		7. $< t_{10} t_7 t_8 t_9 t_{13} t_4 >$ 8. $< t_{10} t_8 t_9 t_{13} t_{12} >$ 9. $< t_{10} t_{14} t_{11} t_{13} t_{16} >$ 10. $< t_{10} t_7 t_8\ t_9 t_{11} >$
		11. $< t_7 t_8 t_9 t_{10} >$ 12. $< t_4 t_8 t_9 t_{10} >$
		13. $< t_7 t_8 t_9 t_{11} t_{13} t_{10} >$

Table 12: Darknet Threat Sequential Patterns

Threat Sequential Patterns Interpretation: This section provides an interpretation of the identified sequential patterns. Please refer to the numbered patterns in Table 12 as a reference to the below interpretations. Note that, patterns (1,2,5,7,8,11,12,13) were previously elaborated in Section 9.5.

- **Pattern 3:** This sequential pattern depicts the following. An attacker

- **Pattern 4:**

- **Pattern 6:** This pattern reveals the following information.

- **Pattern 9:** This pattern can be interpreted as follows.

- **Pattern 10:** This pattern discloses the following information.

9.7 Sequential Rule Mining

Sequential pattern mining has shown interesting results in mining darknet threats. However, we wanted to refine our profiling of darknet threats to produce better results and more elaborated data mining experiments. A sequential threat pattern $t_a t_b t_c$ denotes that t_c is always preceded by $t_a t_b$ but it does not reveal how many times $t_a t_b$ occurs without t_c. Moreover, it does not demonstrate that if $t_a t_b$ occurs, how confidently one can expect the occurrence of threat t_c. This information can be captured by employing sequential rule mining. A number of algorithms including RuleGrowth [104], Top-KRules [105], and CMRules [106] have been presented in previous studies. In the current study, we have used RuleGrowth, as implemented in SPMF[23], which is an open-source data mining framework written in Java. Rule $t_a t_b \Rightarrow t_c$ says that whenever a threat-sequence $t_a t_b$ appears, one can note with certain probability or confidence that threat t_c will appear as well. This can as well be used to investigate whether a rule is strong or weak, in the sense that how strongly threats t_a, t_b, and t_c are correlated to each other.

In order to predict the occurrences of sequential patterns, Fournier-Viger *et al.* defined the RuleGrowth algorithm, which mines sequential rules common to several sequences by pattern-growth. Their algorithm is elaborated to address the problem of prediction, which is based on sequential rule mining. RuleGrowth algorithm finds rules correlating two items and recursively expands them by

[23]http://www.philippe-fournier-viger.com/spmf/

scanning a database searching for items that can be added to their left or right parts.

The algorithm is designed by finding solutions for four different problems. The first problem is related to the left or right expansion of the rules. Let us consider the following rule $a, b \Rightarrow c, d$. By performing a left and then a right expansion of $a \Rightarrow c$, we can obtain $a, b \Rightarrow c, d$. It can also be obtained by performing right and then left expansion of $a \Rightarrow c$. The authors decided to perform a right expansion after a left expansion and not the opposite. This has been chosen to avoid generating repeated rules. The second problem consists of the number of rules that may need to be expanded to find out other rules. These rules should not generate other valid rules if we expand them. A solution is based on the following lemma: an expansion of rule r such that $support(r) < min\_sup$ will not result in a valid rule. The third problem consists of the fact that some rules can be found several times by performing left or right expansions with different items. In order to solve this problem, an item is only added to an itemset of a rule if the item is greater than each item in the itemset based on lexicographic ordering. The fourth problem lies in finding an efficient process to determine each item that can expand a rule to generate a valid rule. The solution is to scan the sequences containing each rule to count the support of an item that may expand the rule. In order to accomplish this, each item that appears in at least *min_sup* in sequences, would result in a rule having at least *min_sup*. In order to improve this process, the authors introduced optimized generation of rules where each item that can expand the left itemset of a rule must appear before the last occurrence of its right itemset in at least $min\_sup * |S|$ sequences containing the rule. A similar approach is applied for the right itemset of a rule as well. RuleGrowth algorithm keeps a record for each sequence of the first and last occurrence of a rule itemsets.

The experiments are generated from outputs of threats collected from a scan of darknet channel, where we isolated five ISPs, namely,

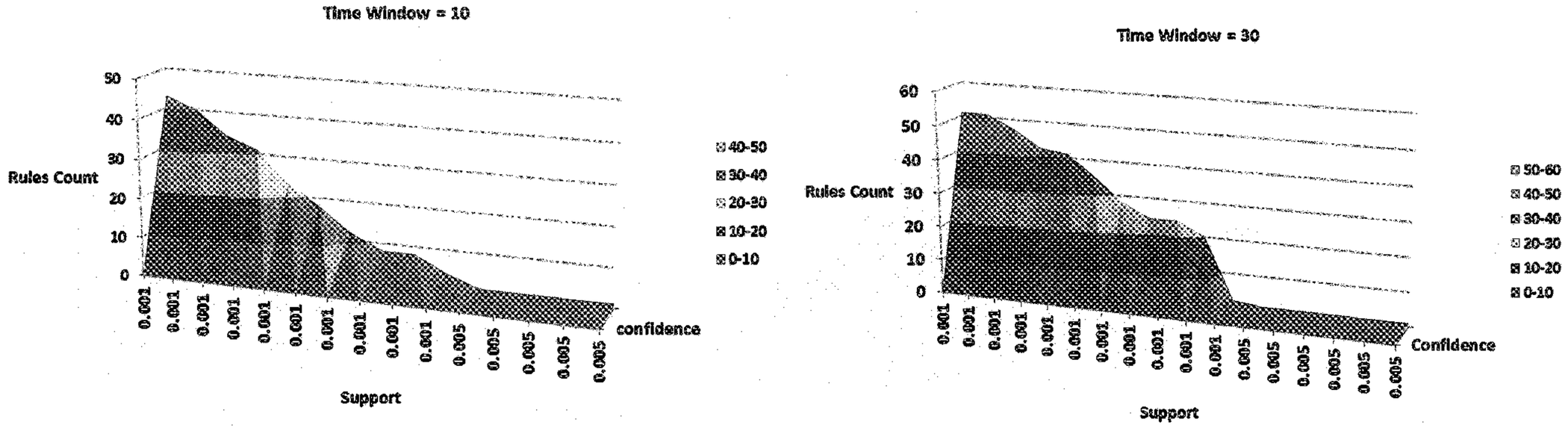
Due to the huge amount of data and computation that have resulted from the algorithm, we focued on analyzing data related to ISP only. We gauge the effect of user-defined minimum support threshold, confidence, window size on the number of rules generated. Figures 56 depicts the number of rules generated by varying the minimum support threshold *min_sup* while keeping window size equal to ten seconds. We repeated the same set of experiments for a window size equal to thirty, as shown in Figure 57.

The experimental result indicates that the number of generated rules is inversely proportional to minimum support and confidence. The count of rules varies mainly from five to eight ranges, thus the variation depends mainly on the support and confidence rather than the window size. For this purpose, we modified the time window to one minute, which resulted in a dramatic change in the number of rules, depicted in Figure 58. Moreover, the rules are richer comparing to other results obtained from other time Windows and they show low level of overlapping with other results obtained previously.

In order to refine the obtained results, we consider the rules with high confidence, i.e., greater than 15%. The total number of rules generated are 858. Figure 59 depicts the distribution of the number of rules per confidence.

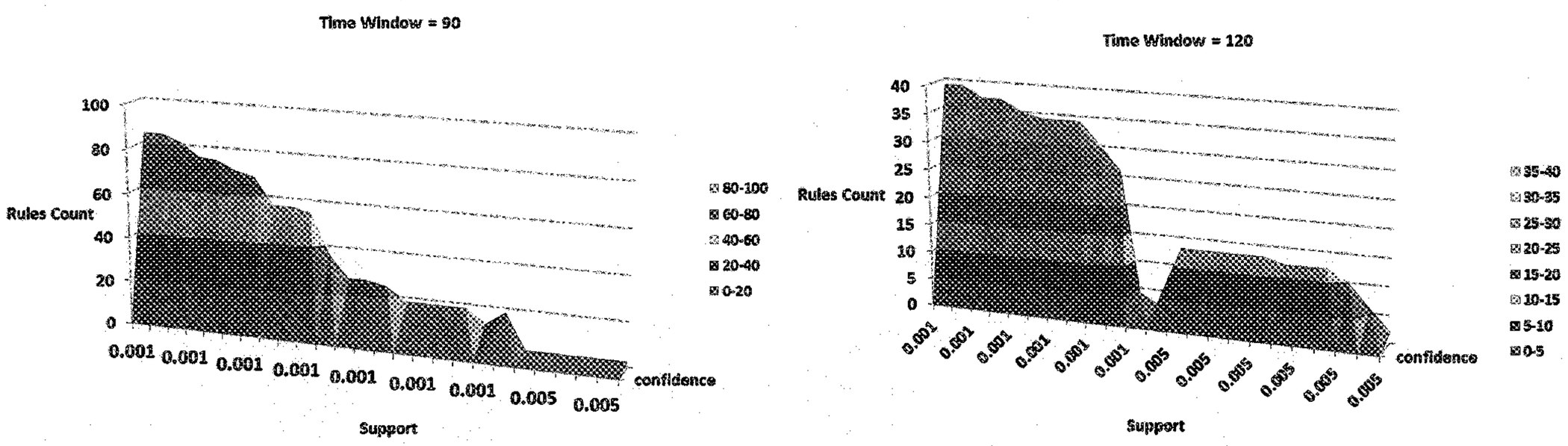
By analyzing the rules, we noticed that the majority of them are composed of lengthier sequences

s.20(1)(c)



(a) 10 (b) 30

Figure 56: No. of Rules vs. Support (a. Window Size = 10, b. Window Size = 30)



(a) 90 (b) 120

Figure 57: No. of Rules vs. Support (a. Window Size = 90, b. Window Size = 120)

as compared with the results obtained from other variations of window sizes. This may be explained by the fact that the confidence in this case is higher. We enumerated 194 rules that have a maximum confidence. The length of the left part varies from 3 to 7 whereas the length of the right part of the rule varies from 1 to 5. This result is due to the fact that the more items (threats) we have in the left part of the rule, the more we are confident to expect the occurrence of other threats. Due to the large number of obtained rules, we chose some rules such that they stand for the simplest representation of all the rules (i.e., rules sharing common left and right parts). Table 13 illustrates the selected *strong* rules, i.e., rules having maximum confidence.

We noticed from Table 13 that all the rules start with t_{14} followed by t_4, t_{13} or t_{10}. This means that all the rules are initiated with ICMP echo, which can lead to a rule pattern where we expect the occurrence of some events with full confidence. An ICMP echo has to be followed by other recognition information actions such as another ping or a windows traceroute, or an unreachable ICMP ping.

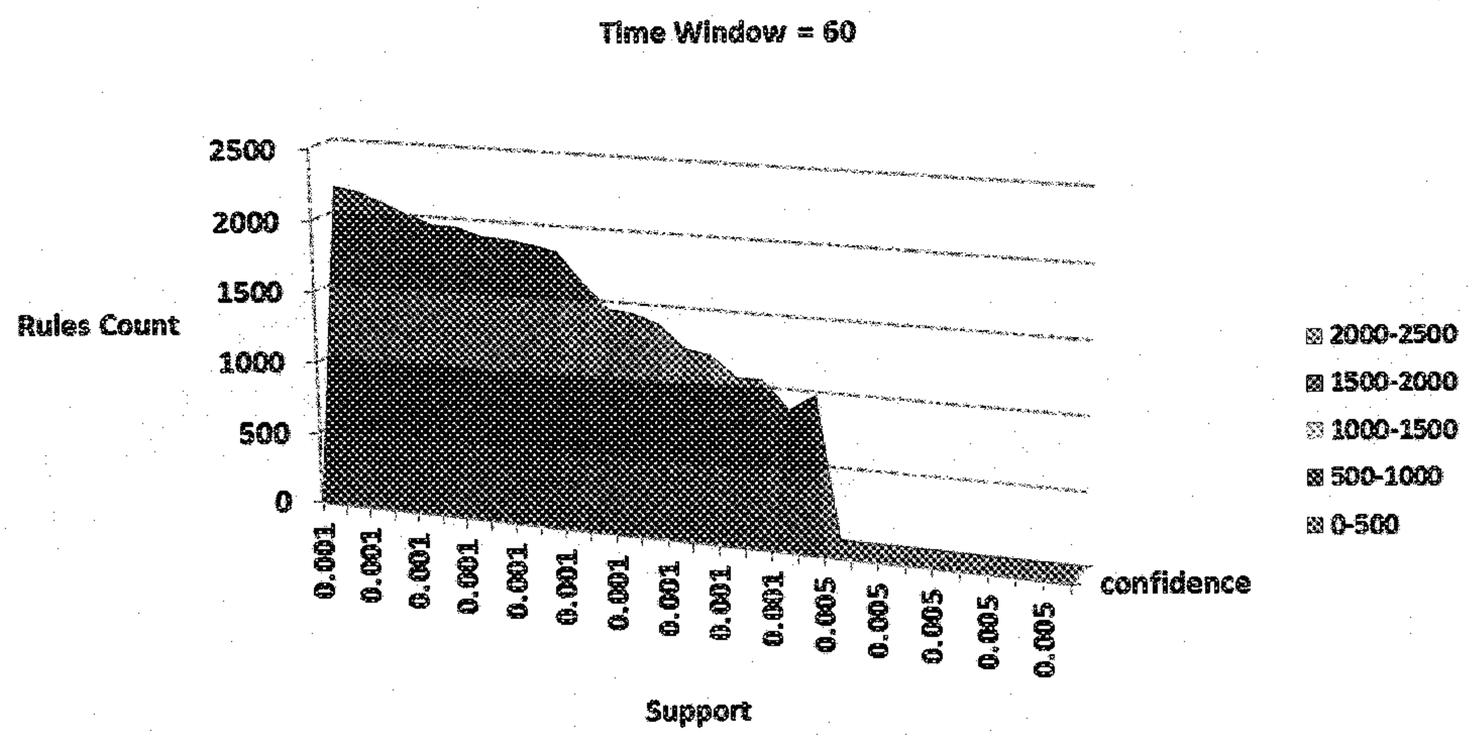


Figure 58: No. of Rules vs. Support (Window Size = 60)

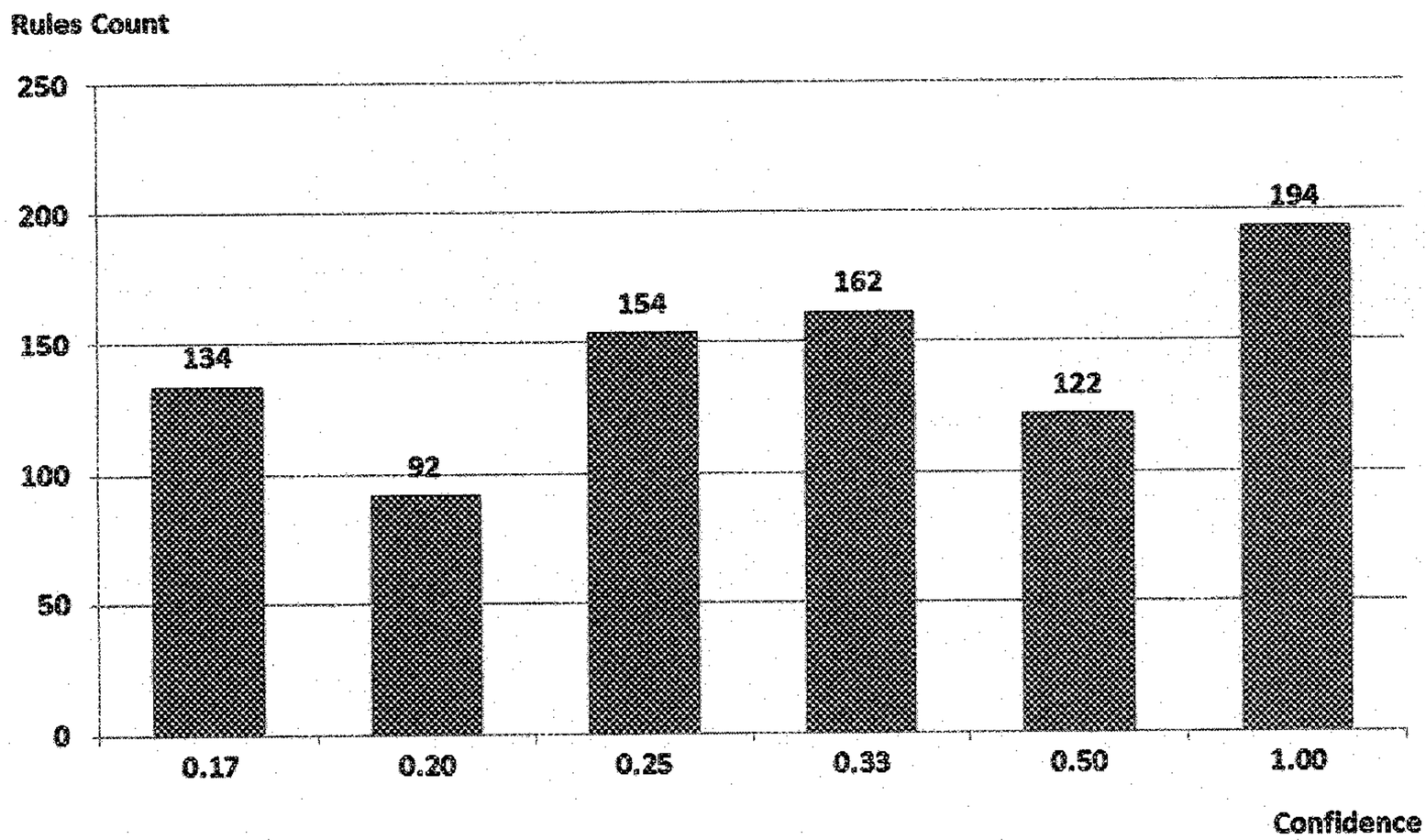


Figure 59: No. of Rules vs. Confidence

These actions can be initial symptoms toward the occurrence of events where malformed ICMP packets are detected or the appearance of packets that have exceeded the maximum number of hops. As a result, more recognition activities can be perpetuated from BSD, Unix hosts or DSL Routers. These sequential rules explains the scenario of scanning activities in darknet since it generates forecasting rules with a 100% confidence. However, it does not provide forecasting about more severe threats such as buffer overflow or denial of service attacks since they have fewer frequencies than scanning activities. In order to find rule patterns that contain severe threats, we decided to look for the rules with low

Rules	Confidence
$\{t_{14}, t_4, t_{16}\} \Rightarrow t_{12}$	1.00
$\{t_{14}, t_4, t_{13}, t_{16}\} \Rightarrow t_7$	1.00
$\{t_{14}, t_4, t_{16}, t_{12}\} \Rightarrow t_8$	1.00
$\{t_{14}, t_4, t_{13}, t_{16}\} \Rightarrow t_{11}$	1.00
$\{t_{14}, t_{13}, t_{11}\} \Rightarrow t_{12}$	1.00
$\{t_{14}, t_{10}, t_4, t_{13}, t_{16}\} \Rightarrow t_{11}$	1.00

Table 13: Sample Threat Sequential Rules

minimum support and low confidence.

In order to find rules representing more severe threats, we considered the output from ISP with a time window of ten seconds and a confidence higher than 10% and lower than 33%. The number of generated rules is not important if we compare it to the one generated from ISP. However, it shows that if we consider a lower support and time window, we can find rules that contain threats that are not limited to scanning activities. Table 14 presents the obtained rules.

Rules	Confidence
$\{t_5\} \Rightarrow t_{11}$	0.16
$\{t_{17}\} \Rightarrow t_{11}$	0.11
$\{t_{10}, t_{17}\} \Rightarrow t_{11}$	0.11
$\{t_{10}, t_5\} \Rightarrow t_{11}$	0.16
$\{t_5\} \Rightarrow t_{13}$	0.19
$\{t_5\} \Rightarrow t_{12}$	0.32

Table 14: Sequential Rules Representing Severe Threats

In these rules, we notice the occurrence of threat t_5, indicating information disclosure by attempting to discover the port where the Remote Procedure Call (RPC) statd system process runs. Such operation can lead to the discovery of vulnerability related to the RPC protocol. This action can be followed by two exception events, namely, t_{13} or t_{12} with confidences equal to 19% and 32%, respectively. These exceptions denote unreachable ICMP ping datagram or a packet that has exceeded allowable hops in network. In order to get rules for prediction of severe threats, we need to consider small time window and low support.

10 Conclusion and Recommendations

We presented in this final report our activities on the analysis of darknet space for predictive indicators of cyber threat activity. In this respect, we presented a taxonomy of definitions and concepts in relation to darknet technologies. Moreover, we discussed the state-of-the-art proposals in terms of science and technology and we identified the underlying research gaps. In addition, we established the elements of an advanced architecture and testbed, which aim to automatically generate cyber intelligence. At this point of time, we are able to automatically generate cyber intelligence that is primarily used for:

- Profiling darknet in terms of its nature of traffic and underlying content as well as its embedded threats. Such information generates indicators of cyber threat activity and allows the in-depth understating of darknet activities and incidents.
- Detection of malicious IPs and domains, the geo-localization of drop locations of sensitive information (i.e., stolen financial information, private credentials, etc.) and the identification of malicious connections and their parameters that Botnets often utilize.
- Providing brand protection services to assist financial institutions, Internet service providers and other corporations in mitigating online fraud and cyber crimes.
- Generating future predicted values of specific darknet threats. Such prediction aids in mitigation of future threat occurrences.
- Identifying co-occurring threat patterns that target specific organizations, which aim to prevent future threat incidents.

The outcome of the presented work, specifically in the area of darknet analysis, provides a significant indicator of the health of the Internet. In this report, we managed to exploit the advantages of correlating various sources of data to generate actionable cyber intelligence that is of benefit to National Defence agencies, public safety and industrial corporations. To analyse the dynamics of self-similar behavior in the darknet traffic, we employed mathematical and machine learning artifacts to uncover correlation features, model and forecast darknet threats.

We assert that such work could be enhanced by performing active analysis on real-time data rather than applying passive analysis. We expect as well to leverage the existing capability by establishing efficient data correlation engines and investing in reliable and accurate forecasting models for the purpose of detecting, preventing, assessing, mitigating and attributing cyber attacks.

References

[1] P. Biddle, P. England, M. Peinado, and B. Willman, "The darknet and the future of content distribution," in *proceedings of ACM Workshop on Digital Rights Management.* ACM, 2002, pp. 1–6.

[2] Biddle, Peter and England, Paul and Peinado, Marcus and Willman, Bryan, "The Darknet and the Future of Content Protection," in *proceedings of the 2003 Digital Rights Management.* Springer, 2003, pp. 344–365.

[3] "Zeus Malware: Threat Banking Industry," http://www.unisys.com/unisys/ri/wp/detail.jsp?id=1120000970016810153, UNISYS, Tech. Rep., last accessed on November 25, 2011.

[4] "Zeus, King of the Underground Crimeware Toolkits," http://www.symantec.com/connect/blogs/zeus-king-underground-crimeware-toolkits, retrieved on November 25, 2011.

[5] P. Sinha, A. Boukhtouta, V. Belarde, and M. Debbabi, "Insights from the analysis of the mariposa botnet," in *Risks and Security of Internet and Systems (CRiSIS), 2010 Fifth International Conference on*, oct. 2010, pp. 1 –9.

[6] K. Zetter, "How Digital Detectives Deciphered Stuxnet, the Most Menacing Malware in History," http://www.wired.com/threatlevel/2011/07/how-digital-detectives-deciphered-stuxnet/all/1, retrieved on July 13, 2011.

[7] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," in *the Proceedings of International workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability.* Springer-Verlag, 2001, pp. 46–66.

[8] I. Clarke, S. G. Miller, T. W. Hong, and O. Sandberg, "Protecting Free Expression Online with Freenet," *IEEE Internet Computing*, vol. 2, pp. 40–49, 2002.

[9] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *the Proceedings of the 13th USENIX Security Symposium*, 2004, pp. 303–320.

[10] R. Geambasu, T. Kohno, A. A. Levy, and H. M. Levy, "Vanish: Increasing Data Privacy with Self-Destructing Data," in *the proceedings of the 18th USENIX Security Symposium.* USENIX Association, 2009, pp. 299–316.

[11] M. Wood and B. Hoffman, "Veiled: A Browser Darknet," http://www.blackhat.com/presentations/bh-usa-09/HOFFMAN/BHUSA09-Hoffman-VeilDarknet-SLIDES.pdf, retrieved in December 2009.

[12] K. J. Higgins, "Researchers Build Anonymous, Browser-Based 'Darknet'," http://www.darkreading.com/security/encryption/217801293/researchers-build-anonymous-browser-based-darknet.html, retrieved in June 2009.

[13] C. Perkins, "Ad hoc On-Demand Distance Vector (AODV) Routing," http://www.ietf.org/rfc/rfc3561.txt.

[14] E. Bangeman, "Study: BitTorrent Sees Big Growth," http://arstechnica.com/old/content/2008/04/study-bittorren-sees-big-growth-limewire-still-1-p2p-app.ars, retrieved on January 05, 2012.

[15] C. Zhang, P. Dhungel, D. Wu, Z. Liu, and K. W. Ross, "BitTorrent Darknets," 2009, pp. 1–9.

[16] V. Paxson, "Bro: a System for Detecting Network Intruders in Real-time," *Computer Networks (Amsterdam, Netherlands: 1999)*, vol. 31, no. 23–24, pp. 2435–2463, 1999.

[17] M. Eto, K. Sonoda, D. Inoue, K. Yoshioka, and K. Nakao, "A Proposal of Malware Distinction Method Based on Scan Patterns Using Spectrum Analysis," in *Neural Information Processing*, ser. Lecture Notes in Computer Science, C. Leung, M. Lee, and J. Chan, Eds. Springer Berlin / Heidelberg, 2009, vol. 5864, pp. 565–572.

[18] M. Eto, D. Inoue, J. Song, J. Nakazato, K. Ohtaka, and K. Nakao, "NICTER: a Large-Scale Network Incident Analysis System: Case Studies for Understanding Threat Landscape," in *the Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, ser. BADGERS '11. New York, NY, USA: ACM, 2011, pp. 37–45. [Online]. Available: http://doi.acm.org/10.1145/1978672.1978677

[19] F. Gagadis and S. D. Wolthusen, "Topological Models and Effectiveness of Network Telescopes," pp. 1–11, 2008.

[20] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, "Network Telescopes: Technical Report," CAIDA, San Diego Supercomputer Center, University of California, San Diego, Tech. Rep., 2004.

[21] C. Leita, V.-H. Pham, O. Thonnard, E. Ramirez, F. Pouget, E. Kirda, and M. Dacier, "The Leurre.com Project: Collecting Internet Threats Information using a Worldwide Distributed Honeynet," in *the Proceedings of the 1st WOMBAT Workshop on Information Security Threat Data Exchange(WISTDE)*, 2008.

[22] Corrado Leita and Ken Mermoud and Marc Dacier, "ScriptGen: an Automated Script Generation Tool for honeyd," in *proceedings of ACSAC*, 2005, pp. 203–214.

[23] Corrado Leita and Marc Dacier and Frédéric Massicotte, "Automatic Handling of Protocol Dependencies and Reaction to 0-Day Attacks with ScriptGen Based Honeypots," in *the Proceedings of RAID*, 2006, pp. 185–205.

[24] S. Zanero, "Observing the Tidal Waves of Malware: Experiences from the WOMBAT Project," *Information Technology for Real World Problems, Vaagdevi International Conference on*, vol. 0, pp. 30–35, 2010.

[25] Georgios Portokalidis and Asia Slowinska and Herbert Bos, "Argos: an Emulator for Fingerprinting Zero-Day Attacks," in *proceedings of ACM SIGOPS EUROSYS*, 2006.

[26] Paul Baecher and Markus Koetter and Maximillian Dornseif and Felix Freiling, "The nepenthes platform: An efficient approach to collect malware," in *the Proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID.* Springer, 2006, pp. 165–184.

[27] Van Horenbeeck, M., "The SANS Internet Storm Center," in *the Proceedings of WOMBAT Workshop on Information Security Threats Data Collection and Sharing WISTDCS '08*, 2008, pp. 17–23.

[28] H. Bos, "Shelia: a Client-Side Honeypot for Attack Detection," http://www.cs.vu.nl/~herbertb/misc/shelia/.

[29] Galante, A. and Kokos, A. and Zanero, S., "BlueBat: Towards Practical Bluetooth Honeypots," in *the Proceedings of IEEE International Conference on Communications(ICC '09)*, 2009, pp. 1–6.

[30] P. Kijewski, C. Overes, and R. Spoor, "The HoneySpider Network fighting client-side threats," *honeyspidernet*, pp. 1–15, 2008.

[31] Portokalidis, Georgios and Homburg, Philip and Anagnostakis, Kostas and Bos, Herbert, "Paranoid Android: Versatile Protection For Smartphones," *Network Security*, pp. 347–356, 2008.

[32] Cova, Marco and Leita, Corrado and Thonnard, Olivier and Keromytis, Angelos and Dacier, Marc, "An Analysis of Rogue AV Campaigns," in *proceedings of Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science. Springer-Berlin / Heidelberg, 2010, vol. 6307, pp. 442–463.

[33] Comparetti, Paolo Milani and Salvaneschi, Guido and Kirda, Engin and Kolbitsch, Clemens and Kruegel, Christopher and Zanero, Stefano, "Identifying Dormant Functionality in Malware Programs," in *proceedings of 2010 IEEE Symposium on Security and Privacy (SP)*, 2010, pp. 61–76.

[34] (2000) DShield: community-based collaborative firewall log correlation system @ONLINE. [Online]. Available: http://www.dshield.org/

[35] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson, "The Internet Motion Sensor: A Distributed Blackhole Monitoring System," in *the Proceedings of Network and Distributed System Security Symposium (NDSS'05)*, San Diego, CA, Feb. 2005, pp. 1–13.

[36] E. Cooke, M. Bailey, Z. M. Mao, D. Watson, F. Jahanian, and D. McPherson, "Toward understanding distributed blackhole placement," in *proceedings of WORM'04*, 2004, pp. 1–11.

[37] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of Internet background radiation," in *proceedings of IMC'04.* ACM, 2004, pp. 1–14.

[38] V. Yegneswaran, P. Barford, and D. Plonka, "On the Design and Use of Internet Sinks for Network Abuse Monitoring," in *the Proceedings of the 7 th International Symposium on Recent Advances in Intrusion Detection (RAID*, 2004, pp. 146–165.

[39] QoSient, LLC., "Argus: Auditing Network Activity," http://www.qosient.com/argus/, 2000–2011.

[40] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click modular router," *ACM Trans. Comput. Syst.*, vol. 18, pp. 263–297, 2000.

[41] Team Cymru, Inc., "Team Cymru Community Services: The Darknet Project," http://www.team-cymru.org/Services/darknets.html, 2011, retrieved in March 2011.

[42] J. Riordan, D. Zamboni, and Y. Duponchel, "Building and deploying Billy Goat, a worm-detection system," IBM Zurich Research Laboratory, pp. 1–12, May 2006.

[43] P. Szor, *The Art of Computer Virus Research and Defense.* Addison-Wesley Professional, 2005.

[44] K. Fukuda, T. Hirotsu, O. Akashi, and T. Sugawara, "A PCA Analysis of Daily Unwanted Traffic," in *the Proceedings of International Conference on Advanced Information Networking and Applications(AINA)*, 2010, pp. 377–384.

[45] K. Limthong, F. Kensuke, and P. Watanapongse, "Wavelet-Based Unwanted Traffic Time Series Analysis," *Computer and Electrical Engineering, International Conference on*, vol. 0, pp. 445–449, 2008.

[46] S. E. Robertson and S. K. J., "Relevance weighting of search terms," *Journal of the American Society for Information Science*, vol. 27, no. 3, pp. 129–146, 1976.

[47] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[48] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *the Proc. of European Conf. Machine Learning (ECML'98).* Springer Verlag, 1998, pp. 137–142.

[49] L. Bilge, E. Kirda, C. Kruegel, M. Balduzzi, and S. Antipolis, "Exposure : Finding malicious domains using passive dns analysis," *Sophia*, pp. 1–17, 2011.

[50] K. Fukuda, T. Hirotsu, O. Akashi, and T. Sugawara, "Correlation among piecewise unwanted traffic time series," in *proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2008)*, 2008, pp. 1–5.

[51] Sourcefire, "Snort: Open-Source Network Intrusion Prevention and Detection System (IDS/IPS)," http://www.snort.org/, last accessed on December 2011.

[52] G. N. Purdy, *Linux iptables: Pocket Reference.* O'Reilly, 2004, ISBN: 978-0-596-00569-6.

[53] M. Rash, *Linux Firwalls: Attack Detection and Response with iptables, psad, and fwsnort*, 3rd ed. San Francisco: No Starch Press, Inc., 2007, ISBN: 978-1-59327-141-1.

[54] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of internet background radiation," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 27–40. [Online]. Available: http://doi.acm.org/10.1145/1028788.1028794

[55] Who.is, 2012, Available at:

[56] P. D. Online, 2012, Available at:

[57] M. S. TechCenter, "Microsoft Security Bulletin MS09-018 - Critical," Available at: http://technet.microsoft.com/en-us/security/bulletin/MS09-018.

[58] S. Shah, "Top Ten Web Attacks," Available at: http://www.blackhat.com/presentations/bh-asia-02/bh-asia-02-shah.pdf.

[59] L. S. Online, "MS Terminal Service Cracking," Available at: http://www.carnal0wnage.com/papers/lso_ms_terminal_server_cracking.pdf.

[60] CorruptedDataRecovery.com., "Port Number 'udp," Available at: http://www.corrupteddatarecovery.com/Port/ udp-Port-Type-

[61] "Snort," Available at: http://www.snort.org.

[62] "The Bro Network Security Monitor," http://bro-ids.org/, last accessed on December 2011.

[63] C. Charras and T. Lecroq, *Handbook of Exact String Matching Algorithms.* King's College Publications, 2004.

[64] P. D. Michailidis and K. G. Margaritis, "On-Line String Matching Algorithms: Survey and Experimental Results," in *International Journal of Computer Mathematics.* Taylor and Francis, 2001, vol. 76. [Online]. Available: http://www.informaworld.com/10.1080/00207160108805036

[65] A. Thomas, "Rapid: Reputation based approach for improving intrusion detection effectiveness," in *Information Assurance and Security (IAS), 2010 Sixth International Conference on*, aug. 2010, pp. 118 –124.

[66] N. I. of Standards and T. N.-N. C.-A. System, "Vulnerability summary for cve-2007-2931," 2011, Available at: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-2931.

[67] Sourcefire-Snort, 2011, Available at: http://www.snort.org/search/sid/

[68] G. Sandbox, "Malware Analysis with GFI SandBox (formerly CWSandbox)," http://www.gfi.com/malware-analysis-tool/.

[69] "Geolocation and Online Fraud Prevention from MaxMind," http://www.maxmind.com, last accessed on December 2011.

[70] Brand Abuse Terms - Mark Monitor. https://www.markmonitor.com/resources/brand-abuse-terms.php, last accessed on March 2012.

[71] Anti-Phishing Working Group. http://www.antiphishing.org/, last accessed on March 2012.

[72] PhishTank. http://www.phishtank.com/, last accessed on March 2012.

[73] Microsoft Internet Explorer - SmartScreen Filter: frequently asked questions. http://windows.microsoft.com/en-US/windows7/smartscreen-filter-frequently-asked-questions, last accessed on March 2012.

[74] Mozilla Firefox - Phishing and Malware Protection. http://www.mozilla.org/en-US/firefox/phishing-protection/, last accessed on March 2012.

[75] Google Chrome and Browser Security. https://www.google.com/chrome/intl/en/more/security.html, last accessed on March 2012.

[76] Stuxnet worm 'targeted high-value Iranian assets'. http://www.bbc.co.uk/news/technology-11388018, last accessed on March 2012.

[77] Number of the week: 780 new malicious programs designed to steal users online banking data detected every day. http://www.kaspersky.com/about/news/virus/2012/Number_of_the_week_780_new_malicious_programs, last accessed on March 2012.

[78] The Spamhaus Project - Frequently Asked Questions. http://www.spamhaus.org/faq/section/Glossary#169, last accessed on March 2012.

[79] Security Information Exchange. [Online]. Available: https://sie.isc.org/

[80] GFI ThreatTrack^{TM}. http://www.gfi.com/internet-security-feeds, last accessed on March 2012.

[81] last viewed January 2010.

[82] last accessed on March 2012.

[83] last accessed on March 2012.

[84] , last accessed on March 2012.

[85] , last accessed on March 2012.

[86] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 414–422.

[87] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 264–271. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390190

[88] University of Oxford, "Software - FastDFA: Detrended Fluctuation Analysis," http://www.eng.ox.ac.uk/samp/dfa_soft.html.

[89] S. Fafinski and N. Minassian, "Uk cybercrime report," 2009, Available at: http://www.garlik.com/file/cybercrime_report_attachement.

[90] S. Hinde, "The Law, Cybercrime, Risk Assessment and Cyber Protection," *Computers & Security*, pp. 90–95, 2003.

[91] Public Safety Canada, "Canada's Cyber Security Strategy," 2009, Available at: http://www.publicsafety.gc.ca/prg/ns/cbr/_fl/ccss-scc-eng.pdft.

[92] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. of the 1993 ACM SIGMOD international conference on Management of data*, vol. 22, no. 2. Washington, D.C., United States: ACM, June 1993, pp. 207–216.

[93] J. Han and J. Pei, "Mining frequent patterns by pattern-growth: methodology and implications," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 2, pp. 14–20, December 2000.

[94] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions of Knowledge and Data Engineering (TKDE)*, vol. 12, pp. 372–390, 2000.

[95] B. C. M. Fung, K. Wang, and M. Ester, "Hierarchical document clustering using frequent itemsets," in *Proc. of the 3rd SIAM International Conference on Data Mining (SDM)*. San Francisco, CA: SIAM, May 2003, pp. 59–70.

[96] J. D. Holt and S. M. Chung, "Efficient Mining of Association Rules in Text Databases," in *the Proc. of the 8th ACM International Conference on Information and Knowledge Management (CIKM)*, Kansas City, Missouri, United States, 1999, pp. 234–242.

[97] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)*, 2nd ed. Morgan Kaufmann, Jan. 2006.

[98] M. A. H. Ian H. Witten, Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Morgan Kaufmann, January 2011.

[99] Agrawal, Rakesh and Srikant, Ramakrishnan, "Mining sequential patterns," in *Proc. 1995 int. Conf. Data Engineering (ICDE'95)*, Taipei, Taiwan, March 1995, pp. 3–14.

[100] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *EDBT*, 1996, pp. 3–17.

[101] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu, "Freespan: frequent pattern-projected sequential pattern mining," in *KDD*, 2000, pp. 355–359.

[102] M. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Machine Learning*.

[103] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Mining sequential patterns by pattern-growth: the PrefixSpan approach," *Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1424–1440, 2004. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1339268

[104] P. Fournier-Viger, R. Nkambou, and V. S.-M. Tseng, "Rulegrowth: mining sequential rules common to several sequences by pattern-growth," in *SAC*, 2011, pp. 956–961.

[105] P. Fournier-Viger and V. S. Tseng, "Mining top-k sequential rules," in *ADMA (2)*, 2011, pp. 180–194.

[106] P. Fournier-Viger, U. Faghihi, R. Nkambou, and E. M. Nguifo, "Cmrules: Mining sequential rules common to several sequences," *Knowl.-Based Syst.*, vol. 25, no. 1, pp. 63–76, 2012.