Noroff University College

Bachelor in
Cyber Security

Slammer Worm: From Outbreak to Stagnation? Investigating Its Network Activity Over Time

ALEXANDER DANIELSEN

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR IN **CYBER SECURITY**

SUPERVISOR
Barry Irwin

Noroff University College
Norway
May, 2025

Abstract

The SQL Slammer worm, known for its for its unprecedented speed of spread, significantly disrupted global internet infrastructure, causing denial of service when it first emerged in early 2003. Despite early containment efforts, uncertainties persist about its presence, as observations of the worm continued for many years after its initial outbreak. This research addresses these uncertainties by systematically analysing the long term activity of the Slammer worm using Internet Background Radiation data collected from a network telescope over the period 2005-2024, specifically focusing on UDP ports 1433 and 1434. Using various packet analysis tools, this research inspected the packets within the dataset, recovering and examining the payloads. Different methods, including payload size analysis and MD5 hash value comparisons, were applied to detect changes in the worms payload over time, as well as to search for possible new malicious signatures. Additionally, the analysis separated Slammer related traffic from other UDP activities, providing a clearer picture of the security challenges surrounding the targeted ports. Doing this, the analysis identified a large number of packet belonging to Slammer throughout its lifetime, as well as for each year individually, with the results visualized through tables and graphs for easier understanding Geolocation methods were applied to detect regional hotspots and patterns, uncovering important shifts in worms activity distribution and its preferable regions during different peaks of the study period.

Keywords: *SQL Slammer, network telescope, IBR, UDP, ports 1433/1434*

Acknowledgements

I would like to express gratitude to professor Barry Irwin for providing access to telescope data, which was main object for the research data of this project. Additional his structure supervision, clear deadlines, and regular meetings significantly contributed to the timely completion of projects milestones. Special appreciation also go to my wife, whose support on the family front was critical throughout this period. Her patience and understanding, particularly in taking additional family responsibilities during the period of research and writing, were invaluable to the projects successful completion.

Mandatory Declaration

1.	I hereby declare that the submission answer is my own work, and that I have not used other sources other than as is referenced and cited correctly, or received help other than what is specifically acknowledged.	Yes
2.	**I further declare that this submission:** • Has not been used for another exam in another course at Noroff University College, at another department/university/college at home or abroad. • Does not refer to or make use of the work of others without acknowledgement. • Does not refer to my own previous work unless stated. • Has all the references given in the bibliography. • Is not a copy, duplicate or copy of someone else's work or answer. • Is not generated using AI generation tools.	Yes
3.	I am aware that a breach of any of the above is to be regarded as cheating and may result in cancellation of the exam and exclusion from universities and university colleges in Norway, cf. University and University College Act § 12-4 and Noroff University College Regulation § 4-5.	Yes
4.	I am aware that all components of this assignments may be checked for plagiarism and other forms of academic misconduct.	Yes
5.	I hereby acknowledge that I have been taught the appropriate ways to use the work of other researchers. I undertake to paraphrase, cite, and reference according to the acceptable academic practices, in accordance with the rules and guidelines, as taught.	Yes
6.	I am aware that Noroff University College will process all cases where cheating is suspected in accordance with the college's guidelines.	Yes

Publication Agreement

Authorisation for electronic publication of the thesis: Through submission you are accepting that Noroff University College has a perpetual, and royalty free right to retain a copy of work for its own internal use, and has the right to make work publicly available - considering any restrictions to publication.

Name: Alexander Danielsen

Date: June 5, 2025

Contents

List of Figures

List of Tables

1

Introduction

1.1 Introduction

The Slammer worm caused one of the most severe disruptions to the internet ever recorded, infecting thousands of systems within minutes. This incident exposed vulnerabilities in network security, especially in systems that were not patched and had far reaching consequences for network infrastructures worldwide (Moore et al., 2003; Schultz et al., 2003). While the initial chaos it caused was significant, activity related to the Slammer worm seems to have reduced in recent years, leading to questions about its current level of threat or potential stagnation.

The research problem explored in this project focuses on the uncertainty surrounding the current activity of the Slammer worm. Although patches were swiftly provided after the outbreak, there have been sporadic detections of related traffic even beyond its peak years (Chindipha & Irwin, 2017). This raises concerns for networks today regarding the worms present status and significance. Assessing whether Slammer still poses a risk, particularly in light of modern security practices.

This research aims to explore the progression of the Slammer worms activities from its peak years to the present day (2005-2024). By examining network traffic data spanning a period, particularly focusing on the last few years this research will evaluate how the worms behaviour has evolved and whether it still generates significant traffic in modern networks. Even if the worms presence is limited this discovery will provide insights into the lifecycle of this network worm and its enduring impact on network security.

1.3 Problem Statement

Although Slammers initial impact has long passed, the question remain about whether the worm truly disappeared or simply became harder to detect. The worms behaviour over time, especially when observed through the lens of long term network telescope data, may reveal signs of continued

presence. Sporadic detections and unexplained payloads raise questions about whether Slammer has been quietly surviving and hiding in unmaintained networks or has it inspired copycat traffic patterns.

The problem statement guiding this research is: How has the Slammer worms activity changed from its original outbreak, and does it continue to pose a threat to modern networks? If it does still exist, under what conditions or within which regions it can be found?

1.4 Research Objectives

The primary objective of this research is to analyse the evolution and movement of the Slammer worms activity from 2005 to 2024.

In order to achieve this primary objective, the following sub-objectives have been identified:

- Examine how the payloads observed in the Slammer worms network traffic have changed over time and assess if the payloads can be accurately identified.
- Identify if there are any other types of attacks targeting this focused UDP traffic on ports 1433 and 1434, focusing on any non Slammer related traffic.
- Analyze whether certain geographical regions or networks have been more frequently affected by Slammer worm traffic, and how that has changed over time.

1.5 Research Methodology

This research uses an empirical, data-driven methodology based on long term passive internet traffic collection. It involves analysing IBR captured by a network telescope over nearly two decades. Tool such as Tshark, Tcpdump and Wireshark were used for packet inspection and traffic filtering. Also Python and Bash scripts were used to repeat the process over whole dataset as well as for visualization of the results, and eventually for automating part of the data process.

Rather to relying on theoretical models, the approach in this project focuses on packet analysis and involves filtering by UDP ports 1433 and 1434, payload validation using size, hash values, and checksum methods, and geolocation of IP sources via WHOIS database queries to map patterns and detect anomalies over time.

1.6 Primary Data Set

The dataset that was used for this research is Internet Background Radiation (IBR) traffic that has been captured by a Network Telescope operated on the SANREN Network by The National Integrated Cyber-Infrastructure System (NICIS). The dataset was provided by Professor Barry Irwin. The data consists of PCAP files filtered by relevant ports and covering the period from 2005 to 2024

The primary ethical consideration that needs to be undertaken is ensuring that the monitored range of the network telescope is not disclosed. There are no foreseen risks in the data set and/or data processing.

1.7 Scope and Limits

The research is constrained by the limitations of the provided dataset. Because UDP based traffic can in be spoofed, so true IP addresses of the initial infection source would possibly not be correct. The network telescope monitors only a portion of the IPv4 traffic, allowing some worm or other malicious activity to remain unseen,like the one using IPv6. The dataset contains missing records due to periodic outages and capture interruptions. These factors together may affect the ability to see the whole picture of Slammer worms evolution and accuracy of the observed activity over time.

1.8 Document Structure

The remainder of this report is organized as follows. Chapter 2 presents a literature review of prior work related to the SQL Slammer worm, Internet Background Radiation, long-term malware persistence, literature on tools, and passive network monitoring. Chapter 3 outlines the research methodology used, including the structure of the data analysis framework, filtering criteria, validation tools, and the approach used to extract and interpret packet data. Chapter 4 presents the findings from the empirical analysis, including observed traffic volumes, geographic patterns, payload validation results, and other notable insights from the dataset. Chapter 5 concludes the study by evaluating how well the research objectives have been met, summarizing the main contributions, and identifying potential directions for future work.

2

Literature Review

2.1 Literature Review Introduction

The following chapter presents examination of existing research, historical studies, and established methodologies related to the SQL Slammer worm and the tools and approaches used to understand long-term worm activity. It begins by exploring concepts such as IBR and network telescopes in observing large-scale malicious events and global scanning activities. Building on this foundation, the review proceeds to examine the characteristics and history of the SQL Slammer worm, highlighting both its initial outbreak an documented instances of its reappearance over time.

The following sections address Network Telescope use and Internet Background Radiation (Section 2.2), the SQL Slammer worms origins and initial outbreak (Section 2.3), its long-term persistence and re-emergence (Section 2.4), packet analysis tools and techniques (Section 2.5), and intrusion detection systems and observations (Section 2.6). By combining insights from these areas, this literature review provides the context for this study and identifies the key gaps we aim to fill, the SQL Slammer worms activity over time and evolution.

2.2 IBR and Network Telescopes

Internet Background Radiation or IBR refers to irregular traffic that circulates on the internet due to various factors such as misconfiguration, malicious activities, and automated scanning. This constant stream of irregular packets provides valuable insight into global internet activity, including the spread of worms, botnet, and other malicious entities (Bortoluzzi et al., 2023; Wustrow et al., 2010). This

suggests that monitoring IBR can improve understanding of the global internet threat landscape and plays a foundational role in network security research.

Network telescopes, also know as darknets, are in reality passive internet sensors that monitor large, unused portion of IPv4 address space to collect IBR. By capturing traffic directed to IP addresses that should not receive any legitimate communication, network telescopes effectively record the “background noise” of the internet (Wustrow et al., 2010). An advantage of sustained IBR monitoring that can be useful for this research is its ability to unpower worm reappearances that may happen months or even years after an initial outbreak. By continuously tracking suspicious traffic over multiple years, researchers can identify whether a worm like SQL Slammer has actually died out or merely gone dormant in isolated networks.

If a person cook a large pot of soup and mix it well enough, tasting any portion of it should give you a good indication of the taste of the entire pot. Similarly, analysing Internet Background Radiation data from a network telescope, which monitors irregular traffic hitting unused IP-space, provides insights that reflect the broader state of the global or regional internet (Bortoluzzi, 2024)

Network telescopes have been widely used to study IBR and understand malicious activities on the internet. According to Wustrow et al. (2010), network telescopes provide a unique vantage point for observing irregular traffic, enabling researchers to detect global scanning actives and spread of malware. The data collected through these telescopes allow for empirical analysis of the wide internet events without interfering with normal network operations.

Irwin (2013) conducted a baseline study of potentially malicious activity across multiple network telescopes, highlighting the importance of diverse sensor placement to capture complete view of IBR. Their findings indicated that malicious traffic pattern vary significantly across different regions and networks, this highlighting the need for widespread and geographically diverse monitoring. Despite all the positive and handy about this tool, network telescopes have a known limitation. As noted by Irwin (2013), passive monitoring using this type of setup cannot observe full TCP connection attempts, as reply traffic is typically blocked and only the first request packet is captured. However, this limitation does not affect the analysis of UDP-based activity such as Slammer, since UDP lacks a three-way handshake altogether. For more complex scenarios involving TCP-based threats, integrating honeypot systems with network telescopes can mitigate this issue by providing more detailed information on malicious activity (Krause, 2021).

Bortoluzzi et al. (2023) proposed a distributed architecture for capturing IBR using cloud based network telescopes. This approach scales effectively and provide better coverage by utilizing the global infrastructure of cloud service providers. The distributed nature of such telescopes allows for improved detection of global threats, including sporadic worm activities. Their work highlights that cloud based telescopes can be be effective without relying on IPv6 traffic, focusing on IPv4 data collection, which aligns with the scope of this study.

Several empirical studies have employed network telescopes to monitor and analyse IBR demonstrating their effectiveness in uncovering larger malicious activities (Bortoluzzi et al., 2023; Irwin, 2013; Wustrow et al., 2010). For instance, Chindipha and Irwin (2017) investigated the SQL Slammer worm and reported signs of its continued presence across multiple regions, demonstrating the present relevance of unsolicited network traffic research at that time.

In an earlier study, Irwin et al. (2007) undertook a geopolitical analysis of long-term network telescope data and noted a markedly high volume of traffic originating from African states, including South Africa. Even though their analysis primarily concerned regional traffic patterns, that observation is useful for IBR research more generally, because it speaks to how geographic and infrastructural factors can shape how malicious scanning and worms spread. This highlights the importance of geographically diverse network telescope deployments, which can capture region-specific traffic patterns and help create a more complete picture of internet threats on a global scale.

2.3 Background on SQL Slammer

The SQL Slammer worm, also known as the Sapphire worm, was a significant event in cybersecurity history due to its unique speed and widespread impact (Moore et al., 2003). Discovered in January 2003, it exploited a buffer overflow vulnerability in Microsoft SQL Server (Microsoft Corporation, 2002; MITRE Corporation, 2002), leading to one of the fastest spreading worm outbreaks ever recorded. Within ten minutes, it infected over 90% of vulnerable hosts, causing significant disruptions worldwide. This initial outbreak in quickly escalated internet-wide, overwhelming many ISP backbones within minutes. Large organizations worldwide spent days mitigating the resulting congestion. Notably, administrators struggled was the worms random scanning, which flooded networks at full speed without using a handshake to allow filtering. Among the important services impacted were financial and governmental services, both of which experienced major downtime in January 2003. Banks ATM networks suffered outages, and certain airline reservation systems stalled under the wave of Slammer traffic. Reports also indicated extremely high scanning activity congesting backbone routers, underscoring how even a small UDP payload could produce extensive collateral damage (Paxson et al., 2003). The worms simplicity and the damage it inflicted highlighted the urgent need for improvements in patch management and defences that can react automatically, especially since patches for the vulnerability had already been available prior to the outbreak (Schultz et al., 2003). However, according to Kristoff (2023), the patches released by Microsoft to cover this buffer overflow vulnerability in SQL servers caused multiple problems in different types of product systems, which may have been one of the reasons some companies deliberately chose not to apply the update.

The worms fast spread was boosted by its small size which was just 376 bytes which allowed it to fit within a single UDP packet (Moore et al., 2003). By comparison, the first internet worm, known as the Morris Worm, was approximately 10 kilobytes in size, and Code Red, another infamous worm, was 4 kilobytes (Chen et al., 2014). Despite its simplicity, the Slammer worm caused widespread network disruptions, impacting services such as banking systems, airline reservations, and emergency services (Travis et al., 2003).

As Schultz et al. (2003) described, SQL Slammer spread by exploiting a buffer overflow vulnerability, having severe impact on the global network and marking a new era in understanding worm propagation. This incident demonstrated that even small payloads could cause massive disruptions, especially when security measures were inadequate or improperly deployed. Furthermore, Moore et al. (2003) highlighted the importance of implementing traffic management techniques, as the worms impact was exacerbated by the lack of these controls, leading to equipment failures and unexpected service disruptions.

Slammer exploited a known vulnerability on UDP port 1434 associated with Microsoft SQL server

2000 by transmitting short packets 376 bytes in size to randomly generated IP addresses (Cisco Systems, Inc., 2003). This random scanning mechanism allowed the worm to spread rapidly, as it continuously probed different parts of the internet for new hosts to infect (Bajaj & Guha Roy, 2004). Because Microsoft SQL services still runs on these ports in many corporate data centers, prior worms and scanning events historically cluster on UDP ports 1433 and 1434, making them a prime location point for detecting reinfection or residual worm traffic. However, because Slammer did not follow predictable address sequence and used minimal packet sizes, security professionals had difficulty spotting and blocking it spread before it reached vulnerable systems, resulting in significant network disruption (Travis et al., 2003).

Accurate detection of worms is further is further complicated by factors like network disruption and the similarity of malicious traffic to legitimate activities. Wei and Mirkovic (2008) addressed the need to correct disruption based errors in network telescope observations, improving the accuracy of worm dynamics analysis.

2.4 Long-Term Persistence and Re-Emergence

Although patches for the original vulnerability were released, evidence of Slammer worm traffic continued to appear well beyond its initial outbreak. Building on an eight year dataset Chindipha and Irwin (2017), identified unexpected spikes in Slammer-related activity between 2014 and 2016. Their analysis pinpointed several /24 nett-blocks that consistently sent worm-like traffic, suggesting that older or neglected machines remained online and unpatched, thereby sustaining the infection cycle. By plotting year by year packet volumes, the authors demonstrated how incomplete patching and reliance on outdated systems enabled Slammer to persist. Their findings highlighted the difficulty in eliminating malware completely, as even a small pool of vulnerable hosts can preserve threats long after their initial discovery.

In a more recent research, Kristoff (2023) monitored approximately 300 hosts worldwide using a distributed set of network sensors, analysing traffic for the Slammers typical 376 byte packets on UDP port 1434 over a two week window. His findings indicated that, nearly two decades after the worms initial outbreak, no active traces of Slammer were detected in contemporary network traffic. While this suggests Slammers presence has largely disappeared, the network filters and operational lessons introduced during its rapid spread still impacting current security practices, demonstrating how major outbreaks can result in enduring defence measures. This outcome also highlights a central issue in present research: even if Slammer now appears dormant in most environments, question remain about whether it could still persist on unmonitored environment or outdated systems, or perhaps it has evolved into a new form distinct from the Slammer variant we have seen in previous outbreaks.

Understanding worm propagation is crucial for analysing potential impact. Stanifor et al. (2004) examined the theoretical limits of worm spread speed, introducing the concept of “flash worms”, worms capable of infecting large numbers of hosts almost instantly. Historical examples like Slammer and Witty demonstrate the efficiency of random scanning techniques, although with notable differences Slammer used a remarkably small 376-byte UDP payload, sending probes continuously, while Witty employed larger packets and scanned addresses randomly at a slightly slower pace. Bajaj and Guha Roy (2004) described and captured the same random-spreading behaviour of the SQL Slammer worm, which they visualized in their paper. The flow diagram shown in Figure 2.1 is inspired by their

illustration of the worms propagation process. Despite differences between Slammer and Witty, both worms generated massive amounts of redundant traffic, repeatedly targeting non-vulnerable or already infected IP addresses, ultimately reducing overall efficiency. By contrast, Stanifor et al. (2004) and later Bulygin (2013) proposed “flash worms,” which rely on precomputed lists of confirmed vulnerable hosts to accelerate the infection process. Instead of guessing addresses, a flash worm assigns subsets of the target list to newly infected nodes, eliminating wasted scans and enabling the potential to infect millions of machines in seconds, far surpassing the impressive speeds of Slammer and Witty. Both studies further analysed how factors such as inaccurate host lists, dropped packets, or deeper infection trees could affect coverage. They suggested techniques like shallower trees, doubling infection paths, or splitting the list at runtime to maintain resilience. While Slammer and Witty highlight the speed achievable through bandwidth-saturating UDP traffic, flash worms, leveraging optimized hit-lists, present a theoretical model for even faster propagation, provided accurate knowledge of vulnerable targets is available.

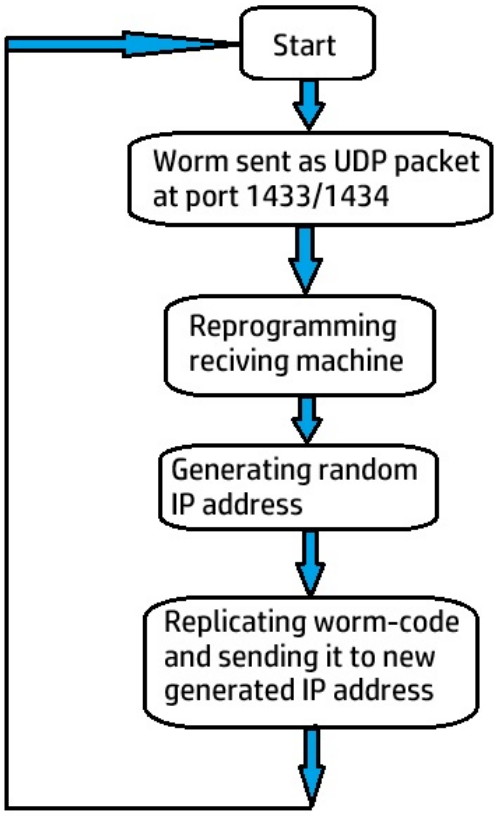

Figure 2.1: Flow Diagram of Slammers Spread Mechanism (Bajaj & Guha Roy, 2004)

Epidemic models have been valuable tools for understanding the spread of network worm over time. Märtens et al. (2016) extended the classical Susceptible-Infected-Susceptible SIS model to account for changes in the effectiveness of worm removal over extended periods, resulting in a time depended SIS model. Their work on Conficker worm revealed that reinfection rates and variations in countermeasures could explain the persistence of worm activity over long durations. Although this modelling approach looks exciting, and the goal of the authors was somewhat similar to ours, this research will

focus purely on empirical analysis of the slammer worms traffic, movement, and evolution, rather than applying theoretical models like SIS.

2.5 Packet Analysis and Tools

Analysing network traffic is is essential for observing worm activities and understanding their behaviour. Packet analysis tool such as Wireshark and Tcpdump are fundamental in this domain. Fuentes and Kar (2005) conducted a comparative study on these tools, highlighting their effectiveness in educational settings for teaching network protocols and packet structures. Ethereal or as it now known as Wireshark provides a GUI graphical user interface that allows for detailed inspection of packets content, making it accusable for users who may not be comfortable using CLI command line interface. Tcpdump on the other hand, is a command line tool that captures packets and display them with less processing overhead, suitable for quick diagnostic and scripting purposes. Authors concluded that while both Wireshark and Tcpdump are both excellent tools for network analysis and education, they each have their strength and limitations. Fuentes and Kar (2005) also suggested that the use of packet sniffers like Wireshark should be restricted to closed networking environments rather than public or campus wide network. This is due to the potential exposure of sensitive information, such as passwords or private communications, which poses ethical and legal concerns.

Building on that earlier work, Pradeepini et al. (2018) introduced a hybrid PCAP analyser that integrates Tshark (the command line variant of Wireshark) to automate the extraction of relevant information from large pcap files. This hybrid analyser incorporates features for customized filtering and reporting, significantly reducing both the detection and analysis time for infections such as SQL Slammer. The authors also demonstrated that deploying the analyser on a single system can facilitate the processing of packet data from multiple machines simultaneously, highlighting the efficiency gains in when processing large datasets.

Virtualized environments offer a safe and controlled setting for malware analysis, mitigating the risk associated with studying malicious code on live network. Ahmad (2019) introduced the V-network, a stable and effective virtualized testbed that creates isolated network environments where worms like SQL Slammer can be safely deployed and observed without endangering real world systems. By simulating network conditions and configurations, the V-network enables researchers to gain insight into spread of worms across different topologies, identifying key characteristics such as scanning patterns, infection rates, and exploited vulnerabilities.

Packet analysis plays an important role in network forensics, providing the means to reconstruct events and gather evidence of malicious activities. Sikos (2020) conducted a comprehensive examination on packet analysis for network forensics, highlighting its importance in both investigative and preventive context. The author noted that packet data can reveal detailed information about network based attacks, including the source and nature of the threat, the method used to infiltrate systems, and the extent of the compromise. Additionally, Sikos (2020) addressed the limitations and challenges of relying solely on packet data, highlighting its complementarity to other forms of evidence, such as firewall logs or CCTV footage. The examination also included a comparison of state of the art packet analysers, assessing their capabilities from various viewpoints.

2.6 Intrusion Detection Systems and Observations

Although a long time has passed since the SQL Slammer outbreak, and it was eventually contained through port blocking and patch deployment, its unseen speed led to different research on early warning systems and preventive measures.

Intrusion Detection systems IDS are used to identify worm infections by analysing network traffic and identifying suspicious patterns. Chen et al. (2014) highlighted the importance of intrusion detection in identifying worms like Slammer, which can evade traditional defences by exploiting unpatched vulnerabilities. Their study highlighted that IDS systems are essential for monitoring network activities and identifying unusual traffic.

Li et al. (2020) explored the application of advance machine learning methods, such as recurrent neural networks RNNs, for detecting worm activity. Their research demonstrated that RNNs could successfully identify patterns of worm activity based on historical network data, which could be used to track ongoing infections. However, their approach relies heavily on the availability of the extensive training datasets, which may limit its general applicability.

Hughes and Qu (2012) proposed using logistic regression models for malware signature detection, suggesting that this approach could significantly improve detection rates compared to older methods. Their study concluded that logistic regression is especially effective when distinct and sufficient characteristics for each malware family are identified. However, they also acknowledged the challenges in applying the model across different types of malware.

2.7 Summary

While network telescopes have significantly contributed to understanding IBR and worm spread, there is a gap in detailed analysing of the SQL Slammer worms activity over an extended period, particularly from 2005 to 2024. Previous studies have focused more on its initial outbreak and sporadic re-emergence but have not provided a detailed analysis of its evolution over time.

The SQL Slammer worm had a major impact on network security, revealing vulnerabilities and causing significant disruptions. Existing literature mainly focuses on its initial outbreak and sporadic re-emergence, leaving a gap in understanding its long-term activity. This project aims to fill that gap by using network telescope data filtered to UDP ports 1433 and 1434 to assess changes in its payload, identify other attacks targeting these ports, and analyse its geographical distribution.

The next chapter (Methodology) details how data will be gathered and processed: it will explain how the project will filter UDP traffic on mentioned two ports, check for the Slammer signature in packet payloads, and map source IP addresses to their countries of origin. This approach builds on established practices from previous studies, but extends them to explore the worms long term activity. By doing so, the research seeks to fill an important gap and determine whether SQL Slammer remains a threat, and in what ways, under current network security conditions.

3

Methodology

3.1 Methodology Introduction

This chapter presents the methods and techniques used to study the re-emergence and long-term persistence of the SQL Slammer worm using network telescope data. Building upon previously work, primarily the study on the reappearance of the Slammer worm. This research extends existing knowledge by examining the worms sustained presence from 2005 to 2024, a significantly longer observational period compared to previous studies. The methodology builds upon established best practices and techniques for detailed empirical analysis and is designed to support a comprehensive investigation into changes over time, payload characteristics, geolocation of malicious sources, and possible detection of other potential threats on UDP ports 1433 and 1434.

In term of structure, this chapter is organized as follows. First, the overall research design and approach are described (section 3.2). Next the data collection process is outlined (section 3.3), detailing the tools and procedures used for packet inspection, analysis, and geolocation (section 3.4). Then, the methods for data preparation are described in detail, including standardizing file organization and verifying ports and filtering out irrelevant traffic (section 3.5). Finally, the detailed analytical methods that applied in this research explained (section 3.6), the study limitations are listed (section 3.7), and then chapter concludes with a summary of the methodological framework (section 3.8).

3.2 Research Design

An observational and analytical approach was adopted to study the SQL Slammer worms activity from 2005 to 2024. Similar to Chindipha and Irwin (2017) research the process involve analytical

techniques, such as time-series analysis, payload examination, and geolocation mapping, was used to understand the traffic targeting UDP ports 1433 and 1434. The primary goal of this research was to determine whether the SQL Slammer worm still poses a threat in present-day networks, assess changes in its behaviour over time, and evaluate its geographical distribution alongside identifying any additional malicious activities on these ports.

3.3 Data Collection

The dataset used for this research consists of Internet Background Radiation (IBR) traffic captured by a Network Telescope operating on the SANREN network, managed by The National Integrated Cyber Infrastructure System (NICIS). Provided specifically for research purposes by Professor Barry Irwin, the dataset contains Packet Capture (PCAP) files covering the period from August 3, 2005, to October 31, 2024. The network telescope acts as a passive sensor that monitors unused IPv4 address spaces for anomalous traffic, capturing background noise that includes worm propagation attempts and automated scanning (Stanifor et al., 2004; Wustrow et al., 2010).

The primary focus was analysing UDP packets sent to port 1433 and 1434, which associated with Microsoft SQL Server and are the targets exploited by the SQL Slammer worm. Although the dataset was initially pre-filtered for these two ports, further examination revealed a large amount of irrelevant traffic. This called for an additional filtering process (described in detail later) to isolate traffic indicative of Slammer worm activity.

3.4 Tools and Procedures

A range of tools and procedures were used to process and analyse the large dataset effectively. The following subsections describe both the packet-level and geolocation tools along with the data processing tools used in this research.

3.4.1 Packet inspection and Analysis Tools

The analysis in this research relied solely on pre-captured PCAP files provided from network telescope datasets. The following tools were utilized for packet filtering, analysis and inspection of the captured traffic:

- **Wireshark**: This graphical network protocol analyser that provides comprehensive tools for visualizing packet data, reconstructing sessions, and decoding protocols. Although it can be slower when handling large datasets compared to command-line tools, it allows for interactive exploration of packet captures (Combs & Contributors, 2023; Pradeepini et al., 2018). In this research, Wireshark was employed for detailed payload analysis and for manually verifying the presence of the specific 376-byte payload associated with SQL Slammer. Its interactive interface enabled the extraction of raw payload files for further analysis and cross-verification with VirusTotal.

- **Tshark**: The command-line version of Wireshark, extensively was used to extracting detailed fields such as frame lengths, timestamps, and IP addresses, and to converting filtered results

into CSV format for further analysis without the graphical drawbacks of Wireshark (Pradeepini et al., 2018). Surprisingly, Tshark became the primary analysis tool for this research, despite the initial assumption that Tcpdump would be more effective due to its speed and simplicity.

- **Tcpdump**: Initially used for filtering of unnecessary information from PCAP files, Ttcpdump focuses specifically on relevant UDP traffic, timestamps, and packet sizes. Its lightweight nature allows rapid extraction and processing of large datasets, making it suitable for initial analyses and rapid traffic filtering (Fuentes & Kar, 2005). However, after initial comparisons by using both Tshark and Tcpdump together for the same tasks during the analysis, Tshark was found to offer more detailed and flexible results needed for this research.

3.4.2 Geolocation Tools

To determine the physical origin of source IP addresses within the dataset, geolocation analyses was preformed mainly with a help of tool named: **Netcat** (`nc`). Netcat was used to query WHOIS database provided by whois.cymru.com, which returns details on Autonomous System Numbers (ASNs) and associated organizations (Cymru Team, 2023; Irwin et al., 2007). These techniques made it possible to pinpoint major hotspots of Slammer worm activity and observe changes in malicious traffic over the study period.

However, there is an noteworthy limitation in Internet Background Radiation (IBR) data, particularly with UDP based threats like Slammer. Because UDP does not require a three-way handshake to confirm a source address, spoofing is comparatively easier (Irwin, 2013). Based on that, some IP addresses observed in the telescope logs could falsely represent their true geographic or organizational origin. This limitation applies regardless of which geolocation tool is used, since the initial data itself may contain false IP source information.

3.4.3 Data Processing Software

For further analysis, data extracted from PCAP files were converted to CSV format using python and Bash scripts. Libraries such as **Pandas** and **Matplotlib** were used for data manipulation, visualization, and statistical assessment. This approach allowed to generate summary tables in both plain text and LaTeX formats (in some examples), that helped more efficiently to manage 19 years of data and format it directly into the final report.

3.5 Data Preparation and File Organization

The provided PCAP files were initially structured inconsistently. Some years were provided as single annual files, while others were already split into monthly files with uneven boundaries. In some cases, captures began on day two or even included packets from the first day of the following month. To standardize the dataset the tools **mergecap** and **editcap** were used for merging and then splitting of the files to create uniform monthly captures, that starts at the first second of the first day of the month (2021-01-01 00:00:00) and ends at the last second of the last day of the month (2021-01-31 23:59:59).

```
mergecap -w 2021-total.cap 2021-*.cap
```

```
editcap -A "2021-01-01 00:00:00" -B "2021-01-31 23:59:59"
\ 2021-total.cap 2021-01-port1433-1434.cap
```

Since there were 19 years of data, and most of them required splitting into twelve months and reformat each month, the manual workload quickly became unmanageable. Eventually, a Bash script was developed to automate this splitting process and ensure efficiency throughout the standardization phase. The script accepted two command-line arguments: the path to the original annual PCAP file and the number of the year being processed. The script checks for leap years to assign the correct number of days to February. Then, for each month, it calls `editcap` to extract the needed range mentioned above. All resulting files were named in a consistent format such as YYYY-MM-port1433-1434.cap, which simplified later analysis and visualization processes.

After the splitting process was complete a new tool was created to quickly review the generated files, which became over 200 monthly PCAP files by that time. A new python was developed, script automatically extracted and formates information from the resulting monthly PCAPs. This new script used **capinfos** to extract the start and end timestamps as well as the file size for each file. It then used `tshark` and `wc -l` to count the number of packets per file. All the output information was assembled into a structured summary table in both plain text and LaTeX formats into an output text file. This dual output approach was chosen to support both internal validation during the research process and be able to include results into the final report if necessary.

3.5.1 Verifying Ports and Filtering Irrelevant Traffic

Although the dataset provided was initially pre-filtered for UDP ports 1433 and 1434, there were still irrelevant traffic targeting numerous other ports which was irrelevant for this research. To quantify and address this issue, a detailed verification was carried out using the following command:

```
tcpdump -nn -r 2021-total.cap | awk -F '.' '{print $10}' | awk -F ':' '{print $1}' |
sort | uniq -c | sort -r
```

This result partly illustrated in the (Table 4.1) indicated that although ports 1433 and 1434 accounted for the majority of traffic, there were still thousands of packets destined for other ports. In total, 2,666 different irrelevant ports and well over 100,000 packets not destined for 1433 or 1434 and this result was only from year 2020. Because of this, an additional filtering using `tcpdump` was applied to extract only packets specifically directed to UDP destination ports 1433 and 1434:

```
tcpdump -r 2021-total.cap "dst port 1433 or dst port 1434"
-w 2021-total_filtered_ports_1433-1434.cap
```

Eventually, a new python script was developed to filter the yearly merged PCAP files for ports 1434 and 1433. Monthly splitting and summary table generation were integrated into the same script, as this approach was faster than filtering over 200 individual monthly files. This ensured that further analysis focused only on relevant traffic. This script can be found in the artefact appendix under point 3 in the README file, named `filter_split_review.py`. Additionally, the results of this analysis are illustrated in (Section 4.2).

3.6 Data Analysis Techniques

The analytical phase of this project integrates several approaches: payload validation, time series analysis, geolocation mapping, source IP evaluation, and detection of additional threats to thoroughly investigate SQL Slammers persistence and evolution, thereby addressing the primary research objectives. These methods correspond directly to the techniques summarized in the literature (Chindipha & Irwin, 2017; Irwin, 2013; Stanifor et al., 2004).

3.6.1 Payload Analysis

This section describes methods for analysing networks packet payloads from UDP traffic on ports 1433 and 1434. In order to confirm that these packets are associated with the SQL Slammer worm, three different methods of payload analysis were used:

1. Analysis by payload size,
2. Analysis by payload hash, and
3. Analysis by payload checksum.

Each method provided an independent way to verify that the observed packets match the characteristics of Slammer worm traffic. In this research, diferent results was compared from `tcpdump` and `tshark` commands. Although `tcpdump` reports a payload length of 376 bytes and `tshark` a frame length of 418 bytes, this difference is due to the inclusion of additional header information in the `tshark` output. Both methods describe the same packets.

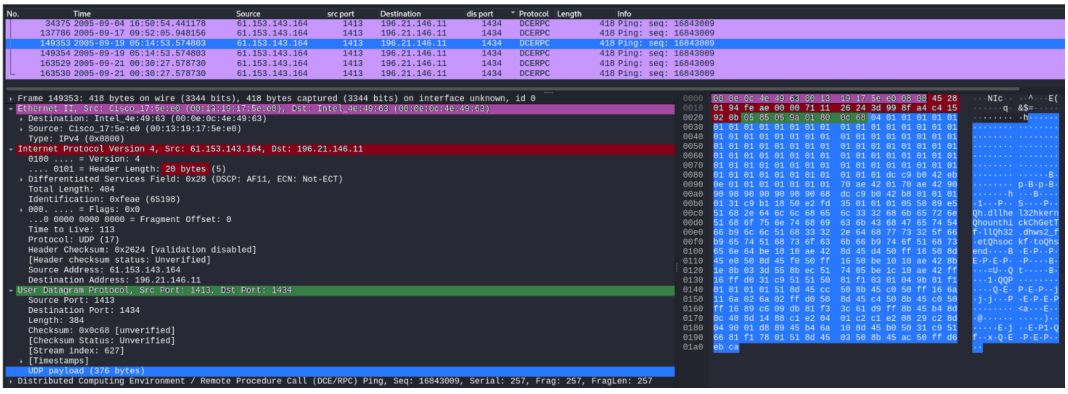

Figure 3.1: Example of how 418 bytes are divided

In Figure 3.1 (the color-coded Wireshark capture), it is clearly highlighted that each worm packet totals 418 bytes, divided across several protocol layers. The Ethernet header is 14 bytes, the IP header is 20 bytes, and the UDP header is 8 bytes, adding up to 42 bytes on top of the known 376-byte payload of the SQL Slammer worm. Because the worms core payload was known to be 376 bytes, filtering for a total of 418 bytes on the wire allows accurate isolation of Slammer traffic, treating the rest as non-Slammer traffic.

As shown in the hexdump on the right in Figure 3.1 and Figure 3.2 bellow, the payload begins with these specific bytes **04 01 01 ...** and ends with **... d6 eb ca**, which have been documented in previous

analyses of the Slammer worm (Chindipha & Irwin, 2017). By highlighting each header in different colours, it becomes clearer where the Ethernet, IP, and UDP layers end, and where the malicious payload begins.

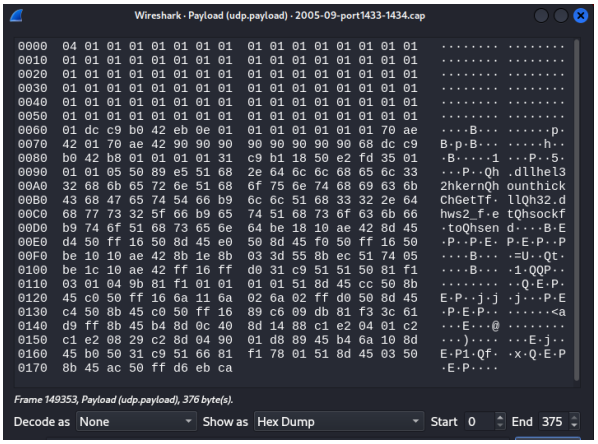

Figure 3.2: Slammer hexdump of 376 bytes

Figure 3.2 further illustrates the worms characteristic once structure separated from the headers. The payload was then extracted in raw format on a virtual machine, producing a local copy of the Slammer worm file that was then cross-checked with VirusTotal (Figure 3.3). This procedure provided the known Slammer hashes, which later used in analysis of payloads by hashes as well further confirmed that the captured data indeed represents a SQL Slammer worm payload Table 3.1.

Hash Type	Hash Value
MD5	a0aa4a74b70cbca5a03960df1a3dc878
SHA-1	e496315d09f0b48fddedd8e25a4e56339339a5bd
SHA-256	4f22864414f474843eb1e4599185e31d51c5b4aefde63b4dc5a850a39aeff3cb

Table 3.1: Hash Value of SQL Slammer worm from Virus-total

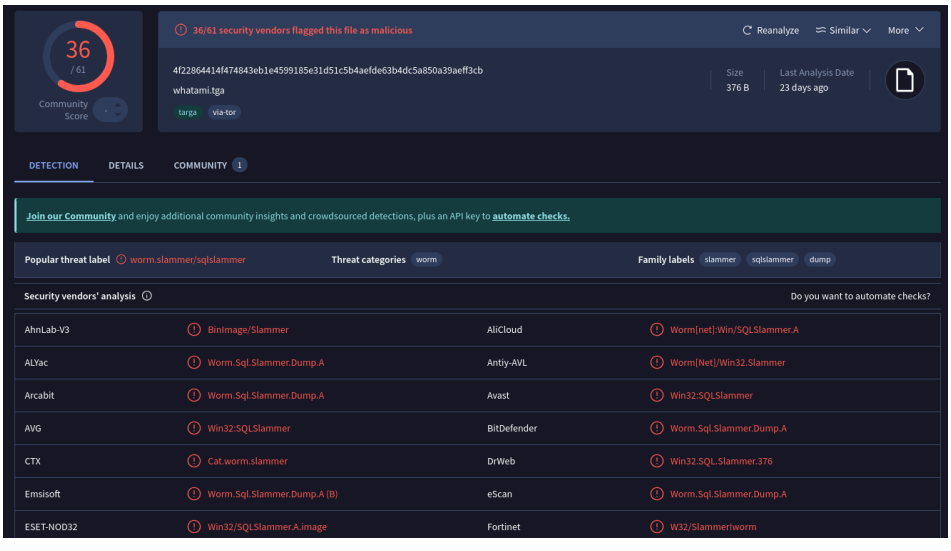

Figure 3.3: Result of uploading Extracted payload to VirusTotal

3.6.2 Analysis by Size

The three validation techniques were implemented as follows. For the size-based analysis, tcpdump was configured to filter frames where the UDP payload length field matched the value 376 bytes. Meanwhile, tshark was set to filter frames with a total frame length of 418 bytes. The exact commands used for this process are provided here under:

```
tcpdump -nr 2005-08-port1433-1434.cap -q 'udp' 2>/dev/null | awk '{print $8}' |
sort | uniq -c | sort -r
```

```
tshark -r 2005-08-port1433-1434.cap -Y "frame.len == 418" 2>/dev/null | wc -l
```

A brief console output examples is shown in figures Figure 3.4 and Figure 3.5, intended just to demonstrate correct syntax and different methods rather than to present numerical results.

Figure 3.4: Example of tcpdump output.

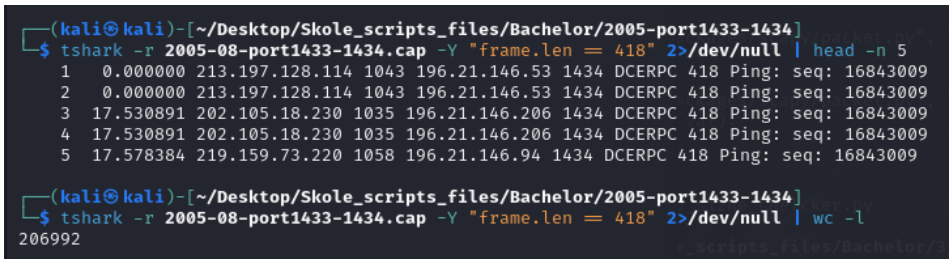

Figure 3.5: Example of tshark output.

3.6.3 Analysis by Hash and Checksum

The analysis based on hash value of payloads was carried out using a developed python script which implemented **scapy** and **hashlib** libraries. This script processed each UDP segment individually, retrieving the raw bytes and generating an MD5 hash value via the hashlib library. Segments producing a MD5 hashes matching the known SQL Slammer hash value a0aa4a74b70cbca5a03960df1a3dc878. The results are then written to a CSV file that lists the packet index, the computed MD5 hash, and whether it matches the known Slammer hash (YES or NO).

A secondary script performed a similar operation but replaced the hash function with a 16-bit additive checksum, comparing the result against the fixed value 0x7654 which was pre-calculated on the confirmed Slammer payload. The result was also written to an CSV file for further analysis. Figure 3.6 and Figure 3.7 present shortened, single-screen outputs from the execution of these scripts. The purpose of these figures is solely to illustrate the script execution and output format.

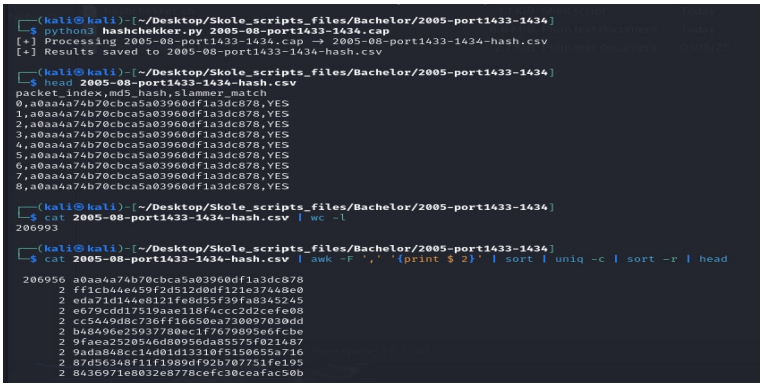

Figure 3.6: Example of hash output.

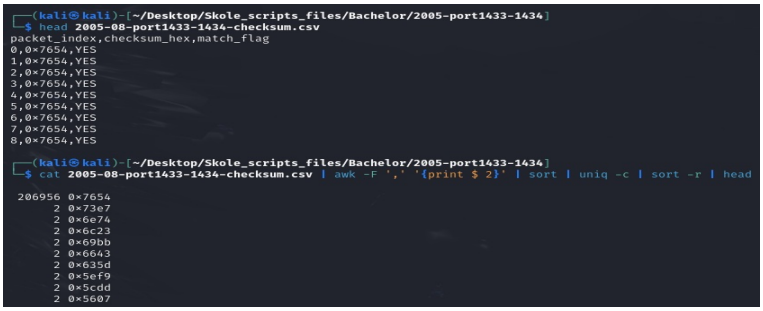

Figure 3.7: Example of checksum output.

Between the two methods the MD5 hash value and the checksum, the checksum script was computationally more efficient. Because MD5 hashing, is a cryptographic function that produces a fixed 128 bit hash. It is more CPU intensive because it performs multiple rounds of bitwise operations, shifts, and modular calculations to ensure resistance to collisions. On other hand, the checksum calculation in the is a much simpler operation. It just adds the byte values in the payload and reduces the result to 16 bits using (`sum(payload) & 0xFFFF`), which makes it significantly faster. However, the hash-based analysis offers clearer visual validation and allows for comparison with known signatures in the literature and services like VirusTotal. Both of these scripts can be found in the artefact appendix under point 4 in the README file, named `hashcheker.py` and `cheksum_finder.py`. The results of these analysis methods are presented in (Sections 4.3.1 - 4.3.2)

3.6.4 Detection of Other Potential Threats

In addition to analyse Slammer worm traffic, the study also investigated whether other malicious payloads were present on UDP ports 1433 and 1434. A dedicated python script extracted raw UDP payload bytes from filtered dataset and computed MD5 hashes and size of the payload. It was the same script we used in hash analysis faze only slightly modified to extract the size of the payload in addition to the hash value. Additional size filtering was applied manually outside the script to optimize analysis efficiency. Although SQL Slammers 376 byte payload remains the smallest known UDP worm in observed attacks (Chen & Robert, 2004; Moore et al., 2003; Stanifor et al., 2004), a conservative lower bound of 90 bytes was selected for payload inspection. The large part of payload was filtered out but the potential to identify previously undocumented, ultra-small worms or attack techniques was still preserved. The output of the script was printed out in the command-line.

Payload hashes were then extracted directly from command-line output and manually copied into a python script as a hard-coded list. The script then loops over each hash, issues an HTTP GET request to the VirusTotal API using the required API key header, and reads the response JSON data to extract fields such as the file name, file size, reputation score, and last analysis statistics like: counts of malicious, suspicious, undetected, and harmless detections. A short delay is inserted between requests to avoid triggering the APIs rate limits. Error handling is also implemented to print out the `response.status_code`. The results of this analysis are presented in (Section 4.3.2). Both of these script mentioned in this section can be found in the artefact appendix under point 11 and 12 in the README file, named `hashes/hash_and_sizhe_chelcker.py` and `virustotal_hashchecker.py`.

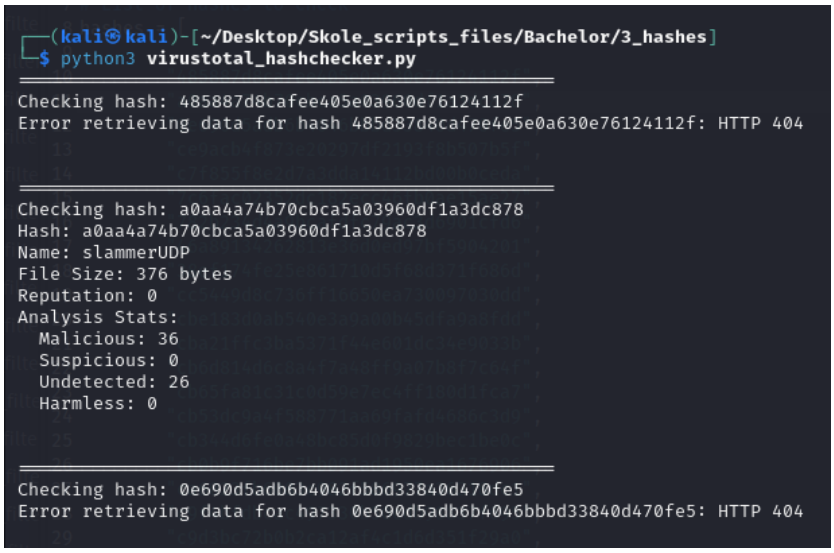

Figure 3.8: Example of the output of VirusTotal script used in the analysis.

3.6.5 Extraction of Packet Counts

A Bash script was created to quantify the monthly volume of Slammer packets across the entire dataset of PCAP files. The script iterates over every capture, invokes *tshark* to count frames whose on-wire length is exactly 418 bytes (the Slammer signature established earlier), extracts the year and month from the filename, and prints a comma-separated line that can be redirected into a CSV file:

```
for pcap in 20**-*-port1433-1434.cap; do
    [ -e "$pcap" ] || continue
    year=$(echo "$pcap"   | cut -d'-' -f1)
    month_num=$(echo "$pcap" | cut -d'-' -f2)
    month_name=$(date -d "2000-$month_num-01" +%B)
    packet_count=$(tshark -r "$pcap" -Y "frame.len == 418" 2>/dev/null | wc -l)
    echo "$year,$month_name,$packet_count"
done
```

In a later, updated version of the script, it accepts one or more directory arguments so that whole year-folders could be processed in a single call. For more details, both versions of this script can be found in the artefact appendix under point 5 in the README file, named `count_376_packets.sh` and

`count_376_packets2.sh`. The resulting CSV file (one line per month) is then easily implemented into an python script that uses **pandas** and **matplotlib** to create time-series visualisations; these graphics are presented in Chapter 4.

3.6.6 Time Series and Volume Analysis

Following confirmation of SQL Slammer worm presence in the captured traffic, the analysis continued with examining the frequency of packets over time. This was done to understand the worms activity pattern and to identify any trends, seasonal fluctuations, or unusual spikes in traffic.

The dataset was processed by extracting timestamps from each packet and aggregating them into daily and monthly counts. A python script was used to automate this process, leveraging `tshark` to extract timestamps from packets matching the 418-byte Slammer signature. The extracted timestamps were then formatted into CSV files for further analysis in python.

Before automating the timestamp extraction process, several commands were tested to determine the most effective approach for the script. Two commands emerged as the most suitable:

```
tshark -r ./2005-port1433-1434/2005-total_filtered_ports_1433-1434 -Y "frame.len == 418"
-T fields -e frame.time -e ip.src -e udp.dstport -E header=y -E separator=,
-E quote=d 2>/dev/null > timestamp_slammer_2005_data.csv
```

```
tshark -r ./2005-port1433-1434/2005-total_filtered_ports_1433-1434 -Y "frame.len == 418"
-T fields -e frame.time_epoch -e ip.src -e udp.dstport -E header=y -E separator=,
-E quote=d 2>/dev/null | awk -F, 'NR==1 {print $0; next} {cmd="date -d @"$1"
+\"%Y-%m-%d %H:%M:%S\""; cmd | getline timestamp; close(cmd); $1=timestamp; print $0}'
OFS=, > timestamp_slammer_2005_data.csv
```

The second command was ultimately chosen for the script due to its ability to provide a more detailed and adjustable time format, which proved to be more suitable for the analysis (Figure 3.9).

```
┌──(kali㉿kali)-[~/Desktop/Skole_scripts_files/Bachelor/1_timestamps]
└─$ tshark -r ../2005-port1433-1434/2005-total_filtered_ports_1433-1434 -Y "frame.len == 418" -T f
ields -e frame.time_epoch -e ip.src -e udp.dstport -E header=y -E separator=, -E quote=d 2>/dev/nu
ll | awk -F, 'NR==1 {print $0; next} {cmd="date -d @"$1" +\"%Y-%m-%d %H:%M:%S\""; cmd | getline ti
mestamp; close(cmd); $1=timestamp; print $0}' OFS=, | head
frame.time_epoch,ip.src,udp.dstport
2005-08-03 09:52:53,"213.197.128.114","1434"
2005-08-03 09:52:53,"213.197.128.114","1434"
2005-08-03 09:53:10,"202.105.18.230","1434"
2005-08-03 09:53:10,"202.105.18.230","1434"
2005-08-03 09:53:10,"219.159.73.220","1434"
2005-08-03 09:53:10,"219.159.73.220","1434"
2005-08-03 09:53:12,"218.95.64.15","1434"
2005-08-03 09:53:12,"218.95.64.15","1434"
2005-08-03 09:53:18,"61.143.101.100","1434"
```

Figure 3.9: Time format chosen for extraction

The resulting CSV files, which were generated for each month across the study period, were then used to create yearly and monthly packet count visualisations using **Matplotlib** in python. These visualizations revealed trends such as spikes in traffic, changes over time, and long-term reduction in Slammer activity. Detailed statistics and time series graphs was created and provided quantitative insights into the worms evolution over time.

3.6.7 Geolocation and Source Analysis

Geographical analysis of Slammer traffic was performed using the previously generated CSV files from the time series section. To begin, all unique IP addresses for each year were extracted into separate files, skipping headers and removing unwanted characters such as double quotes. The resulting IP lists were formatted specifically for querying the WHOIS database provided by whois.cymru.com. Each IP file was prepared by inserting the line 'begin' at the start, 'verbose' as the second line, and concluded with an 'end' line to correctly format the request. The file was then submitted to the Cymru WHOIS database using the following command:

```
cat "uniq_ips_2005.query" | nc whois.cymru.com 43 > "enrich_2005.cymru"
```

To efficiently manage this repetitive task while going through almost 20 years of data, a Bash script was created to automate the IP extraction and WHOIS query generation process across all years. The results from individual queries (2005–2024) were merged into a single comprehensive file named 'Cymru', which was later used for analysis. A sample of the merged Cymru file is shown in (Figure 3.10), excluding the first few lines that contained headers or incomplete data entries.

```
┌──(kali㉿kali)-[~/Desktop/Skole_scripts_files/Bachelor/2_geolocation]
└─$ tail -n +9 1_merged.cymru | head
7018    | 12.104.70.171    | 12.0.0.0/9          | US | arin     | 1983-08-23 | ATT-INTERNET4, US
7018    | 12.104.70.72     | 12.0.0.0/9          | US | arin     | 1983-08-23 | ATT-INTERNET4, US
7018    | 12.105.226.252   | 12.0.0.0/9          | US | arin     | 1983-08-23 | ATT-INTERNET4, US
7018    | 12.105.236.121   | 12.0.0.0/9          | US | arin     | 1983-08-23 | ATT-INTERNET4, US
7018    | 12.108.173.21    | 12.0.0.0/9          | US | arin     | 1983-08-23 | ATT-INTERNET4, US
7018    | 12.108.61.184    | 12.0.0.0/9          | US | arin     | 1983-08-23 | ATT-INTERNET4, US
7018    | 12.110.5.5       | 12.0.0.0/9          | US | arin     | 1983-08-23 | ATT-INTERNET4, US
7018    | 12.13.62.4       | 12.0.0.0/9          | US | arin     | 1983-08-23 | ATT-INTERNET4, US
7018    | 12.14.48.124     | 12.0.0.0/9          | US | arin     | 1983-08-23 | ATT-INTERNET4, US
7018    | 12.144.195.117   | 12.128.0.0/9        | US | arin     | 1983-08-23 | ATT-INTERNET4, US
```

Figure 3.10: Example of Cymru file

Earlier analyses of the Slammer worm have often relied on counting unique IP addresses or network blocks as a proxy for measuring the worms spread (Chindipha & Irwin, 2017). This study follows a similar approach, while acknowledging that IP addresses may not represent individual devices due to NAT, dynamic leasing, or spoofing. Still, patterns in which addresses repeatedly send packets to Slammer-targeted ports can provide meaningful insight. For instance, a high proportion of recurring addresses might indicate chronically infected networks or unpatched machines. Meanwhile, observing rapidly shifting source IPs may indicate temporary outbreaks. Based on this reasoning, two types of visual tables were later generated: one summarising activity by country, the other by /16 network block.

Although combined dataset results initially provided valuable insights, there was a concern that the totals alone could obscure important temporal changes. Previous time series and packet count analyses had already indicated that Slammer activity reached its highest levels in the first few years before gradually decreasing. To better visualize these patterns, the data was examined year by year from 2005 through 2024.

The yearly analysis used the same method as the tatoal/ merged one, but applied it individually per year. For each years enriched WHOIS file, the number of unique IP addresses was counted. Same data as already mentioned, country based IP mappings were extracted using `awk` magic, and /16 netblocks were generated by combining the first two octets of each IP. The following shell pipeline was used to extract the wanted results:

```
tail -n +2 enrich_2005.cymru | wc -l

awk -F '|' '{print $4}' enrich_2005.cymru | sort | uniq -c | sort -rn |
awk '{print $2,"&",$1,"&", sprintf("%.2f", ($1/44136)*100)"\\%\\\\ \\hline"}' | head

awk -F '|' '{print $2, $4}' enrich_2005.cymru | grep -E '[0-9]+' |
awk '{split($1, x, "."); print x[1]"."x[2]".0.0/16", $2}' |
sort | uniq -c | sort -rn | awk '{print $3,"&",$2,"&",$1,"&",
sprintf("%.2f", ($1/44136)*100)"\\%\\\\ \\hline"}' | head
```

To further optimize and speed up the process, a python script was developed to automate the calculation of the results. The script was configured to output the results in a format compatible with Overleaf, including the full table structure with headings and endings. Although it remains unclear whether converting the process into a python script actually saved time, since handling LaTeX-specific characters like '\' proved to be challenging and required additional troubleshooting. Nevertheless, the final outcome was highly fulfilling. The results of this analysis are described in (Section 4.5), and the scripts along with usage details can be found in the artefact appendix under point 9 and 10 in the README file, named `geolocator.sh` and `time_saver.py`.

3.7 Limitations

Although the methodology employed in this study is robust and multifaceted, several limitations must be acknowledged:

3.7.1 Limited IP Coverage

The network telescope only monitors a small subset of the IPv4 address space. Consequently, the findings represent only a fraction of global SQL Slammer worm activity and might not capture all geographically or temporally limited outbreaks.

3.7.2 Temporal Data Gaps

Periodic equipment failures and network outages, such as significant data gaps observed in the year 2016, resulted in incomplete data collection. These gaps can compromise the accuracy of pattern analysis and may lead to underestimation or misrepresentation of the worms activity levels during those intervals (Irwin, 2013).

3.7.3 Geolocation Accuracy

Although Travis et al. (2003) stated in his research that the SQL Slammer worm itself did not spoofed its source address and always used the real IP of the infected host. The Network telescope data naturally contains a lot of address-spoofed traffic (e.g. misconfiguration, reflected DDoS) as part of the "Internet Background Radiation" (Krause, 2021). As a result, some addresses in the telescope logs may not represent the true origin of the traffic. Moreover, geo-based identification, and ASNs may also show incorrect ownership or regional locations over time. With this in mind, queries to whois.cymru.com generated more historical mapping rather than guaranteed accurate attribution data.

3.8 Summary

The methodology detailed in this chapter builds upon established practices in network security and worm activity research, integrating packet inspection, payload analysis (by size, hash, and checksum), time series analysis, geolocation mapping, and additional threat detection techniques. Custom scripts for automating data standardization, extraction, and analysis ensured reproducibility and rigorous quantitative assessment. Despite limitations related to data completeness, geolocation accuracy, and temporal gaps, the comprehensive approach provides robust insights into the persistence and evolution of SQL Slammer worm activity from 2005 to 2024

The strengths of this methodology lie in its empirical approach, applying real world data collected over an extended period to provide insights into worms activity. The use of multiple tools and cross-referencing of data sources contributes to the robustness of the findings. Additional, drawing on the methods used by Chindipha and Irwin (2017) and other researches provides a solid foundation for the analysis, ensuring consistency with previous research while also extending the scope to cover more recent years.

The following chapter will present the outcome that will be produced from these methods, explaining in detail how the collected data was processed, analysed, and aligned with the research goals. This discussion will highlight both the strengths and weakness of the chosen approach, offering further clarity on SQL Slammer present day importance and any newly discovered attack patterns.

4

Results and Analysis

4.1 Analysis Introduction

This chapter presents the process and results obtained from analysing network traffic associated with the SQL Slammer worm. It includes findings on the identification of Slammer activity, verification of payload characteristics, and visualization illustrating traffic patterns. Result related to payload analysis by size, hash, and checksum are also described. Additionally, geographic patterns in Slammer traffic and investigations into other potential threats are reported. All analyses in this chapter build directly on the methodologies previously described, ensuring clarity and structure in the presentation of research findings.

4.2 Verifying Ports and Filtering Out Irrelevant Traffic

Initial inspection showed that many packets in the telescope captures were destined for SQL Servers well-known ports 1433 and 1434. In the 2021 dataset alone, more than 6,736 distinct destination ports were observed. Table 4.1 illustrates the five most common ports that year; which resulted in over five hundred thousand packets that was targeting other services.

Dst. Port Number	Packet Count	Percentage
1433	7,113,218	82.81%
1434	953,682	11.10%
8081	39,732	0.46%
5060	39,305	0.46%
6881	39,277	0.46%

Table 4.1: Top 5 Ports 2021

To remove this extraneous traffic, every annual capture was filtered so that only packets whose destination port was 1433 or 1434 remained for further analysis (Section 3.5.1). The effect of this step is summarised in Table 4.2. Across all years 1,7 million packets which was approximately 2.3 percent of the original dataset were eliminated, leaving 74.8 million packets for subsequent Slammer-specific analysis. The reduction was particularly noticeable from 2020 to 2024, suggesting either: a decline in SQL Server scanning activity during this period, or changes in data collection methods that may have filtered background traffic differently than in prior years.

Year	Packets	Filtered Packets	Removed Packets	Percentage Removed
2005	1,116,283	1,114,949	1,334	0.12%
2006	1,053,001	1,051,717	1,284	0.12%
2007	1,912,692	1,909,694	2,998	0.16%
2008	1,496,597	1,494,129	2,468	0.16%
2009	783,022	773,559	9,463	1.21%
2010	549,231	535,535	13,696	2.49%
2011	493,453	484,857	8,596	1.74%
2012	463,449	456,826	6,623	1.43%
2013	585,852	579,655	6,197	1.06%
2014	731,059	715,841	15,218	2.08%
2015	1,494,119	1,480,521	13,598	0.91%
2016	1,408,688	1,401,097	7,591	0.54%
2017	9,558,239	9,542,785	15,454	0.16%
2018	8,355,310	8,338,635	16,675	0.20%
2019	7,738,891	7,717,482	21,409	0.28%
2020	14,847,419	14,732,898	114,521	0.77%
2021	8,589,911	8,066,900	523,011	6.09%
2022	6,188,375	5,765,916	422,459	6.83%
2023	4,839,620	4,518,105	321,515	6.64%
2024	4,429,768	4,201,516	228,252	5.15%
Total	**76,634,979**	**74,882,625**	**1,752,354**	**2.28%**

Table 4.2: File Sizes Before and After Filtering (in Bytes)

The large reduction in the 2020–2024 captures, where more than 1.6 million packets were removed in these four years alone, highlights why port verification is essential before any Slammer-specific payload analysis. With non SQL traffic eliminated, the remaining packet stream now provides a clean foundation for the payload validation in the following section.

4.3 Payload Analysis

The analysis methodology established in Chapter 3 was applied to review filtered network telescope data, measuring the volume of SQL Slammer traffic detected over a nineteen year monitoring period.

Initial processing identified 74,8 million packets that met the port based filtering criteria. Every one of those packets was analysed with the three methods defined in (Section 3.6.1 - Section 3.6.3), which was: size, MD5 hash value and 16-bit checksum. In order to determine how much genuine SQL Slammer traffic had reached the telescope and whether any other malicious payload had ever appeared on the same ports.

4.3.1 Reconstructing the Slammer population

The length filter alone (`frame.len == 418`) reduced the data to 4,104,391 packets for the whole period of 19 years. Running the MD5 hash value script on the same dataset had a mismatch of only few hundred packets and produced a clean set of **4,104,018** packets where MD5 hash value matching known Slammer value`a0aa4a74b70cbca5a03960df1a3dc878`. The checksum method was implemented last with a same result as the hash value, additionally confirming that all slammer traffic i identified in the period 2005 - 2024.

The validation process was initially tested using data from August 2005, the earliest complete month in the dataset. In that month, both tshark and tcpdump identified 206,992 frames with a length of 418 bytes (376 byte payload). The MD5 and checksum tests retained 206,956 frames, rejecting the same thirty-six truncated packets marked as malformed by Virus-Total. Once the scripts consistently reproduced these results, the same validation pipeline was applied to each subsequent month, spanning from August 2005 to October 2024.

After all packets was filtered, sorted and analysed, an CSV file was created, containing every UDP payload over 19 years of data. The CSV file was containing a packet number, a hash value of payload and the size of the payload it self. Making it easy to visualize the result of this analyse by using little bit of 'awk' magic (Figure 4.1).

```
┌──(kali㉿kali)-[~/Desktop/Skole_scripts_files/Bachelor/3_hashes]
└─$ head -n 5 all_payloads_size_n_hash_2005-2024.csv
packet_number,hash_value,size
0,a0aa4a74b70cbca5a03960df1a3dc878,376
1,a0aa4a74b70cbca5a03960df1a3dc878,376
2,a0aa4a74b70cbca5a03960df1a3dc878,376
3,a0aa4a74b70cbca5a03960df1a3dc878,376

┌──(kali㉿kali)-[~/Desktop/Skole_scripts_files/Bachelor/3_hashes]
└─$ awk -F ',' '{print $3}' all_payloads_size_n_hash_2005-2024.csv | sort | uniq -c | sort -rn | head -n 5
4104391 376
3120040 18
 134767 25
  82045 24
  44022 29

┌──(kali㉿kali)-[~/Desktop/Skole_scripts_files/Bachelor/3_hashes]
└─$ awk -F ',' '{print $2}' all_payloads_size_n_hash_2005-2024.csv | sort | uniq -c | sort -rn | head -n 5
4104018 a0aa4a74b70cbca5a03960df1a3dc878
1641021 b7abbcf89138a883965046a3e76ca5c6
 100917 ff035bff2dcf972ee7dfd023455997ef
  97349 9417df9a4c3940a7d2412e7cd1b56978
  64155 b477b771a9d55f494d319b41f1bfebc1
```

Figure 4.1: The result of Slammer analyse by size and hash value.

4.3.2 Presence of other malicious payloads

To analyse whether any malicious payloads other than Slammer were present in the dataset, the same CSV file containing all hash values from all years was used. The results showed that the next most common payload length was only 18 bytes s illustrated in (Figure 4.1), and that a large number of small payloads were present, too small in size to represent malicious UDP payloads. Therefore, further hash analysis was performed only on payloads that were 90 bytes or larger, as described in Chapter 3. The filtered list of the most frequent payload sizes is shown in (Table 4.3), which highlights Slammers dominance at 376 bytes, but also reveals other larger payloads with the potential to be malicious.

Occurrences	Hash Values	Size in Bytes	% of Filtered traffic
4,104,018	a0aa4a74b70cbca5a03960df1a3dc878	376	5.48%
17,941	08071481e8f076cf0e68d58e0b3b8362	95	0.024%
280	485887d8cafee405e0a630e76124112f	411	0.00037%
273	0e690d5adb6b4046bbbd33840d470fe5	108	0.00036%
256	ce9acb4f873e20297df2193f8b507b5f	798	0.00034%
256	c7f855f8e2d7a3dda14112bd00b0ceda	798	0.00034%
256	7c6fac02352dc182ecc46fb0ae15ae27	798	0.00034%
256	782b234d4a9bcafdfc79ac3d6901cfd6	798	0.00034%
256	46a89134262813e36d0ed97bf5904201	798	0.00034%
256	38ef174fe25e861710d5f68d371f686d	1472	0.00034%
256	15f2c0638f1deb20c26c2ccbe875c228	798	0.00034%
255	e238a28f468ea0d51c33d50a96f46838	192	0.00034%

Table 4.3: Top occurrences of payload hashes. (2005-2024)

To learn whether any of those non-Slammer payloads were malicious, the 300 most frequent hashes were submitted to the VirusTotal API. Unfortunately, despite significant effort, time and analysis , the outcome was not as productive as initially anticipated. Out of the top 300 groupings of payload hashes analysed, only two hashes were identified in VirusTotal. One hash, “a0aa4a74b70cbca5a03960df1a3dc878”, which matched directly to the known Slammer worm signature, with a payload size of 376 bytes was flagged clearly as malicious. The other detected hash, “e238a28f468ea0d51c33d50a96f46838”, corresponded to a payload labelled `'ntp_123_monlist.pkt'` with a payload size of 192 bytes, which VirusTotal categorized as non-malicious.

Interestingly, the (Table 4.3) reveals a pattern of payloads with exactly 256 occurrences each, six of which share the same payload size of 798 bytes. This suggests that these packets may originate from an automated scanning tool or scripted probing activity, possibly testing for the same service or vulnerability across a hardcoded set of 256 target IP addresses. Since they share different hash values but identical lengths and counts, they could also represent crafted payloads used in reconnaissance. Another possibility is that these payloads are part of a botnet operation, where infected hosts are instructed to perform similar scans or probes in a synchronized manner. Although VirusTotal returned no known malicious labels for these samples, the highly structured repetitive pattern stands out from the rest of the non-Slammer data. Further more in-depth behavioural or content based analysis may be required to understand their purpose and determine whether they represent harmless probing or something more malicious.

4.3.3 Monthly Packet Counts

(Table 4.4) shows an an output of the CSV produced by the counting script described in detail in (Section 3.6.5), while (Figure 4.2) and (Figure 4.3) visualize the complete nineteen year presence of SQL Slammer in one merged chart.

The results of this analysis based on `'frame.len == 418'` confirm that the captured packets contain the SQL Slammer worm payload. The complete results of this multi-year analysis, partially illustrated in (Table 4.4), can be inspected in more detail in the resulting CSV file included in the artefact appendix. This CSV file can be found under the name (`all_time_slammer_376_port1433_1434.csv`). To better understand the worms presence at specific point in time, a closer analysis of when these

Year	Month	Packet_count	% of Filtered traffic
2005	August	206,992	72.70%
2005	September	196,424	81.12%
2005	October	174,722	78.83%
2005	November	185,978	78.73%
2005	December	105,759	81.23%
2006	January	68,750	72.19%
2006	February	58,091	58.02%
.....	.....	.....	.....
2011	October	3,887	13.63%
2011	November	7,784	13.45%
2011	December	9,997	22.76%
2012	January	13,792	25.89%
2012	February	8,141	19.94%
.....	.....	.....	.....
2016	November	3,151	1.21%
2016	December	21,885	6.14%
2017	January	13,410	4.80%
2017	February	11,166	4.02%
2017	March	13,810	7.75%
.....	.....	.....	.....
2024	September	0	0
2024	October	0	0

Table 4.4: Sample of monthly SQL Slammer packet counts

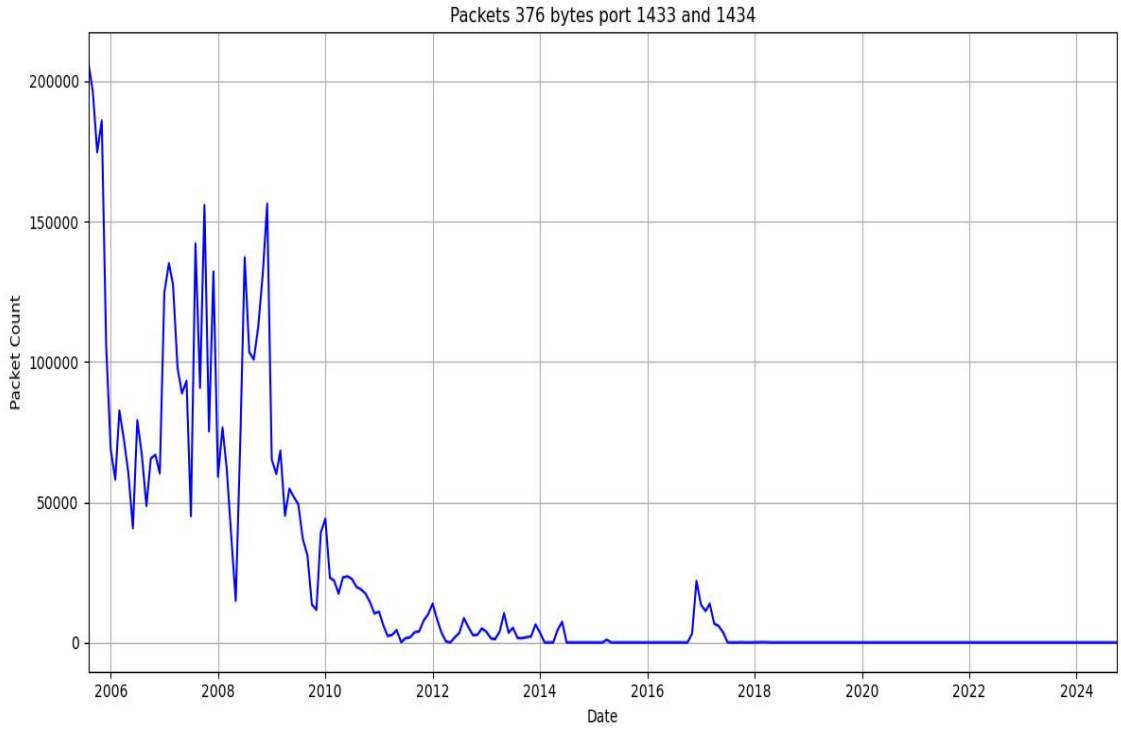

Figure 4.2: Slammer 376-byte packets 2005 - 2024 (Line Diagram)

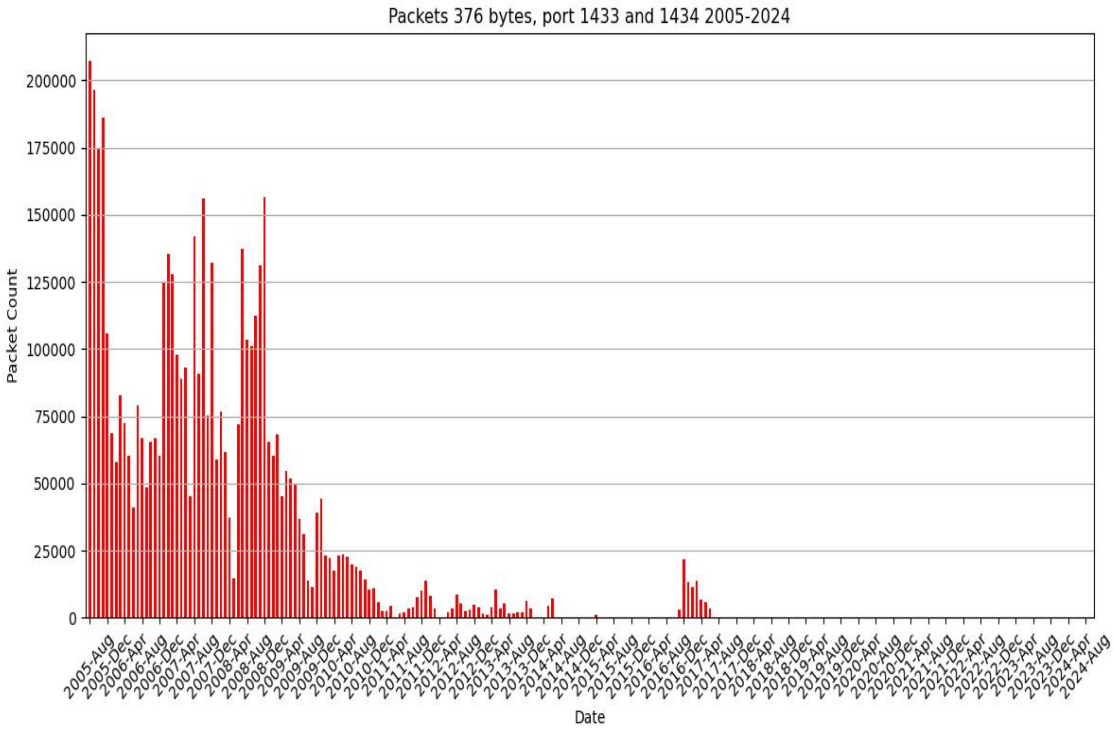

Figure 4.3: Slammer 376-byte packets 2005 - 2024 (Bar Chart)

packets were observed is needed. The next phase of this research investigates the temporal distribution of Slammer traffic, focusing on how activity changed over the years. Analysing traffic patterns over time, allows detection of peaks in infection rates, periods of inactivity, and possible explanations for changes. The following section presents the time series and volume analysis, which provides insight into the more detailed behaviour of Slammer worm traffic over the years.

4.4 Time Series and Volume Analysis

Once the the dataset was filtered trough with the help of a new timestamp-script described in more detail in (Section 3.6.6), the results were converted to CSV. The result was 4,1 million CSV lines, so not an easy nut to crack by human eye. So a time series visualizations were generated in form of an line chart and, bar chart with the help of `matplotlib` (Appendix A). These graphs showed the number of Slammer packets observed each day and month for each year. The results confirmed that Slammer traffic was highly active in the earlier years but significantly declined after 2008. While some years showed consistent activity, others contained major gaps, either due to data loss or actual reductions in Slammer worm traffic.

Notably, the data for 2016 was incomplete even before any filtering was applied. Large portions of traffic were missing, particularly between April 16 and May 25, as well as between July 11 and November 8. This means that even before applying any thresholds or exclusions, the dataset had gaps that may affect the analysis. Additionally, some of the original pcap files received were very small, meaning that even before filtering, there were cases where no Slammer traffic was detected at all. For example, data from 2011 to 2014 consisted of files only 50-60 MB in size per year, which

already suggested minimal UDP traffic presence.

4.4.1 Overview of changes in time series analysis:

The traffic pattern begins in August 2005 with an initial peak of 206,992 packets mark with Slammer pattern. Following this rise, activity gradually declines through the remainder of the year, with September recording 196,424 packets, October 174,722, November 185,978, and December 105,759. This downward trend suggests the initial outbreak was subsiding, though significant scanning activity remained.

In 2006, monthly totals varied between 40,859 packets in June and 82,774 in March, averaging approximately 64,200 packets per month, or around 2,100 per day. The lowest volumes occurred in June and September, which may indicate temporary measurement gaps or localized filtering rather than the complete disappearance of the worm.

The first quarter of 2007 saw sustained Slammer activity, with 124,600 packets in January and a peak of 135,116 in February. It continued this pattern before dropping to 44,974 packets in July, then rose again next month to regular levels of over 100,000 packets per month.

2008 began with moderate activity in January (59,045) and February (76,576), followed by a drop in April (37,371) and May (14,874). However, a rebound occurred in June (71,914), July (137,200), and December (156,326), marking the second-highest monthly volume for the year.

A gradual decline was observed in 2009, starting at 68,296 packets in March and falling to 31,058 by September. A sharper drop followed in October (13,539) and November (11,618), before a slight December spike to 39,180. Daily counts showed intermittent bursts, consistent with sporadic scanning waves or clean-up efforts.

In 2010, monthly traffic steadily decreased from 44,076 in January (including a daily spike of 5,234) to 10,304 in December. This may suggests improved patching and network filtering or extinction of vulnerable systems, which reduced Slammer scans by roughly half over the 7 years.

Traffic continued shrinking in 2011, with January recording 10,966 packets before dropping to the low thousands from April to October with a minimum 87 daily packets in June. Minor recoveries occurred in November (7,784) and December (9,997), with occasional daily spikes of 3,000–4,000 packets, possibly from a small clusters of unpatched systems.

Early 2012 saw moderate activity 13,792 packets in January, and 8,141 in February, followed by near-silence in April (346) and May (10). The remaining months had isolated bursts, such as 8,665 in August and 4,972 in December, possibly representing the leftover unpatched machines continuing to scan.

By 2013, monthly counts remained in the low thousands, ranging from 1,156 in March to 10,399 in May, reaching occasional just above 500 daily packets. The worms presence is still present, but far from its former values. There is more down time the spikes on yearly bases.

2014 saw almost no activity from February to April and July to November, except for a brief resurgence in May (4,475) and June (7,342). January recorded 3,477 packets, but by years end, traffic had nearly vanished.

In 2015, the only notable event was an April spike of 950 packets. March, May, and June each recorded fewer than ten packets, indicating near-total disappearance, with the last two packets observed on June 30th. This raises the question of whether the capture outages mentioned by the data owner had already begun in the third quarter of 2015.

No traffic was recorded from January to October 2016 due to known capture outages. This was mentioned by data owner in advance of the research. When monitoring was restored, November registered 3,151 packets, followed by a December spike to 21,885, a late peak that year.

From 2017 to 2018, activity dwindled further. In 2017, a few thousand packets appeared monthly and peaking around 13,800 in March, before dropping to almost nothing after June. In 2018, only a handful of packets were detected each month, maximum 104 packets in March, marking the final traces of scanning (Chindipha & Irwin, 2017).

No Slammer packets were detected in 2019 or 2020, suggesting that the worm may have finally died out, 16 years after its initial outbreak, due to the disappearance of vulnerable systems. A single packet appeared in May 2021. Whether this was an anomaly or a valid scan is unclear, but it underscores how sporadic residual activity can occasionally reappear even after long periods of inactivity (Kristoff, 2023).

From 2022 to 2024, no traffic from Slammer was found. At this stage, it seems that the worms activity has either ceased entirely or become too rare to be detected by the telescope data.

A more detailed and complete view of these patterns, both monthly and daily, can be explored through the raw CSV files, which are included in the artefact appendix. These files are named `all_time_slammer_376_port1433_1434.csv` and `1_timestamps/2005-2024_timestamp.csv`. Scripts used to generate these results are also documented in the appendix README file under points 5 and 7.

By examining the patterns shown in the bar and line charts in Appendix A, it became evident that Slammers presence in the network has faded over time. While it was once a highly active worm, its impact has shrunk significantly, likely due to improved security measures, patching, and changes in global network infrastructure. The next section will take this analysis further by investigating the geographical origins of these packets, identifying which regions continued to be affected by Slammer traffic in later years.

4.5 Geolocation Analysis

This section builds upon the geolocation and source analysis methods described in (Section 3.6.7) by presenting the results of Slammer traffic grouped by country and /16 network blocks from 2005 to 2024.

4.5.1 Analysis and Visualization

The analysis of unique IP addresses involved in Slammer worm activity from 2005 to 2024 in the (Table 4.5) shows clear geographical patterns. China has the highest number of unique IP addresses with 54,370, accounting for 33.87% of all Slammer related hits during these years. The United States

follows with 20,932 unique IPs, representing 13.04% . The United Kingdom and Japan also rank high, with 12,912 (8.05%) and 10,268 (6.40%) unique addresses, respectively. Brazil, India, and Russia each contribute moderately, while Spain, France, and Taiwan show the lowest count among the top ten. This results may indicate that Slammer traffic has primarily originated from or passed through China and the United States, highlighting these regions as central points in Slammer worm propagation over the years studied.

The analysis of Slammer worm sources by /16 netblocks from 2005 to 2024 (Table 4.6) highlights that specific IP ranges in China dominate the list of sources. The top three netblocks, all located in China, account for considerable number of unique IPs, with 220.184.0.0./16 at the top, containing 2,569 unique IP addresses and making 1.60% of total unique IPs. Other notable Chinese netblocks include 60.176.0.0/16 with 1,996 IPs (1.24%) and 222.183.0.0/16 with 1,277 IPs (0.80%). The United States is appears in the top ten with a single netblock 4.245.0.0/16, contributing 932 unique IP addresses (0.58%). India is also represented with 220.226.0.0/16, accounting for 688 unique IPs (0.43%).

Further analysis though outside the primary scope of this research, of broader netblocks beyond the /16 standard reveals that Chinese ranges continue to lead, with 60.176.0.0/12 (4,812 IPs, 3.00%) and 220.184.0.0/13 (4,277 IPs, 2.66%) as the top two. These are followed by the United Kingdom's 172.128.0.0/11 (3,882 IPs, 2.42%) and 172.160.0.0/11 (3,439 IPs, 2.14%). This concentration of Slammer activity within a few large Chinese netblocks may indicate localized vulnerabilities or a high density of compromised hosts in those regions.

Country	Unique IP Addresses	Percentage of Unique Hits
CN (China)	54370	33.87%
US (United States)	20932	13.04%
GB (United Kingdom)	12915	8.05%
JP (Japan)	10268	6.40%
BR (Brazil)	5824	3.63%
IN (India)	4566	2.84%
RU (Russia)	4101	2.55%
ES (Spain)	3518	2.19%
FR (France)	3232	2.01%
TW (Taiwan)	2914	1.82%

Table 4.5: Top 10 Slammer Hits from Unique Sources per Country (2005-2024)

Country	/16 Netblock	Unique IPs	Percentage of Total Unique IPs
CN (China)	220.184.0.0/16	2569	1.60%
CN (China)	60.176.0.0/16	1996	1.24%
CN (China)	222.183.0.0/16	1277	0.80%
CN (China)	218.0.0.0/16	963	0.60%
US (United States)	4.245.0.0/16	932	0.58%
CN (China)	222.182.0.0/16	926	0.58%
CN (China)	60.186.0.0/16	879	0.55%
CN (China)	220.191.0.0/16	828	0.52%
CN (China)	218.72.0.0/16	757	0.47%
IN (India)	220.226.0.0/16	688	0.43%

Table 4.6: Top 10 Slammer Worm Sources from Unique IP Addresses by /16 Netblock (2005-2024)

Additionally, an alternative visualization method was explored by feeding data from Table 4.8 into

ChatGPT (version 4.0) to generate a heatmap (Figure 4.4), visually representing the concentration of unique IP addresses across top countries. For transparency, this method was introduced at the same university where this thesis is being evaluated and was approved by the supervising professor before its implementation.

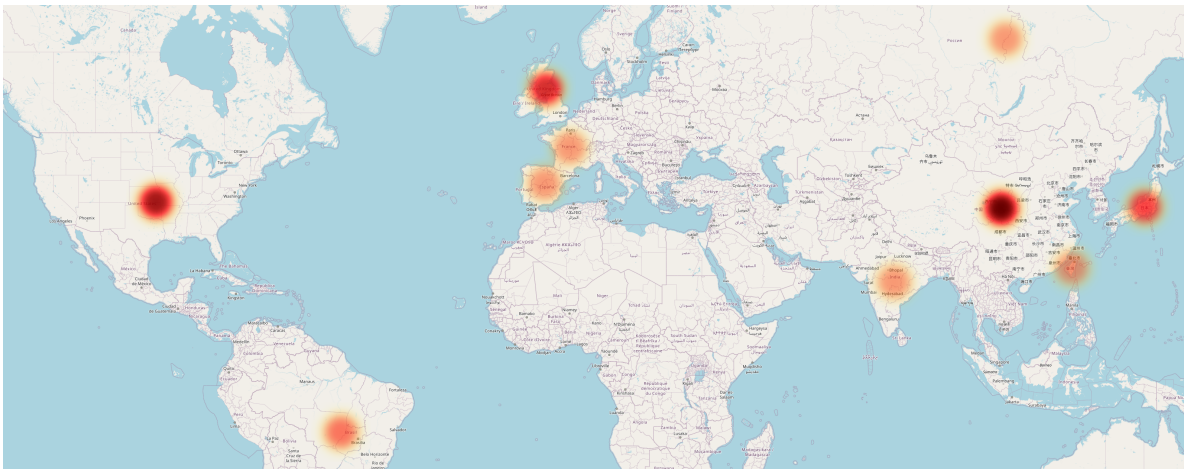

Figure 4.4: AI-made Visual representation of findings from Table 4.8

The detailed annual findings from the country and netblock analysis, generated using the methods described in Chapter 3, contain too much data to include in the main report. To keep the document clear while still showing all results, these findings appear as tables in the appendix. The complete tables are available in Appendix-B.

4.5.2 Annual Findings

The Slammer worm showed high activity levels during 2005 and 2006, with China clearly emerging as the dominant source, accounting for approximately 31–34% of all unique IP addresses. The United States and United Kingdom followed as secondary sources but with significantly fewer hits. Specific Chinese netblocks, particularly 220.184.0.0/16 and 60.176.0.0/16, contained most of the active IPs during this period.

Activity decreased significantly in 2007 and 2008, though China remained the primary source with 28–37% of detections. The United States maintained its second place, while other countries including Russia, Japan, and Brazil appeared more frequently. The worms spread became slightly more scattered during these years, with Chilean and Saudi Arabian netblocks appearing among the top sources for the first time.

In 2009, China strongly reasserted its dominance with nearly 44% of IP hits, while Russia and the United States continued at much lower levels. A new pattern emerged in 2010 as Russia significantly increased its presence, particularly through the 94.242.0.0/16 netblock, while both China and the United States showed reduced but steady activity compared to earlier years.

The downward trend continued through 2011 and 2012, with Russia overtaking China as the leading source, primarily through the 94.242.0.0/16 netblock. China remained active but fell to second place,

followed by consistent but diminished presence from the United States and Taiwan. Total unique IP counts during these years were dramatically lower than in the initial outbreak period.

Activity became minimal in 2013 and 2014, with China regaining the top position in 2013 while Russia remained close behind. Taiwan and the United States continued to appear but with extremely low IP counts. By 2014, detections dwindled to approximately 100 unique IPs total, consisting of scattered minor hits from various countries including Chile and Sri Lanka.

From 2015 through 2017, worm activity reached near extinction levels, with China accounting for 88–90% of the few remaining detections. The United States and South Korea appeared briefly in 2015 before disappearing again, while 2016 and 2017 saw most activity concentrated in specific Chinese netblocks like 171.8.0.0/16.

The period from 2018 to 2024 showed almost no detectable activity. China remained the sole source in both 2018 and 2021, each year producing only a handful of IP addresses. By 2021, only a single IP from the Chinese 180.169.0.0/16 netblock was observed. No standard Slammer activity was recorded in the telescope data for the intervening years or any period following 2021, indicating the worms effective disappearance from monitored networks.

This detailed geolocation analysis can be reviewed by the tables in the Appendix-B, the tables clearly illustrate the geographical evolution of Slammer worm activity from the third quarter of 2005 to its complete disappearance after 2021.

4.6 Summary

The analysis also presents a lifetime timeline, showing that Slammer was initially highly active, especially between 2005 and 2008, with a gradual decline leading to almost no activity after 2018. Network traffic verification indicated that a portion (1.7 million packets) captured by the network telescope for this research was irrelevant and required a filtering process to isolate Slammer traffic on UDP ports 1433 and 1434. Leaving **74.8 million** packets available for detailed analysis by size, hash, and checksum confirmed that most captures matching the worms known Slammer signature. The results of the first analysis identified **4.1 million** frames that match the known 376-byte and hash of the SQL Slammer worm payload. A separate sweep of the 300 most common non Slammer hashes against the VirusTotal API returned only two matches on the platform. The known Slammer signature and one other `'ntp_123_monlist.pkt'` which was marked as non malicious, confirming that no other malicious UDP payloads were present on ports 1433/1434 or that these payloads still not identified by the known security vendors.

Monthly and daily time-series an analysis show heavy activity from 2005–2008, a steady decay through 2014, and only isolated spikes thereafter. No Slammer packets were seen in 2019–2020, a single stray frame appeared in May 2021, and nothing at all was identified during 2022–2024. Capture gaps in 2016 were confirmed as pre-warned by the data provider.

Geolocation analysis showed that much of the activity originated from specific netblocks and regions, with China and the United States at the top of the list most years, followed by Great Britain and Japan in the earlier years. Russia appeared in high positions between 2009 and 2015, and Taiwan was

prominent between 2010 and 2013. Country based IP and /16 netblocks result-tables that support these findings are provided in Appendix-B

These findings highlight the worms continuous presence over several years, as well as the eventual near disappearance of its traffic in the datasets studies. The next chapter will assess the relevance of the Slammer worm over time, and reflect on the research process. It will also discuss possible reasons behind the worms evolution and suggest directions for future work.

5

Conclusion

5.1 Introduction

This chapter brings together the elements of this research on the SQL Slammer worm, summarizing how each step has contributed to understanding its long-term behaviour. The research started with an detailed review of the literature on related subjects such as: network telescopes, Internet Background Radiation, different researches done on other worms, tools and methods that are used for this type of research, and the history of Slammers initial outbreak as well as later observations of the worm. The next step was to describe methodology of the research, an empirical approach combining packet based inspection, payload validation, time-series analysis, and geolocation analysis to track Slammers activity. After methodology, Chapter 4 presents the results, analysing the worms presence, mapping its peaks and declines as well as locating geographical hot spots. Together, these steps have provided a clear picture of Slammer activity over almost two decades, as well as how the same process can be replicated by following the outlined methodology.

This concluding chapter begins by providing a summary of the research, revisiting key points from the literature, methodology, and findings (Section 5.2). This is followed by (Section 5.3), which reflects on how each research objective and sub-objective was addressed throughout the study. (Section 5.4) discusses the main contributions this research makes to the field of network security and malware analysis. Finally, (Section 5.5) outlines possible directions for future work, including technical, analytical, and theoretical extensions of this study.

5.2 Summary of Research

The literature review traced how network telescopes have long enabled passive monitoring of unsolicited internet traffic and how early studies rated the Slammer worm as one of the fastest-spreading worms in history (Moore et al., 2003; Stanifor et al., 2004). One of the major takeaways from the literature was the documented findings of Chindipha and Irwin (2017), suggesting that SQL Slammer might be making a comeback after years of inactivity, potentially indicating its presence in the years to come. In contrast, the more recent study by Kristoff (2023) stated that there is no evidence of Slammers presence anywhere on the internet, suggesting that it has completely died out. This contrast raised the question of whether Slammer had truly disappeared or if it is still lurking in under-patched corners of the internet.

The methodology chapter explained the full process of how 19 years of network telescope captures were filtered to UDP traffic on ports 1433 and 1434, then standardized into monthly PCAPs, and carefully analysed using size-based filtering, MD5 hash values, and checksum inspection to identify the known signatures of the SQL Slammer payload. A large number of separate automated scripts were developed for each step of the research. The methodology section describes the core functions of these scripts, along with simplified examples of the commands used and where the scripts were applied. Automated scripts counted monthly Slammer packets, extracted timestamps for time-series graphs, and mapped source IP addresses via geolocation lookups. This systematic process gave the research both a detailed and reproducible outcome. The full code of the scripts mentioned in Chapter 3 can be found in the artefact appendix, along with a description of how to use them in the README file.

The results chapter showed that 74.8 million UDP packets remained after initial port filtering to ports 1433 and 1434, of which 4.1 million matched the Slammer signature. The time-series results recorded a large presence of the worm with peaks and drops from 2005 to 2008, followed by a multi-year decline, a sporadic reappearance in late 2016, and near-total extinction by 2019, with only a single Slammer packet in 2021 and none thereafter. Geolocation analysis revealed persistent hotspots in China and the United States, which later shifted to Russia in later years. No other malicious payloads were detected on the same ports. These findings confirmed that Slammer never fully vanished until recent years, if one includes the small reappearances in 2018 and 2019, which might be much larger and more visible with more resources and additional telescopes in multiple regions. However, based on the findings of this research, the outcome aligns more closely with Kristoff (2023) statement that Slammers days have come to an end, rather than with the conclusion of Chindipha and Irwin (2017) paper published six years earlier.

5.3 Research Objectives

The primary objective of the research was to analyse the evolution and movement of Slammer worm activity from 2005 to 2024. This has been achieved through continuos monitoring of UDP traffic on ports 1433 and 1434 across a nineteen year period (Section 4.6), revealing both the worms rapid spread and activity in earlier years and its its decline, with clear evidence of spikes in 2016 and near zero activity after 2019.

The first sub-objective examined how Slammer payloads have changed over time an whether they

could be accurately identified. Payload-size filtering isolated 376 byte payload carried in a 418 byte network frame, and MD5 hashing confirmed that result by matching payload hashes against known Slammer hash value, with no significant deviations over the study period. Hash probes along with checksum analysis provided an validation that the worms core payload remained unchanged and could still be reliably recognized (Section 3.6.1, Section 4.3).

The second sub-objective investigated whether other malicious attacks targeted the same UDP ports 1433 and 1434 which belong to Microsoft SQL Server. After extracting year by year hash values of all UDP payloads in the dataset and filtering for frames 90 bytes or larger, the top 300 payload hashes were submitted to VirusTotal. Only one non-Slammer payload pattern was identified, which was related to NTP traffic, however this hash was categorized as non-malicious by VirusTotal (Section 3.6.4, Section 4.3.2). In addition, the analysis uncovered a group of non-Slammer payloads, each appearing exactly 256 times, with six of them sharing an identical payload size of 798 bytes. While not flagged as malicious, their similarity in count and structure may suggest automated scanning or botnet driven probing activity. These findings overall suggest that Slammer was the only confirmed threat exploiting these UDP ports, with no verified evidence of other UDP based worms or malware in the same space.

The third sub-objective explored-if certain geographical regions or networks were more frequently affected by Slammer traffic the others. Geolocation of 4.1 million packets showed that China and United States were the most dominant sources throughout whole period of research, making together over 46 percent of unique IPs caring Slammer malicious payloads. With Russia emerging as an major source around 2009-2015 (Section 3.6.7, Section 4.5). Initially there was an additional subsub-objective to this geolocation analysis. An attempt was made to compare traffic spikes with known geopolitical events as well as cyber security events over time. The author had a particular interest in exploring potential connections between traffic originating from regions such as Russia and Ukraine, as well as Taiwan and China, and major geopolitical events linked to these regions. Such as the 2014 annexation of Crimea, the NotPetya attack in 2017, and the full-scale invasion of Ukraine 2022. The idea was that SQL Slammer could have served as one of many tools in the broader cyber warfare arsenal particularly in “unpatched” regions, alongside well-known exploits such as EternalBlue, Mimikatz, and others. However, the volume and complexity of such linking-process turned out to be too broad to be thoroughly analysed in this research. Given the limited time and resources available, as well as the age of the last real Slammer spikes identified in this dataset, pursuing this research direction would have reduced the analysis to speculation rather than the serious research this project aimed to maintain.

Overall, each objective in the scope was met by aligning empirical evidence with the research design. The primary objective was satisfied with a detailed time-series overview, while each sub-objective achieved through methodical analysis, as detailed in the sections (Chapter 3, Chapter 4).

5.4 Research Contribution

This work makes several contributions to the field of network security, and more specifically to the analysis of malware in UDP traffic. First, it presents the longest continuous telescope-based study of any internet worm, and specifically the SQL Slammer worm known to the author, extending a previously published eight-year analysis to a full nineteen years (2005–2024). This extended period

helps fill a critical gap in understanding worm persistence over nearly two decades. The study also provides detailed timestamp data, tables, and graphs covering the entire period, offering others the opportunity to build upon this dataset for further research.

The study demonstrates a robust multi-method procedure, combining port filtering and different types of payload and geographical analysis to isolate the Slammer worm payload with a high rate of success. The structured methods, as well as the tools and scripts developed during this research, could serve as valuable starting points for others conducting similar investigations.

The analysis clearly shows that no other UDP-based malicious payloads exploited ports 1433 and 1434 during the study period, highlighting the Slammer worms specific attack vector and the importance of timely patching to close security gaps.

5.5 Future Work

Building on these findings, several directions for further research are proposed. First, a detailed geopolitical analysis could explore correlations between worm traffic spikes and specific geopolitical events. This would possibly require combining network telescope data with open source intelligence on cyber campaigns and political incidents, perhaps with an finer time resolution than monthly summarises.However, this approach may be limited by the cost and availability of historical data, which can be a significant challenge.

Using time based epidemic models, such as updated version of the SIS model, could help visualize how the speed of patching affects how quickly the worm disappears or come back again after patching. These models could also be used alongside real data to test how different defence strategies might work.

As IPv6 adoption continues to grow, and with the expectation that it will eventually replace IPv4 as the dominant internet protocol, a possible direction for future work could be to replicate this research using a network telescope configured for IPv6 traffic. Although this would completely exclude research on Slammer worm, which is specific to IPv4, it could still help anticipate the behaviour of future worms in the next-generation internet.

Finally, an other possible direction for future work could be development of a complete analysis tool based on the scripts and commands used in this research. Throughout this project, several smaller scripts were created to handle individual tasks for each step of the analyses like: filtering by port, checking payload size, validating hashes, enriching IP addresses with geolocation, and generating output files, tables and charts in a different folders. Each script works well on its own in most cases, while some scripts required pre- or post-processing using command-line filtering to achieve the wanted result. A useful improvement would be to merge all these components into a single, unified tool that automates the entire analysis process. In addition, the tool could feature a simple prompt menu or a graphical interface that allows the user to select PCAP files and run all steps with the click of a button. Such a tool would make the analysis process faster, reduce human error, and enable less experienced users to apply the same methods without needing in-depth knowledge of technical details or programming languages like Python or Bash.

References

Ahmad, M. A. (2019). The V-network: a testbed for malware analysis [Accessed: 2024-10-17]. *Science World Journal*, *14*(3), 70–76. https://doi.org/10.1109/ICACCCT.2016.7831716

Bajaj, P., & Guha Roy, A. (2004). Source Code Analysis of Worms. *Proceedings of the Midwest Instruction and Computing Symposium 2004*. https://www.micsymposium.org/mics_2004/Bajaj.pdf

Bortoluzzi, F. (2024). Lecture Notes from Computer Network Attack (UC3CNA10).

Bortoluzzi, F., Irwin, B., Beiler, L. S., & Westphall, C. M. (2023). Cloud Telescope: A distributed architecture for capturing Internet Background Radiation. *2023 IEEE 12th International Conference on Cloud Networking (CloudNet)*, 77–85. https://doi.org/10.1109/CloudNet59005.2023.10490018

Bulygin, Y. (2013). A Spread Model of Flash Worms. *Security Evaluation Center of Excellence, Intel Corporation*, 7. https://www.c7zero.info/stuff/flash__bulygin_malware06.pdf

Chen, T., Blasco, J., Alzubi, J., & Alzubi, O. (2014). Intrusion Detection. *Engineering & Technology Reference*, 1–9. https://doi.org/10.1049/etr.2014.0007

Chen, T., & Robert, J.-M. (2004). Worm Epidemics in High-Speed Networks. *Proceedings of the IEEE Computer Society*, 48–53. https://doi.org/10.1109/MC.2004.36

Chindipha, S. D., & Irwin, B. (2017). An analysis on the re-emergence of SQL Slammer worm using network telescope data. *Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, 222–227. https://api.semanticscholar.org/CorpusID:64758335

Cisco Systems, Inc. (2003). SAFE SQL Slammer worm attack mitigation [Accessed: 2024-09-24]. https://www.cisco.com/web/FR/documents/pdfs/tdm/threat/sqlsw_wp.pdf

Combs, G., & Contributors. (2023). *Wireshark Users Guide* [Accessed: 2024-10-18]. https://www.wireshark.org/docs/wsug_html_chunked/

Cymru Team. (2023). IP to ASN Mapping Service [Accessed: 2024-10-18]. https://team-cymru.com/community-services/ip-asn-mapping/

Fuentes, F., & Kar, D. C. (2005). Ethereal vs. Tcpdump: a comparative study on packet sniffing tools for educational purpose. *Journal of Computing Sciences in Colleges*, *20*(4), 169–176. https://doi.org/10.5555/1047846.1047873

Hughes, K., & Qu, Y. (2012). A Theoretical Model: Using Logistic Regression for Malware Signature Based Detection. *Proceedings of the 10th IEEE International Conference on Dependable, Automatic and Secure Computing*, 1–6.

Irwin, B. (2013). A baseline study of potentially malicious activity across five network telescopes. *2013 Cyber Conflict (CyCon)*, *5*, 18 pages. https://catalog.caida.org/paper/2013_1_irwin_b_researchgate_6547386

Irwin, B., Pilkington, N., Barnett, R., & Friedman, B. (2007). A Geopolitical Analysis of Long Term Internet Network Telescope Traffic. *Security and Networks Research Group (SNRG), Department of Computer Science, Rhodes University, Grahamstown, South Africa, Conference: SATNAC 2007*.

Krause, H. (2021). Approaches to Analysing Malware Received from a Reactive Network Telescope [Accessed: 2024-09-24]. https://inet.haw-hamburg.de/teaching/ws-2020-21/project-class/henning_krause_fw1.pdf

Kristoff, J. (2023). Remembering SQL Slammer [Accessed: 2024-10-16]. https://www.netscout.com/blog/asert/remembering-sql-slammer

Li, Z., Rios, A. L. G., & Trajković, L. (2020). Detecting Internet Worms, Ransomware, and Blackouts Using Recurrent Neural Networks. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2165–2172. https://doi.org/10.1109/SMC42975.2020.9283472

Märtens, M., Asghari, H., van Eeten, M., & Van Mieghem, P. (2016). A time-dependent SIS-model for long-term computer worm evolution. *2016 IEEE Conference on Communications and Network Security (CNS)*, 207–215. https://doi.org/10.1109/CNS.2016.7860487

Microsoft Corporation. (2002). Microsoft Security Bulletin MS02-039 [Accessed: 2024-12-28]. https://learn.microsoft.com/en-us/security-updates/securitybulletins/2002/ms02-039

MITRE Corporation. (2002). CVE-2002-0649 [Accessed: 2024-12-28]. https://www.cve.org/CVERecord?id=CVE-2002-0649

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., & Weaver, N. (2003, January). *The Spread of the Sapphire/Slammer Worm* (tech. rep.). CAIDA. https://catalog.caida.org/paper/2003_sapphire

Paxson, V., Moore, D., Savage, S., Shannon, C., Staniford, S., & Weaver, N. (2003). Inside the Slammer Worm. *IEEE Security and Privacy*, *1*(4), 33–39. https://doi.org/10.1109/MSECP.2003.1219056

Pradeepini, G., Sai, G., & Aruna, V. (2018). Hybrid Pcap Analyser using T-Shark a tool that Makes use of Open Source Analyser that Can Meet Industrial Standards. *International Journal of Engineering & Technology*, *7*(4.17), 85–88. https://www.sciencepubco.com/index.php/ijet/article/view/21808

Schultz, E. E., Mellander, J., & Peterson, D. R. (2003). The MS-SQL Slammer Worm. *Network Security*, *2003*(3), 10–14. https://doi.org/10.1016/S1353-4858(03)00310-6

Sikos, L. (2020). Packet analysis for network forensics: A comprehensive survey. *Forensic Science International: Digital Investigation*, *32*, 12. https://doi.org/10.1016/j.fsidi.2019.200892

Stanifor, S., Moore, D., Paxson, V., & Weaver, N. (2004). The Top Speed of Flash Worms. *ACM Workshop on Rapid Malcode (WORM)*, 33–42. https://doi.org/10.1145/1029618.1029624

Travis, G., Balas, E., Ripley, D., & Wallace, S. (2003). *Analysis of the SQL Slammer Worm and Its Effects on Indiana University and Related Institutions* (tech. rep.) (Accessed: 2024-10-17). Advanced Network Management Lab, Indiana University. https://cisre.egr.uh.edu/wp-content/uploads/2023/09/slammer.pdf

Wei, S., & Mirkovic, J. (2008). Correcting congestion-based error in network telescopes observations of worm dynamics. *Internet Measurement Conference (IMC)*, 125–130. https://doi.org/10.1145/1452520.1452536

Wustrow, E., Karir, M., Bailey, M., Jahanian, F., & Huston, G. (2010). Internet Background Radiation revisited. *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, 62–74. https://doi.org/10.1145/1879141.1879149

A

Time Series Charts of Slammer Packet Activity (2005–2024)

Appendix A presents the bar and line charts generated from the time series analysis, visualizing the monthly and daily packet counts of Slammer related traffic observed over the 2005–2024 study period. The retrieval methods are described in Section 3.6.6, and the results are discussed in Section 4.4.

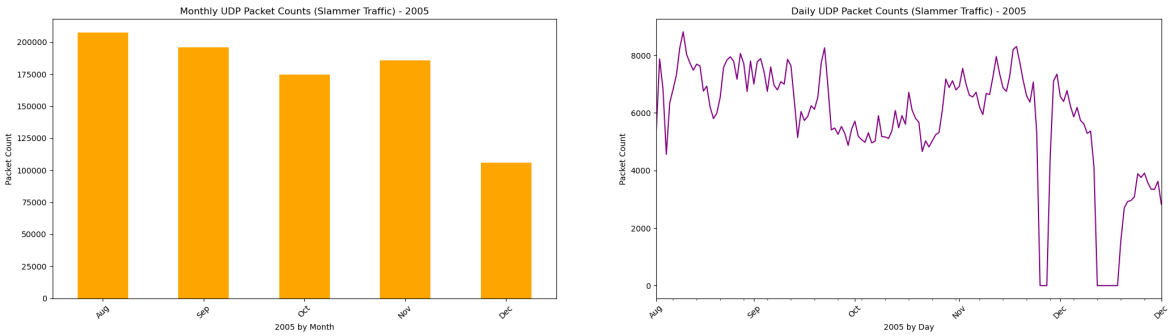

Figure A.1: Monthly and Daily Timestamp Analysis for 2005

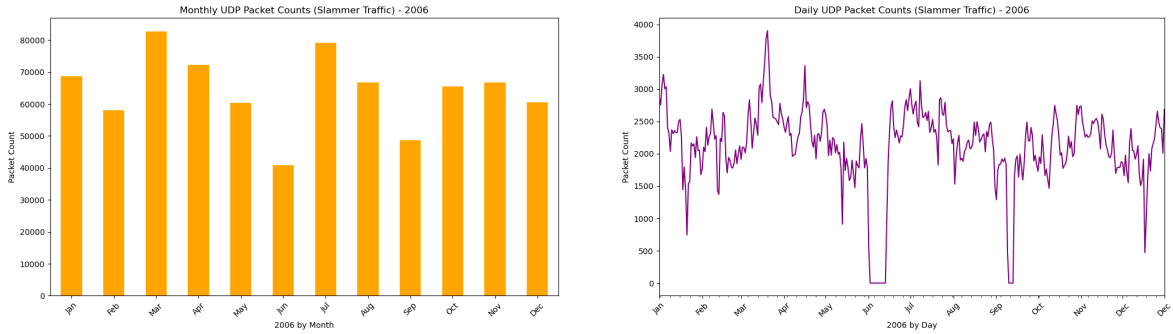

Figure A.2: Monthly and Daily Timestamp Analysis for 2006

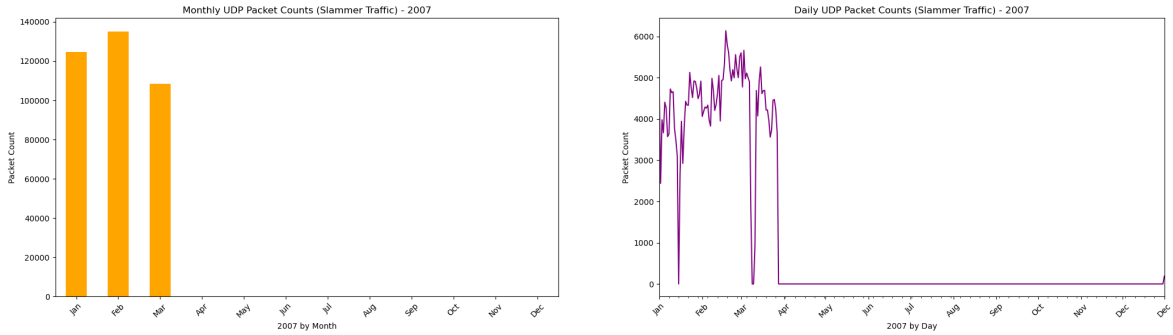

Figure A.3: Monthly and Daily Timestamp Analysis for 2007

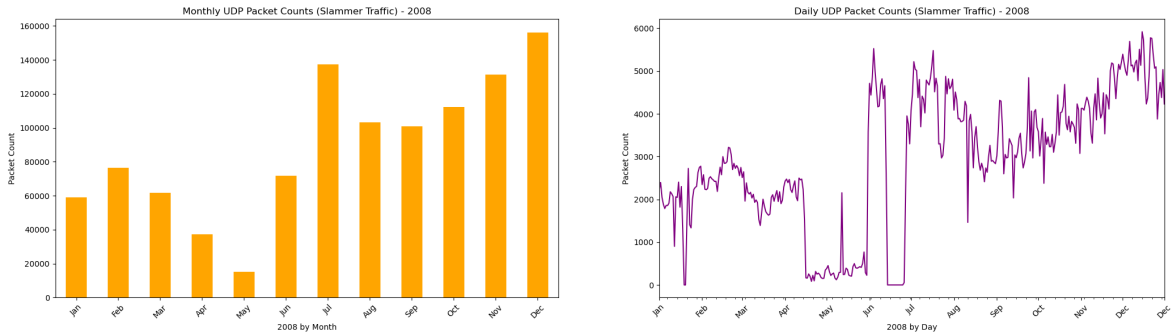

Figure A.4: Monthly and Daily Timestamp Analysis for 2008

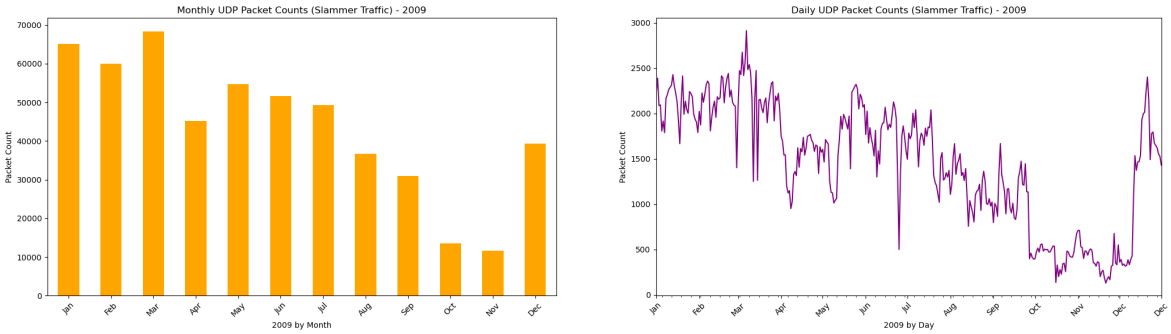

Figure A.5: Monthly and Daily Timestamp Analysis for 2009

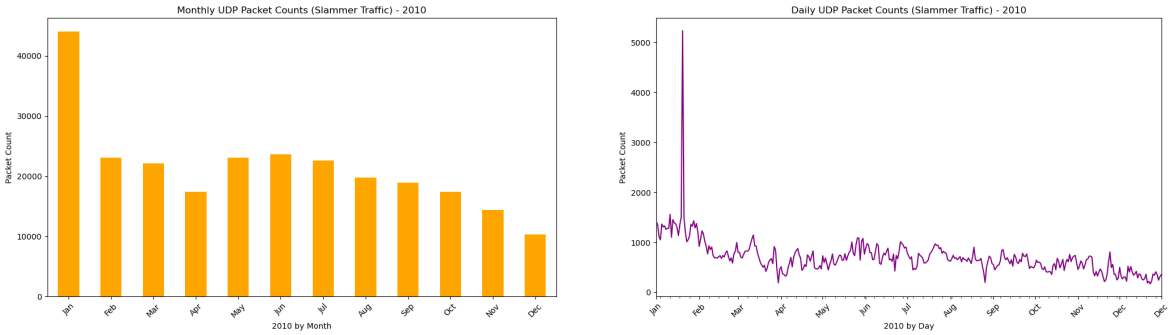

Figure A.6: Monthly and Daily Timestamp Analysis for 2010

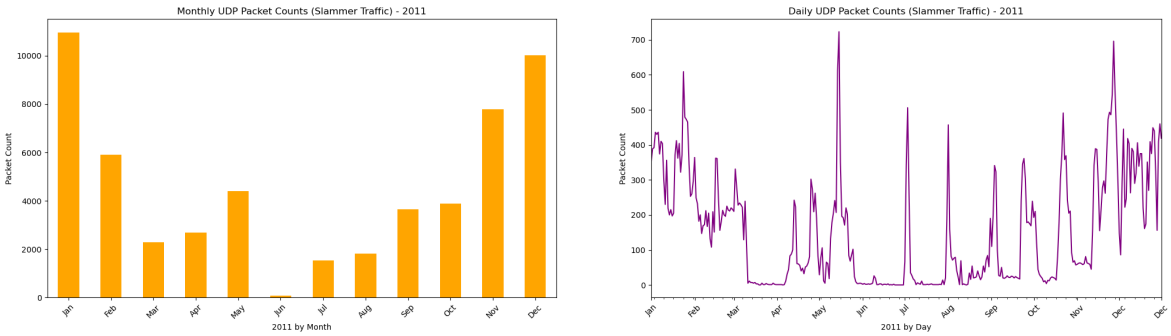

Figure A.7: Monthly and Daily Timestamp Analysis for 2011

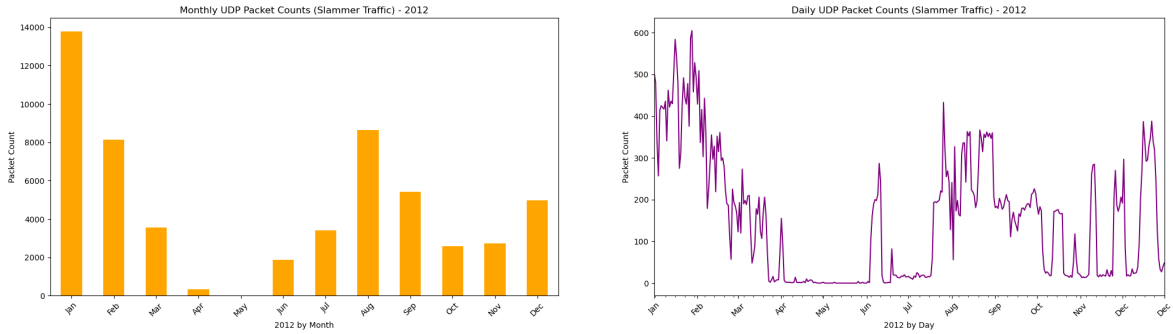

Figure A.8: Monthly and Daily Timestamp Analysis for 2012

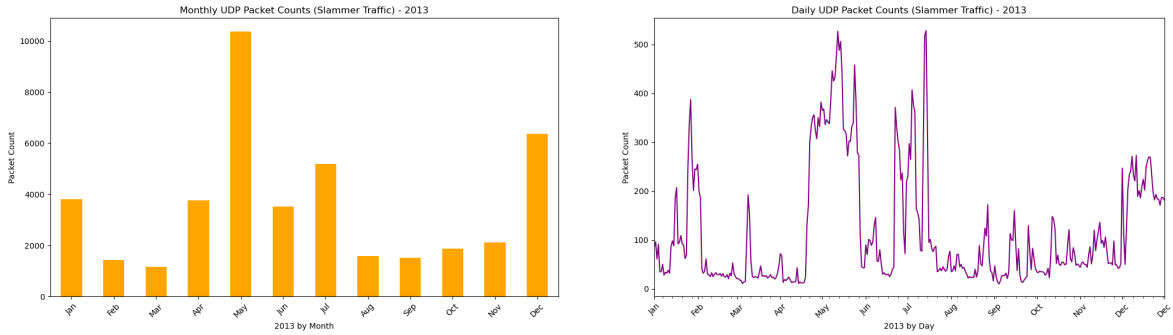

Figure A.9: Monthly and Daily Timestamp Analysis for 2013

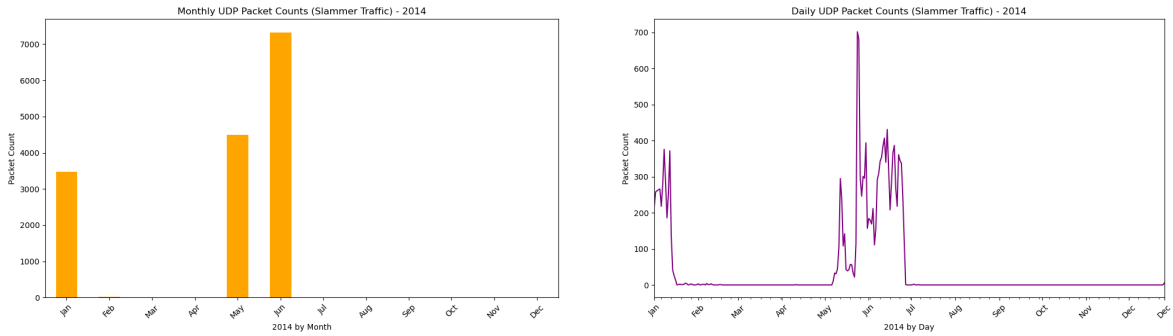

Figure A.10: Monthly and Daily Timestamp Analysis for 2014

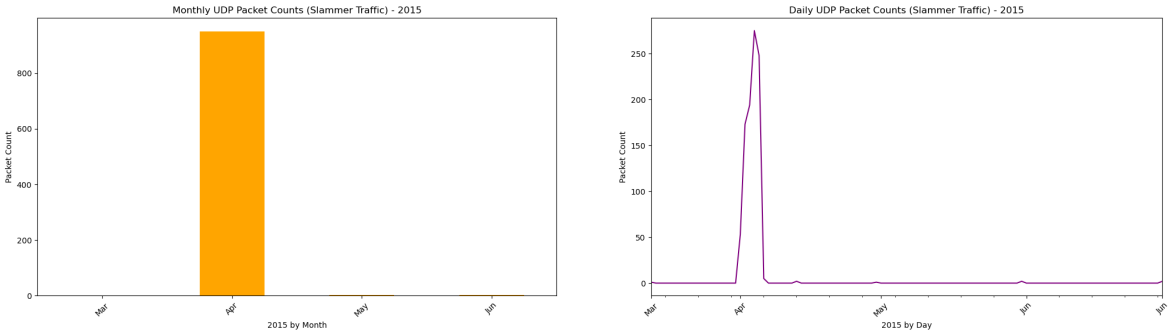

Figure A.11: Monthly and Daily Timestamp Analysis for 2015

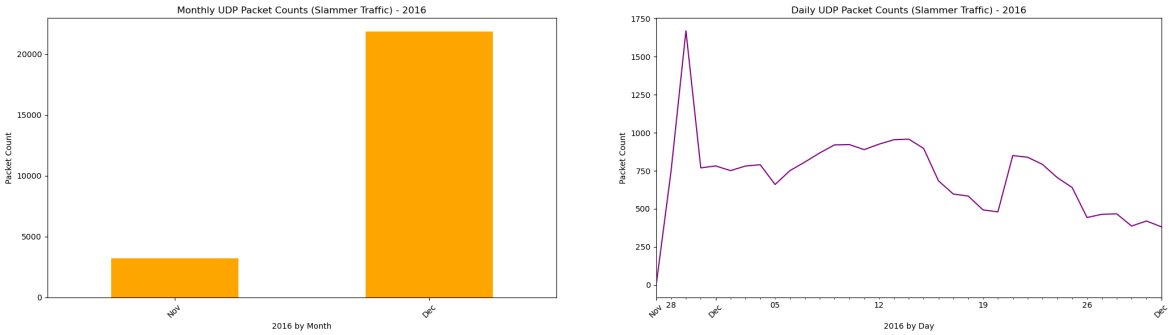

Figure A.12: Monthly and Daily Timestamp Analysis for 2016

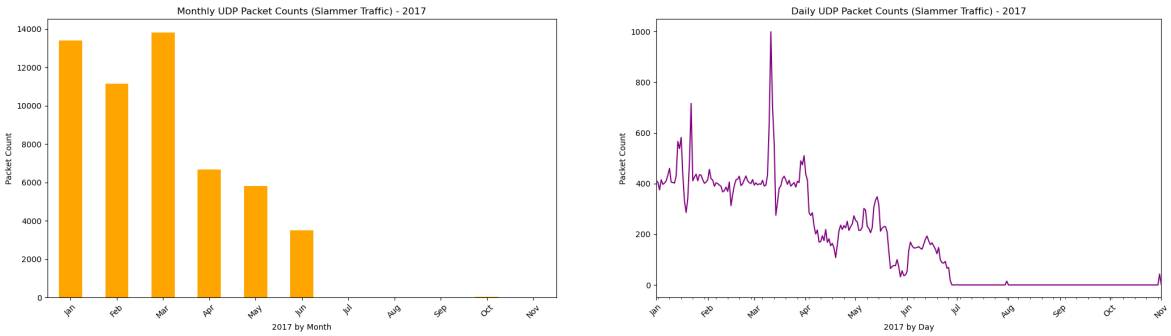

Figure A.13: Monthly and Daily Timestamp Analysis for 2017

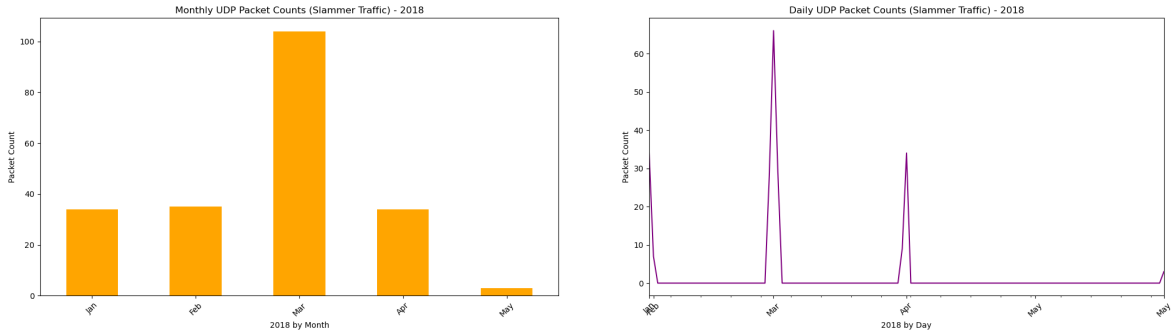

Figure A.14: Monthly and Daily Timestamp Analysis for 2018

B

Geolocation Tables of Slammer (2005–2024)

Appendix B presents the complete tables of Slammer related IP activity grouped by country and /16 netblocks, generated for each year from 2005 to 2024. These tables were produced using the geolocation analysis methods described in Section 3.6.7 and support the findings discussed in Section 4.5.2.

Country	Unique IPs	Percentage
CN	14863	33.68%
US	6204	14.06%
GB	4675	10.59%
JP	2589	5.87%
BR	1480	3.35%
IN	1031	2.34%
FR	1017	2.30%
ES	979	2.22%
MX	792	1.79%
TW	756	1.71%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CN	220.184.0.0/16	1318	2.99%
CN	60.176.0.0/16	1024	2.32%
CN	222.183.0.0/16	435	0.99%
CN	218.0.0.0/16	427	0.97%
CN	218.72.0.0/16	355	0.80%
CN	220.191.0.0/16	354	0.80%
US	4.245.0.0/16	347	0.79%
CN	218.6.0.0/16	274	0.62%
CN	222.182.0.0/16	266	0.60%
CN	218.88.0.0/16	263	0.60%

(b) Top /16 Netblocks by Slammer Hits

Table B.1: Slammer Activity 2005 (Total Unique IPs 44136)

Country	Unique IPs	Percentage
CN	20785	30.86%
US	8970	13.32%
GB	6251	9.28%
JP	4605	6.84%
BR	2516	3.74%
IN	2083	3.09%
ES	1651	2.45%
FR	1442	2.14%
TW	1050	1.56%
MX	1047	1.55%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CN	220.184.0.0/16	1241	1.84%
CN	60.176.0.0/16	994	1.48%
CN	60.186.0.0/16	750	1.11%
CN	222.182.0.0/16	596	0.88%
CN	222.183.0.0/16	572	0.85%
CN	218.0.0.0/16	504	0.75%
CN	220.191.0.0/16	451	0.67%
US	4.245.0.0/16	428	0.64%
IN	220.226.0.0/16	405	0.60%
CN	218.72.0.0/16	392	0.58%

(b) Top /16 Netblocks by Slammer Hits

Table B.2: Slammer Activity 2006 (Total Unique IPs 67360)

Country	Unique IPs	Percentage
CN	2913	27.95%
US	1511	14.50%
GB	755	7.24%
JP	734	7.04%
BR	403	3.87%
IN	384	3.68%
DE	263	2.52%
FR	257	2.47%
ES	237	2.27%
RU	224	2.15%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CN	222.183.0.0/16	127	1.22%
CN	58.51.0.0/16	101	0.97%
IN	220.226.0.0/16	68	0.65%
CN	58.50.0.0/16	50	0.48%
CN	220.184.0.0/16	49	0.47%
MA	81.192.0.0/16	45	0.43%
US	71.101.0.0/16	45	0.43%
CN	60.166.0.0/16	45	0.43%
MA	196.206.0.0/16	44	0.42%
IN	203.94.0.0/16	43	0.41%

(b) Top /16 Netblocks by Slammer Hits

Table B.3: Slammer Activity 2007 (Total Unique IPs 10422)

Country	Unique IPs	Percentage
CN	7883	37.26%
US	1976	9.34%
JP	1436	6.79%
BR	817	3.86%
GB	778	3.68%
RU	615	2.91%
IN	590	2.79%
DE	573	2.71%
AR	553	2.61%
PL	342	1.62%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CL	190.82.0.0/16	149	0.70%
CN	219.139.0.0/16	145	0.69%
CN	116.209.0.0/16	144	0.68%
SA	84.22.0.0/16	132	0.62%
CN	59.172.0.0/16	132	0.62%
CN	116.207.0.0/16	126	0.60%
CN	59.173.0.0/16	124	0.59%
CN	222.183.0.0/16	124	0.59%
CN	60.166.0.0/16	106	0.50%
CN	58.45.0.0/16	105	0.50%

(b) Top /16 Netblocks by Slammer Hits

Table B.4: Slammer Activity 2008 (Total Unique IPs 21154)

Country	Unique IPs	Percentage
CN	5358	43.52%
US	1195	9.71%
JP	538	4.37%
RU	497	4.04%
BR	379	3.08%
GB	330	2.68%
SA	308	2.50%
DE	275	2.23%
IN	245	1.99%
ES	243	1.97%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
SA	82.118.0.0/16	160	1.30%
CN	222.219.0.0/16	117	0.95%
CN	122.6.0.0/16	112	0.91%
CL	190.82.0.0/16	97	0.79%
CN	60.166.0.0/16	95	0.77%
DE	79.219.0.0/16	92	0.75%
CN	119.96.0.0/16	80	0.65%
CN	222.86.0.0/16	77	0.63%
CN	59.172.0.0/16	76	0.62%
US	4.245.0.0/16	74	0.60%

(b) Top /16 Netblocks by Slammer Hits

Table B.5: Slammer Activity 2009 (Total Unique IPs 12311)

Country	Unique IPs	Percentage
CN	1514	30.70%
US	697	14.13%
RU	464	9.41%
JP	242	4.91%
TW	186	3.77%
IN	150	3.04%
BR	133	2.70%
DE	125	2.53%
ES	93	1.89%
SA	86	1.74%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
RU	94.242.0.0/16	117	2.37%
US	4.245.0.0/16	75	1.52%
DE	79.219.0.0/16	54	1.09%
SA	82.118.0.0/16	53	1.07%
CN	222.219.0.0/16	44	0.89%
CN	113.15.0.0/16	43	0.87%
IN	59.93.0.0/16	38	0.77%
HK	121.202.0.0/16	30	0.61%
CN	113.17.0.0/16	30	0.61%
RU	94.41.0.0/16	29	0.59%

(b) Top /16 Netblocks by Slammer Hits

Table B.6: Slammer Activity 2010 (Total Unique IPs 4932)

Country	Unique IPs	Percentage
RU	257	19.02%
CN	208	15.40%
US	206	15.25%
JP	84	6.22%
TW	73	5.40%
BR	48	3.55%
AR	34	2.52%
IN	32	2.37%
RO	29	2.15%
IT	24	1.78%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
RU	94.242.0.0/16	165	12.21%
RU	212.106.0.0/16	19	1.41%
CN	120.202.0.0/16	16	1.18%
SE	85.228.0.0/16	11	0.81%
RO	188.26.0.0/16	11	0.81%
RU	109.225.0.0/16	10	0.74%
SG	202.55.0.0/16	8	0.59%
US	71.75.0.0/16	7	0.52%
CN	61.178.0.0/16	7	0.52%
CN	183.245.0.0/16	7	0.52%

(b) Top /16 Netblocks by Slammer Hits

Table B.7: Slammer Activity 2011 (Total Unique IPs 1351)

Country	Unique IPs	Percentage
RU	263	31.46%
CN	108	12.92%
US	98	11.72%
TW	79	9.45%
GB	32	3.83%
JP	31	3.71%
IN	21	2.51%
AR	19	2.27%
BR	17	2.03%
MY	12	1.44%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
RU	94.242.0.0/16	114	13.64%
RU	109.225.0.0/16	42	5.02%
RU	212.106.0.0/16	29	3.47%
TW	122.121.0.0/16	20	2.39%
TW	111.253.0.0/16	11	1.32%
CN	222.37.0.0/16	10	1.20%
GB	172.129.0.0/16	10	1.20%
AR	186.110.0.0/16	9	1.08%
GB	172.162.0.0/16	9	1.08%
RU	94.41.0.0/16	7	0.84%

(b) Top /16 Netblocks by Slammer Hits

Table B.8: Slammer Activity 2012 (Total Unique IPs 836)

Country	Unique IPs	Percentage
CN	153	26.24%
RU	122	20.93%
TW	69	11.84%
US	40	6.86%
BR	26	4.46%
IN	17	2.92%
MX	15	2.57%
AR	13	2.23%
GB	10	1.72%
FR	10	1.72%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
RU	109.225.0.0/16	38	6.52%
RU	31.172.0.0/16	25	4.29%
RU	94.242.0.0/16	17	2.92%
TW	114.40.0.0/16	11	1.89%
CN	183.63.0.0/16	10	1.72%
CN	122.82.0.0/16	10	1.72%
CN	123.85.0.0/16	9	1.54%
TW	111.253.0.0/16	8	1.37%
RU	212.106.0.0/16	7	1.20%
RU	89.112.0.0/16	5	0.86%

(b) Top /16 Netblocks by Slammer Hits

Table B.9: Slammer Activity 2013 (Total Unique IPs 583)

Country	Unique IPs	Percentage
CN	34	31.78%
RU	10	9.35%
US	8	7.48%
CL	6	5.61%
ID	5	4.67%
BR	5	4.67%
TW	4	3.74%
LK	4	3.74%
AR	4	3.74%
LV	3	2.80%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CL	146.83.0.0/16	6	5.61%
CN	124.117.0.0/16	4	3.74%
LK	112.134.0.0/16	4	3.74%
RU	91.243.0.0/16	3	2.80%
CN	222.221.0.0/16	2	1.87%
CN	211.81.0.0/16	2	1.87%
AR	190.220.0.0/16	2	1.87%
CN	182.242.0.0/16	2	1.87%
CN	120.213.0.0/16	2	1.87%
CN	118.203.0.0/16	2	1.87%

(b) Top /16 Netblocks by Slammer Hits

Table B.10: Slammer Activity 2014 (Total Unique IPs 107)

Country	Unique IPs	Percentage
CN	7	33.33%
US	5	23.81%
KR	2	9.52%
RU	1	4.76%
PL	1	4.76%
PK	1	4.76%
IN	1	4.76%
CO	1	4.76%
AR	1	4.76%
AF	1	4.76%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CN	61.178.0.0/16	2	9.52%
KR	168.131.0.0/16	2	9.52%
RU	81.176.0.0/16	1	4.76%
US	68.166.0.0/16	1	4.76%
US	66.251.0.0/16	1	4.76%
CN	61.235.0.0/16	1	4.76%
US	50.206.0.0/16	1	4.76%
CN	211.139.0.0/16	1	4.76%
PK	203.215.0.0/16	1	4.76%
AF	203.215.0.0/16	1	4.76%

(b) Top /16 Netblocks by Slammer Hits

Table B.11: Slammer Activity 2015 (Total Unique IPs 21)

Country	Unique IPs	Percentage
CN	217	88.21%
US	11	4.47%
VN	2	0.81%
VE	2	0.81%
TW	2	0.81%
TH	2	0.81%
IN	2	0.81%
AR	2	0.81%
UA	1	0.41%
RU	1	0.41%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CN	171.8.0.0/16	54	21.95%
CN	123.149.0.0/16	35	14.23%
CN	101.81.0.0/16	17	6.91%
CN	180.164.0.0/16	14	5.69%
CN	1.199.0.0/16	12	4.88%
CN	1.192.0.0/16	10	4.07%
CN	58.39.0.0/16	8	3.25%
CN	58.34.0.0/16	4	1.63%
CN	222.69.0.0/16	4	1.63%
CN	101.85.0.0/16	4	1.63%

(b) Top /16 Netblocks by Slammer Hits

Table B.12: Slammer Activity 2016 (Total Unique IPs 246)

Country	Unique IPs	Percentage
CN	321	90.17%
US	11	3.09%
IN	7	1.97%
UG	3	0.84%
MX	3	0.84%
TW	2	0.56%
CR	2	0.56%
VE	1	0.28%
RU	1	0.28%
PL	1	0.28%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CN	171.8.0.0/16	57	16.01%
CN	123.149.0.0/16	35	9.83%
CN	223.72.0.0/16	31	8.71%
CN	58.39.0.0/16	18	5.06%
CN	222.69.0.0/16	18	5.06%
CN	180.174.0.0/16	17	4.78%
CN	222.72.0.0/16	16	4.49%
CN	116.238.0.0/16	16	4.49%
CN	61.172.0.0/16	14	3.93%
CN	180.162.0.0/16	10	2.81%

(b) Top /16 Netblocks by Slammer Hits

Table B.13: Slammer Activity 2017 (Total Unique IPs 356)

Country	Unique IPs	Percentage
CN	5	100.00%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CN	61.178.0.0/16	4	80.00%
CN	112.12.0.0/16	1	20.00%

(b) Top /16 Netblocks by Slammer Hits

Table B.14: Slammer Activity 2018 (Total Unique IPs 5)

Country	Unique IPs	Percentage
CN	1	100.00%

(a) Top 10 Countries by Unique IP Addresses

Country	/16 Netblock	Unique IPs	Percentage
CN	180.169.0.0/16	1	100.00%

(b) Top /16 Netblocks by Slammer Hits

Table B.15: Slammer Activity 2021 (Only 1 Unique IP)

Word count metrics

NUC Bachelor Project Word Count:

Total Sum count: 15008 Words in text: 14675 Words in headers: 182 Words outside text (captions, etc.): 151 Number of headers: 63 Number of floats/tables/figures: 22 Number of math inlines: 0 Number of math displayed: 0
(errors:4) NOTE: References are excluded.