An Analysis on the Re-emergence of SQL Slammer Worm Using Network Telescope Data

Stones Dalitso Chindipha*, Barry Irwin *†
*Department of Computer Science, Rhodes University, Grahamstown, South Africa.
[1]g15c7469@campus.ru.ac.za
†Council for Scientific and Industrial Research, Pretoria, South Africa.
[2]b.irwin@ru.ac.za

***Abstract*—The SQL Slammer worm is a self propagated computer virus that caused a denial of service on some Internet hosts and dramatically slowed down general Internet traffic. An observation of network traffic captured in the Rhodes University's network telescopes shows that traffic observed in it shows an escalation in the number of packets captured by the telescopes between January 2014 and December 2016 when the expected traffic was meant to take a constant decline in UDP packets from port 1434. Using data captured over a period of 84 months, the analysis done in this study identified top ten /24 source IP addresses that Slammer worm repeatedly used for this attack together with their geolocation. It also shows the trend of UDP 1434 packets received by the two network telescopes from January 2009 to December 2015. In line with epidemic model, the paper has shown how this traffic fits in as SQL Slammer worm attack. Consistent number of packets observed in the two telescopes between 2014 and 2016 shows qualities of the Slammer worm attack. Basic time series and decomposition of additive time series graphs have been used to show trend and observed UDP packets over the time frame of study.**

***Index Terms*—Code-Red, worm, SQL Slammer, network telescope, packet.**

I. Introduction

The SQL Slammer/Sapphire worm takes advantage of the MSSQL Server's remote stack buffer over flow vulnerability by scanning TCP 1433 or UDP 1434 ports after which it tries to install on the scanned server [11]. From here on the paper will refer to the worm as Slammer worm. This paper focused on traffic destined for UDP port 1434 with data gathered from two of Rhodes University's network telescopes. The UDP packets of Slammer worm are of 404 bytes in magnitude and attacks both Microsoft SQL server and Microsoft SQL desktop [11].

When a server is infected, it tries to open a back-door for the hacker and start another scanning process of infection. The SQL Slammer worm was first observed in January 2003 bringing down important system like bank's ATM service, continental Airline's ticketing and check-in systems among others [6].

The damage caused by the worm before it was stopped was estimated to be around one billion dollars [10]. After infecting its first server, the Slammer worm was doubling its victims every few seconds to a point that it managed to achieve world wide spread in roughly ten minutes [7]. Despite the fact that firewalls were present in the attacked systems, most of them could not contain the Slammer worm as they were permissive due to the worm's speed [13].

Using data obtained from two of Rhodes University's network telescope, the paper has shown the trend that has happened with UDP port 1434 since January 2009 to December 2016 with the aim of tracking the SQL Slammer worm's behaviour as it exhibits signs of reemerging into the Internet World.

Secondly, the paper shows normal time series graphs for network telescopes 146 and 196 to assess the differences in the number of packets each received from source IP addresses. It also compares the sources IP addresses from the two telescopes to identify if there were any similarities between them.

The remainder of the paper is organized as follows:

Section II explains related work done on the SQL Slammer worm, Section III explains the methodology used to carry out the analysis in this paper, Section IV provides the interpretation and results of the study. Using Rstudio for plotting, normal time series and decomposition of additive time series graphs are shown in the results section to point out the observed trend of packets.

Furthermore, Section IV shows unique top ten /24 source IP addresses that were targeting Rhodes University's network and their geolocation. This is followed by Section V which offers conclusion of the paper and closes the paper by offering future potential work that will be done as a follow up to the current study.

II. Related Work

Once a system is infected by SQL Slammer worm, the most obvious symptoms of infection are escalation of network traffic and the infected machine slows down most of the time to the point that it is barely functional [9]. While on the infected machine, instead of introducing new processes, it masquerades itself under sqlserver.exe and it make the infected machine demand more resources than it would need under normal circumstances [7].

Being a worm that uses random probing of IP addresses, Slammer worm initially spreads exponentially, however the rapid infection of new hosts becomes less effective as the worm spends more effort retrying addresses that are either already infected or immune, in which case its effect is not visible as it was in its initial phase [1]. Past work done on the worm reveal that a "Slammer worm life cycle" was highly

likely to happen again, just as it was with the case of Code-Red worm, however, the time frame was not specified [11].

Unlike Code-Red worm, Slammer worm did not contain any payload, instead it overloaded networks ensuring that servers are out of operation i.e. it did not write on the disk rather remains in memory [5]. Slammer worm spreads by randomly scanning a wide scope of IP addresses first, after which, it identifies the most vulnerable ones from its initial selection on which most of the exploitation occurs [7]. This paper has a case study of this scenario where the initial scope of IP addresses being infected was wide but later on with time the worm focused on a few IP addresses where more packets were observed.

What made SQL Slammer worm harder for administrators to contain it was its rapid speed of spreading across networks and systems. Considering that SQL Slammer worm is limited by the availability of bandwidth then it follow without loss of generality that fast paced Internet connections are more likely to accelerate the spread of the worm [13].

Recent events, dating back to November 28 and December 4, 2016, have shown that SQL Slammer worm attacks may have resurfaced, affecting 172 countries across the globe, with United States of America being the one heavily affected [9]. The source IP addresses responsible for initiating the largest number of attempted attacks were geographically located in China, Vietnam and Mexico, with no one ready to take credit for the re-emergence of the worm [10].

Its reemergence now, 14 years later, makes it one of the most long-lived threats. In December 2016, researchers at Check Point confirmed that Slammer worm is back online targeting the same ancient flaw in Microsoft SQL server 2000 buffer overflow vulnerability [8]. Whether the Slammer Worm is back to stay for good remains a question to be answered at this point.

III. Methodology

In this section, the paper explain the model used to validate the traffic that was traced from the spread of SQL Slammer worm, detail the trace collection methodology, describe approaches for characterizing the type of hosts infected and their geographic location.

A. Epidemic Model

A computer worm that randomly scans new hosts to infect new ones often times follow the simple epidemic model borrowed from biological epidemiology model [2]. Working with the assigned parameters of the model, it (the model) assumes that given a population of constant number of hosts (Q hosts) are initially all vulnerable but uninfected except for a small number that are infected and contagious, then these susceptible and infected hosts mix randomly, with an infection parameter β characterizing the rate of infection between susceptible-infective pairs.

Once infected, a host remains permanently infected i.e the model does not give room for recovery or a host being taken off the network during the epidemic's timescale. More realistically, certain hosts might be invulnerable to infection like when a computer is off for example, but those cases are simply discounted from the population of interest. Figure 1 shows a summary of how random scanning worm epidemic model works.

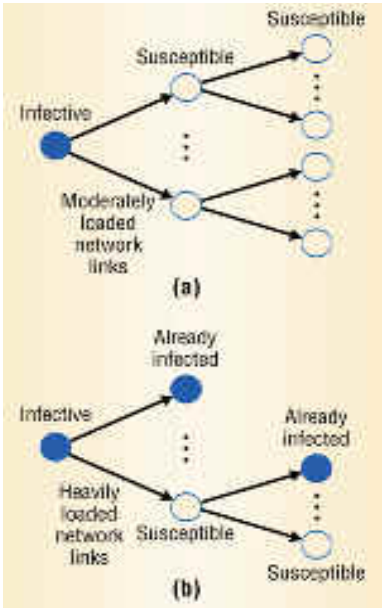

Figure 1. Random-scanning worm epidemic.(a) In the early phase, an infected host scans likely susceptible, which in turn scan other susceptible hosts, leading to exponential growth. (b) In the later phase, the epidemic slows down due to inefficient scanning and network congestion [1].

B. Data Collection and Analysis

The data used for the analysis of the SQL Slammer worm covered a period between January 1, 2009 and December 31, 2016. The time-frame was chosen because ti give ample time to do an analysis on the behaviour of the worm but also it is further away from the initial attack done in 2002. Data used for this study was collected from two /24 network telescopes i.e. network telescope 146 and 196.

Hosts (source IP address host) were considered to be infected if they sent at least one hundred UDP packets on port 1434 to none existent hosts on these networks during this time period. The requirement of one hundred packets helps to eliminate random source denial-of-service attacks other than the Slammer worm. Normally, the attacker sends UDP packets, typically large ones, to single destination via port 1434 with the aim of saturating the Internet pipe [12].

Using Wireshark in analysing the packets, the study was able to identify the the content of the SQL Slammer worm payload which agree with what was found in [3]. Figure 2 shows the data content of the SQL Slammer worm payload.

Using the simple epidemic model explained in Figure 1 we observed the IP addresses that were being targeted in January 2009 to December 2016. In the process taking into consideration the number of unique source IP addresses and number of packets being sent by each IP address. This was done by running ***tshark*** and ***tcpdump*** scripts that were specifically designed to extract unique source IP addresses and UDP packets that were passing through port 1434. Using ***tracereport*** *[4],* we were able to produce a report of the time of the first packet and last packet of the trace, the duration of the trace, total packets, and average number of packets per second.

Data (376 bytes)

```
0000 04 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
0010 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
0020 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
0030 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
0040 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
0050 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
0060 01 dc c9 b0 42 eb 0e 01 01 01 01 01 01 01 70 ae
0070 42 01 70 ae 42 90 90 90 90 90 90 90 90 68 dc c9
0080 b0 42 b8 01 01 01 01 31 c9 b1 18 50 e2 fd 35 01
0090 01 01 05 50 89 e5 51 68 2e 64 6c 6c 68 65 6c 33
00a0 32 68 6b 65 72 6e 51 68 6f 75 6e 74 68 69 63 6b
00b0 43 68 47 65 74 54 66 b9 6c 6c 51 68 33 32 2e 64
00c0 68 77 73 32 5f 66 b9 65 74 51 68 73 6f 63 6b 66
00d0 b9 74 6f 51 68 73 65 6e 64 be 18 10 ae 42 8d 45
00e0 d4 50 ff 16 50 8d 45 e0 50 8d 45 f0 50 ff 16 50
00f0 be 10 10 ae 42 8b 1e 8b 03 3d 55 8b ec 51 74 05
0100 be 1c 10 ae 42 ff 16 ff d0 31 c9 51 51 50 81 f1
0110 03 01 04 9b 81 f1 01 01 01 01 51 8d 45 cc 50 8b
0120 45 c0 50 ff 16 6a 11 6a 02 6a 02 ff d0 50 8d 45
0130 c4 50 8b 45 c0 50 ff 16 89 c6 09 db 81 f3 3c 61
0140 d9 ff 8b 45 b4 8d 0c 40 8d 14 88 c1 e2 04 01 c2
0150 c1 e2 08 29 c2 8d 04 90 01 d8 89 45 b4 6a 10 8d
0160 45 b0 50 31 c9 51 66 81 f1 78 01 51 8d 45 03 50
0170 8b 45 ac 50 ff d6 eb ca
```

Figure 2. SQL Slammer payload

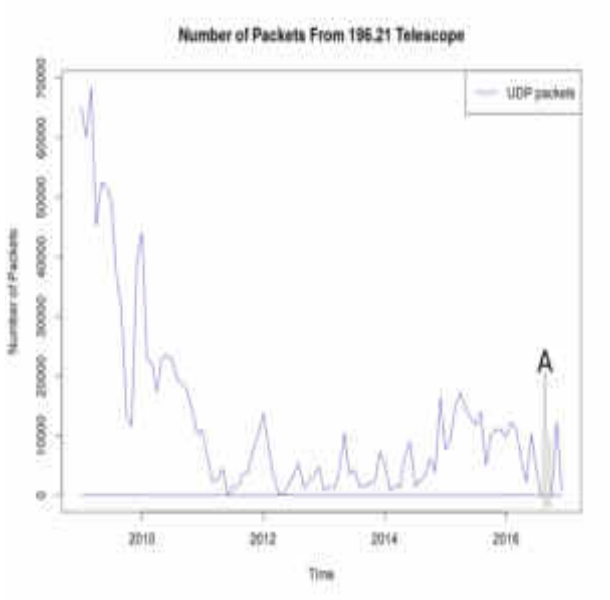

Figure 3. Pattern and trend of UDP packets from 196

Table I
TOP TEN /24 NETBLOCK IP ADDRESS BLOCKS AND THEIR GEOLOCATION

Position	/24 Netblock	%	Country of Origin
1	216.218.206.X	39.10	United States (US)
2	122.225.100.X	13.78	China (CN)
3	61.145.123.X	6.24	China (CN)
4	218.204.137.X	6.18	China (CN)
5	218.30.22.X	6.00	China (CN)
6	218.23.37.X	5.23	China (CN)
7	212.252.124.X	4.72	Turkey (TR)
8	71.6.216.X	4.29	United States (US)
9	202.99.11.X	4.28	China (CN)
10	218.75.199.X	3.02	China (CN)

For plotting purposes and ease of use, the output of the scripts was converted to .csv files so that ***python*** and ***Rstudio*** manages to read them for further analysis. We also used **geoiplookup** to determine the latitude, longitude, and country of origin of each of the top ten source IP address infected with the worm. Geoiplookup uses public data sources such as WHOIS and DNS, as well as specialized measurement to geographically place IP addresses.

IV. RESULTS AND DISCUSSION

In this section of the paper, we present the results of our trace analyses and discuss their meaning. We first looked at the pattern and trend observed in UDP port 1434 using decomposition of additive time series graphs and normal time series graphs, then identify the top ten /24 unique source IP addresses together with their geolocation.

The two network telescopes combined detected more than 14,400 unique /24 source IP addresses coming from different places across the globe, all of these were of /24 subnet mask. The study looked at the top unique IP addresses and their geolocation.

Table II
NUMBER OF PACKETS UDP PACKETS RECEIVED BY NETWORK TELESCOPE 196 AND 146

Telescope	# of packets	%
196	46,840,867	51.24
146	44,570,578	48.76
Total	91,411,445	100

Table I shows top /24 IP address blocks and their geographic location. shows top /24 IP addresses, percent contribution and their geographic location. These top ten netblock IP addresses sent out about 35.227% of the total packets that were received by the two network telescopes with netblock 216.218.206.X being the highest sender and appearing over a period of 24 months i.e appeared only from January 2015 to December 2016. Source IP address netblock 71.6.216.X appeared more than any other address, i.e from August 2014 to December 2016. 216.218.206.X contributed more than the others because it appeared in both telescopes while the other netblock IP addresses appeared in one of the two telescopes and for less that seven months.

As it can be seen from Table I, out of the ten IP addresses present, seven of them came from China, however these did not come from one city. The top IP address blocks from China and the other one from Turkey were the ones that were dominant between 2009 and 2010, there after they received very minimal packets and never resurfaced again. On the other hand, the two from United States were the ones responsible for the uprise from mid 2014 to end of 2016. They contained roughly 70% of the total packets that were received by the network telescope.

A. Network Telescope 196 vs 146

Table II shows two telescopes used for this research. With a total of 91,411,445 UDP packets received, 44,570,578 were received by 146 and 46,840,867 were captured by 196. Figure

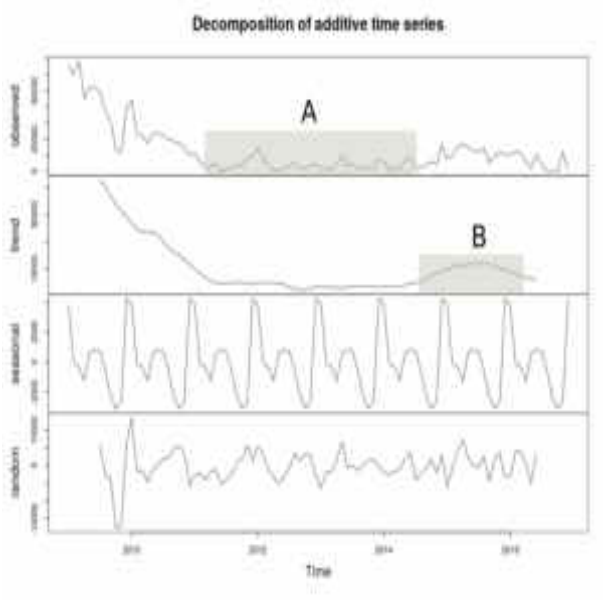

Figure 4. Decomposition of additive time series graph with packets received by network telescope196

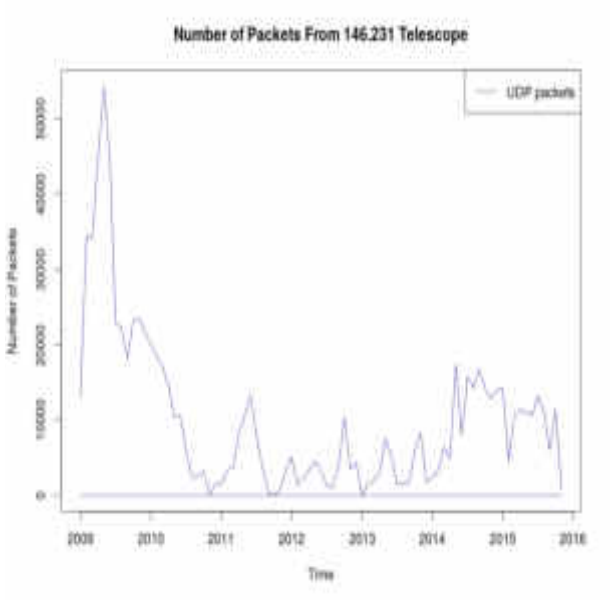

Figure 5. Pattern and trend of UDP packets received by network telescope 146

3 shows a time series graph that was plotted using data collected from 196 network telescope. During the early stages of data collection (2009 - 2011), there was a wide range of unique source IP addresses (roughly 400) collected by the telescopes. About 17% of these received roughly 1000 packets per month while the rest of the 83% received less than 20 packets per month. This wide range of unique IP addresses caused the huge spike shown if Figure 3.

However, the top 10% contributed about 90% of the total packets received in each of the first two years. This is what contributed to the high numbers at the beginning of the 2009 till end of 2011. In line with epidemic model, the number of packets started to reduce just as the range of unique IP address did, from 400 unique IP addresses in 2009 to about 70 unique IP addresses in 2016. This is due to the fact that at this time, no new IP addresses were registered and with the few IP addresses that were present, they sent packets at a very low rate.

At times, no packet was sent to the telescopes as shown in Figure 3 where the graph touches the horizontal line. The point where the graph touches the horizontal line are instances where zero packets were recorded by the telescope. Figure 3 (Section labeled A) also show that between June to October 2016 there were no packets recorded hence the gap that is seen in the graph. However it picked in November to December in the same year just as Check point researchers observed with their telescopes.

Figure 4 shows both the observed and trend layers of packets collected by 196 network telescope. From Figure 4, it can be seen that between 2011 and 2014 (shown by the section labeled A on Figure 4) the number of packets received were very low as compared to any pointing time of the time frame. However when a closer look is made on Section B of the same figure, there is an unexpected rise in the number of packets matching the pattern observed in telescope 146.

The data collected shows a significant reduction of both the unique source IP addresses and the number of packets received by the IP addresses. The random and seasonal components have not been discussed because they have been consistent and there were no significant changes apart from the beginning of 2009 which marks the initial data collection point.

The highest number of packets sent by the IP addresses was roughly 200 packet per month. This is the same too for network telescope 146 shown in Figure 6 where the number of packets received began to take a decline from end of 2010, with a steady increase to about 12,000 packets mid 2011. However after that, it unsteadily declines until end of first quarter of 2014 (shown by the section labeled A on the graph) where source IP addresses 216.218.206.X and 71.6.216.X first appeared and made a steady increase from June 2014 to end of December 2016.

B. Hypothesis being tested

The hypothesis that was being tested at the beginning of this study was that, we expected a smooth decline in the number of packets received by the two network telescopes. However, as it can be observed in Figures 4 and 6, specifically looking at the trend and observed traffic layers in these two graphs, this is not the case. From 2014 onwards (see Section Labeled B on Figure 6), there is an unexpected rise in the number of packets received by both telescopes. This, as explained earlier in this section, is a result of the new IP addresses that were introduced and unlike the previous years, a new geolocation is responsible for this rise. It is because of this new rise in the number of packets that is making the authors of this paper believe that SQL slummer worm might be coming back into the Internet. What makes this more interesting is the fact that the number of packets sent by each of these unique /24 source IP addresses is the same for each year, a pattern which was not seen in the early stages of data collection.

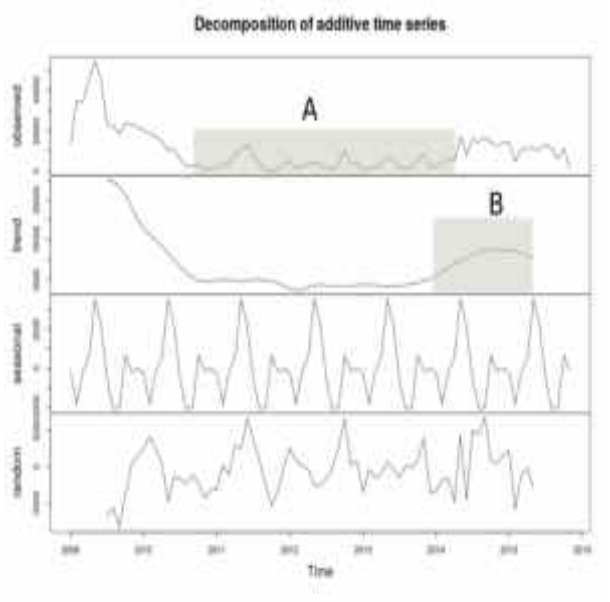

Figure 6. Decomposition of additive time series graph with packets received by network telescope 146

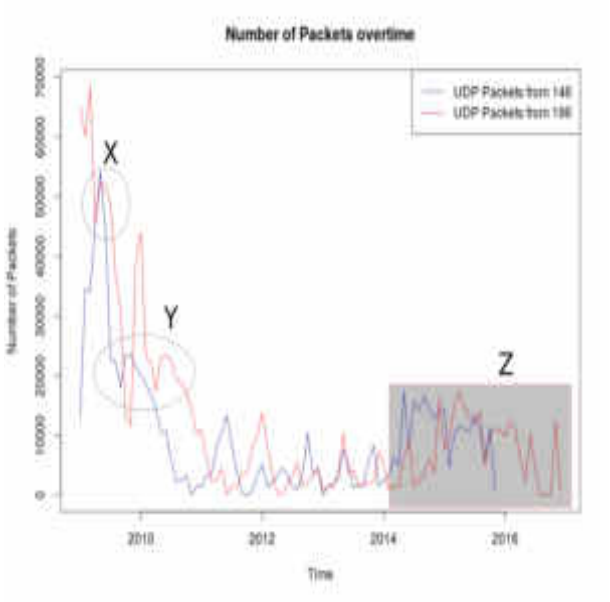

Figure 7. Pattern and trend of traffic received by network telescope 196 and 146

Figure 7 shows a comparison of two of the network telescopes that were used for this study. From Figure 7, it can be seen that network telescope 196 received more packets than 146. This is the case because of two reasons: from the beginning, January 2009, telescope 196 received more packets from the same IP address than telescope 146. Secondly, as it can be seen from Figure 7, network telescope 196 had more data collection time than 146. We also noted the similarities in the number of packets received by both telescopes and the lagging shown in Figure 7 shown in Sections X, Y and Z respectively.

V. Conclusion

Looking at the data and results that we have, it shows that SQL Slammer worm may be making its way back into the Internet world. The expected curve for both telescopes was expected to take a continuous decline from 2011 onwards however the results show otherwise. Focusing on network telescope 196, which expand all the way to December 2016, it shows another increase in the number of packets it received especially from November heading towards December. This agrees with the results shown by Check Point Research that between 28 November and 4 December a number of countries were hit by the worm after 14 years of silence.

The study also confirms with one notion pointed by Check point that a good portion of the source IP addresses that are responsible for the spread of SQL Slammer worm came from China, this is shown in Table I. However recent events in the two telescopes show that the recent uprise in the number of packets received by the two telescopes was coming from USA who in the recent events are the victims.

From the data that we have it shows that the worm has been active from 2014 onwards, except that it was not attracting international attention at that time as it has done end of 2016. The epidemic model proves that the number of infected hosts tends to decline over time and that the most prone hosts become primary targets for future attacks as the number of packets sent through them increase while the number of IP addresses decrease.

A. Future Work

Having similar results to those check point researchers have shown from the months of November and December 2016, to confirm the coming back of Slammer worm, the authors will continue to collect more data for this year and run the check again to see if the worm is indeed here to cause more harm as it did in 2003. Using the epidemic model the researchers of this paper will expand to test and compare results with port 1433 TCP of which little has been written in line to the spread of SQL Slammer worm.

The paper will also explore more on the similarities and lagging shown in Figure 7, particularly in the sections labeled X, Y and X. Initial analysis also shows that there is more than the SQL Slammer worm payload associated with the rise shown in Section Z of Figure 7 as such a further study will be done explore the identified payload associated with destination port 1434/UDP.

Acknowledgements

The authors would like to express their gratitude for the financial support it received from Beit Trust, Telkom SA, Tellabs/CORIANT, Easttle, Bright Ideas 39 and THRIP. This work was undertaken as part of Distributed Multimedia CoE at Rhodes University, however, the authors acknowledge that opinions, findings and conclusions or recommendations expressed here are those of the author(s) and that none of the aforementioned sponsors accept liability whatsoever in this regard.

REFERENCES

[1] T.M. Chen and J-M. Robert. Worm Epidemics in High-speed Networks. *IEEE Computer Society*, 37(6):48–53, June 2004.

[2] D.J. Daley, J. Gani, and J.M. Gani. *Epidemic Modelling: An Introduction*, volume 15. Cambridge University Press, 2001. ISBN: 9780521014670.

[3] H. Dongmei. Use offense to inform defense. Find Flaws Before the Bad Guys do. Case study:Attack of Slammer Worm. *SANS Institute*, 2003.

[4] Edgewall. Trac - Integrated SCM & Project Management, January 2017. [Date Accessed online: 10 March, 2017] Available: https://trac.edgewall.org/.

[5] D. Forte. Slammer - The Return of the Network Nightmare. *Network Security*, 2003(2):17 – 18, 2003.

[6] M. Lad, X. Zhao, B. Zhang, D. Massey, and L. Zhang. Analysis of BGP Update Surge During Slammer Worm Attack. In *International Workshop on Distributed Computing*, pages 66–79. Springer, 2003.

[7] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. *IEEE Security & Privacy*, 99(4):33–39, 2003.

[8] P. Paganini. The Slammer Worm is Back After 13 years to Target Ancient SQL Servers , February 2017. [Date Accessed Online: 23 March, 2017]. http://securityaffairs.co/wordpress/56028/malware/slammer-worm-2017.html.

[9] D. Palmer. SQL Slammer Worm Comes Back from the Dead After a Decade of Inactivity, February 2017. [Date Accessed Online: 23 March 2017]. http://www.zdnet.com/article/sql-slammer-worm-comes-back-from-the-dead-after-a-decade-of-inactivity/.

[10] D. Pauli. Slammer Worm Slithers Back Online to Attack Ancient SQL Servers, February 2017. [Date Accessed online: 23 March 2017]. https://www.theregister.co.uk/2017/02/05/sql_slammer_back/.

[11] E. Schultz, J. Mellander, and D. Peterson. The ms-sql slammer worm. *Network Security*, 2003(3):10 – 14, 2003.

[12] A. Singh and D. Juneja. Agent Based Preventive Measure for UDP Flood Attack in DDoS Attacks. *International Journal of Engineering Science and Technology*, 2(8):3405–3411, 2010.

[13] C.C. Zou, L. Gao, W. Gong, and D. Towsley. Monitoring and Early Warning for Internet Worms. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 190–199. ACM, 2003.

Stones Dalitso Chindipha is currently studying towards his PhD at Rhodes University in the Security and Networks Research Group (SNRG). His research interest include network security, passive security monitoring, cyber security, Internet Background Radiation, data analytics and data visualisation.