National College of Ireland

Efficient Detection Of Malware Beaconing

MSc Internship
Cybersecurity

Shairin Gomes
Student ID: x18108211

School of Computing
National College of Ireland

Supervisor: Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	SHAIRIN GOMES		
Student ID:	X18108211		
Programme:	MSC IN CYBERSECURITY	**Year:**	2018-2019
Module:	ACADEMIC INTERNSHIP		
Supervisor:	IMRAN KHAN		
Submission Due Date:	12/07/2019		
Project Title:	EFFICIENT DETECTION OF MALWARE BEACONING		
Word Count:	4664	**Page Count**	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………

Date: ………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not enough to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

MSc Internship in Cyber Security

Shairin Gomes
X18108211

Abstract

Several attempts have been made towards implementing strong and protective firewalls on the network system as well as on the host systems, however, the traceability of the malware remains undetected in instances when the same lies in the lower layers of the network security. With respect to malwares, beaconing is the behaviour that encompasses of the practice of sending short as well as regular communications to an attacker or controlled host, from the infected host, for communicating that the infected host malware has been activated, and the same is alive as well as functioning, thereby ready to take the instructions. Beacons are considered as one of the suitable means for the detection of the malwares in the network security and therefore, focus has been made towards the same. The current study had employed the use of PSO and AES techniques for the detection of the malware beaconing to determine the threats within the network security. A model has been proposed based on these two techniques and the same was implemented to determine the efficacy of the proposed model. Based on the results that have been acquired from the tests that were conducted using the proposed model, it can be concluded that the utilisation of models based on AES and PSO techniques are indeed effective and therefore, can be used for the detection of the malware beaconing in any network security during cyber-attacks, so that the necessary steps can be taken for overcoming the issue.

Keywords : Malware beaconing, PSO, AES, DSR and beacon signals.

1 Introduction

The advancements that have taken place in the area of digital technologies and cyber world, there has been an emergence in the cyber-attacks that render the network system vulnerable to various susceptibilities. The brutal and extensive attacks that have been taking place on the networks have been becoming great concern as the same have been damaging the cyber security. Several attempts have been made towards implementing strong and protective firewalls on the network system as well as on the host systems, however, the traceability of the malware remains undetected in instances when the same lies in the lower layers of the network security. At other times, the malware utilises the TCP/IP stacks for crossing over the firewalls. The TCP is interrupted by the adversaries as the same establishes a connection between two different computers with the help of 3-way handshake. Furthermore, after securing a reliable communication, the cyber attacker can initiate a DDoS attack on the network server, thereby disrupting the entire system. With respect to malwares, beaconing is the behaviour that encompasses of the practice of sending short as well as regular communications to an attacker or controlled host, from the infected host, for communicating that the infected host malware has been activated, and the same is alive as well as functioning, thereby ready to take the instructions (Vijayakumar and Ganapathy, 2018).

Beacons have been determined to often originate from the infected internal organisational hosts such as zombies or bots and the same are sent to the control and command (C&C or C2) servers that are situated outside of the corporate network. The current research study has focused towards the identification of the efficient means that would aid in effective detection of the malware beaconing so that the network security can be established and improved. This is a nice introduction with a figure presented in Figure 1.

2 Related Work

Various studies have been conducted in the area of detecting malware in network. Different approaches have also been developed to effective detection of the malware that are present in any network. The following section has focused towards the determination of the various aspects that are associated with the malware detection processes and their successes, with respect to the studies that have already been conducted in these aspects.

2.1 Detecting the malware behaviour

One of the leading issues associated with the cyber security of the modern times is the emergence of the targeted attacks which have been conducted by the adversaries who have the access to various sophisticated tools. It has been mentioned by Shalaginov, Franke, and Huang (2016) that by utilisation of the common behaviour of the malware, known as "beacon", that fundamentally implies the manner in which the infected host communicates with the Control and Command servers at the regular time intervals of relatively small time variations, as well as analysis of such beacons and its activities through the passive network monitoring, the detection of the prospective malware infection is achievable (Shalaginov, Franke and Huang, 2016) . It has been proposed by Fehrman (2018) that the detection of the malware beaconing through an effective method comprises of detection of the same in a network through the capture of the network traffic that represents the sets of tuples and each of these tuples is in turn an Open Systems Interconnection layer 4 that comprises of defined, IP addresses, port addresses and destination IP, while observing the frequency of the connection. Furthermore, Fehrman (2018) also proposed of a machine that has the network connection, with non-transitory machine of readable storage medium, along with a hardware processor that connects with the both (Patents.justia.com, 2018).

2.2 DNS on target to detect the malware beaconing

It has been mentioned by Shalaginov, Franke, and Huang (2016) that through the evaluation and event correction of the DNS log, that is achievable through concentration on the high latency and low latency of the malware beaconing, any kind of threat to the security can be determined. It has been assumed by Shalaginov, Franke, and Huang (2016) that in the low latency, the time interval of the beacon is short and the same results in the infected hosts to frequently communicate with the C2 server (that is less than 3) every day. The investigations of Shalaginov, Franke, and Huang (2016) was based on the data that were generated in a day. Furthermore, the low latency is detectable on enterprise network as well as in single hosts. In case of the high latency, it has been considered by Shalaginov, Franke, and Huang (2016) that a malware is detectable as the same would communicate at a fixed time daily and the same would be communicated to the C2 at the later stage. As a result of the same, the analysis of the high latency beacons is achievable every week as evidenced in the works of Zoldi, et al. (2016) (Shalaginov, Franke and Huang, 2016). The independency of the C2 communications method host victims can send feedback details to the C2 servers with the evidences that have been accumulated in the DNS records, through the consideration of the records that implies that the IPv4 addresses are frequently utilised. The detections that are based on the beaconing demonstrates or evidences that all the hosts that are linked with the malicious domains have been compromised or affected with the infection, while the domain that is linked with the infected hosts are present on the shortest path (Duque and bin Omar, 2015; Zhang, et al., 2018) . This path that comprises of the two hosts that are interconnected and have common vertex of domains, are classified under the “bridge domains”, as these are the rarest domains and these send the infectious emails as well as downloadable malware that can easily bypass the network security prior to C2.

2.3 Comprehensive beaconing - BAYWATCH

The Hu, et al. (2016) have mentioned that through the in-depth evaluation that was conducted by the researchers on the area of malware beacon detections, an innovative method called the BAYWATCH was identified. This is a comprehensive filtering method that evaluates the beaconing attributes of the network while communicating with another network (Hu et al., 2016). The algorithm detection design that was developed was done on the foundation of signal processing and these were captured at different regular time intervals or scales, further experimented by Hu (2018). As a result of the same, the BAYWATCH lists the different cases of beaconing from the severity of high to low. The results that were analysed by Hu, et al. (2016) were associated with 26 suspicious beaconing that were observed daily and of these identified beacons, about 96 per cent of them were the topmost malicious ones. For acquiring more integrated data or details regarding the efficacy of BAYWATCH in the real-world network and data sets, Hu, et al. (2016) performed detection of beaconing over the various web proxy logs that were gathered within the corporate networks. The entire process required 35 hours for processing the log that were accumulated over a month, during the weekdays, across the 3.3 million network links which were set up on an average of estimated 14 minutes, which were taken for evaluation. For the weekends, the logs were acquired for 26 million of such connection combinations and the evaluation procedure required about one and half hours. Through the results that were acquired from those evaluations, it was evidenced that BAYWATCH indeed has the capability of evaluating nearly millions of beacon activities, every day (Hu et al., 2016).

2.4 Optimization technique over malware

A particle swarm optimisation (PSO) algorithm utilises the maximum flow objectives for selecting the optimal locations of the agents during every time step of the network operation. The researchers Dengiz, Konak and Smith (2011) had stated that the selection of the particle swarm optimisation (PSO) for the dynamic mobile ad hoc network (MANET) can be considered as the heuristic approach as the same solves the issues that are associated with the environmental changes, with respect to locating and tracking the packets as well as in the population based meta algorithm. Dengiz, Konak and Smith (2011) had proposed towards enhancing the network connectivity towards the dynamics of MANET by maintaining the control of the network. The implementation of the approach enhances the performance of the MANETs. As per Dengiz, Konak and Smith (2011), MANET management system has the ability of improving the network connectivity with the use of controlled network nodes, named as agents. These agents have the capabilities of wireless communications with the other nodes that exists in MANET. However, the movements of these nodes and therefore, their locations, are fundamentally dynamically determined for optimising the network connectivity. From the computational results, it has been determined that the proposed approach is efficient in enhancing the connectivity of the MANETs as well as in predicting the movements of the user nodes, as the deploying agents critically enhances the overall performance of the MANET. The representation of the wireless ad hoc network communications as the network flows as well as optimisation, by the utilisation of a maximum flow model is considered as an advantageous and novel approach. Furthermore, this can also be utilised with any signal attenuation model while calculating the data flow rates (Dengiz, Konak and Smith, 2011). However, the dynamic nature of the network problem is considered as a challenge, however, the same also facilitates the optimiser in gaining additional information by leveraging the information that have been acquired during the process of optimisation (Du and Swamy, 2016). The significance of predicting the user location for enhancing the network connectivity have been well demonstrated by Dengiz, Konak and Smith (2011) and the same can be utilised for the detection of malwares through beacon nodes (Dengiz, Konak and Smith, 2011). The effective approach by Zafar Ali and Tahir Rahim Soomro et al. (February 2018) described that the mining can help to detect hidden malwares. PSO is efficient in filtering out the redundant and irrelevant packets, due to this feature of PSO their aim was to reduce the 9 Portable Executable packed and unpacked files and using PSO will give the accurate result and reduce the processing time.

2.5 Software Defined Firewall for TCP/IP

Malwares based on networks have been posing serious threats on the security of the host machines. Whenever the malware adopts any private TCP/IP stack for the purpose of communication, the network and personal firewalls at times fail in identifying the malicious traffic (AlEroud and Alsmadi, 2017). The existing firewall policies do not possess any convenient update mechanism that further makes it difficult to detect the malicious traffic (Gao, et al., 2018). The software defined firewall (SDF) that have been proposed by Gao, et al. (2018) is fundamentally a novel security design that protects the host machines while enabling the programmable security policy control through abstraction of the firewall architecture into data planes and control. The control plane fortifies the easy security control policy update that also exists in the software defined networking (SDN) architecture. The difference lies in the aspect that the SDF further gathers the host information for providing application level traffic control as well as for improving the accuracy of malicious traffic detection. The SDF has been designed such that the same can easily be implemented as well as deployed in the modern network. Furthermore, Gao, et al. (2018) had implemented a prototype of SDF and evaluated its performance in several experiments of real world. A novel architecture of firewall that comprises of the data planes and controls, were abstracted within the software defined networking architecture. Furthermore, it also has the capacity of detecting the malware traffic every time it is utilised as a private TCP/IP stack that is used for bypassing the traditional firewalls (Gao, et al., 2018).

3 Research Methodology

By maintaining the section 2.5 “Novel Approach in Software Defined Firewall” into consideration, the research study have taken the detection techniques in the detection model, that is Flow Pool, followed by Signature Matching, followed by SVM classifier along with the novel proposal of the detection model with the addition of the matching algorithm with the help of flow pool that would enhance the set of restriction and rules for the packets. The replacement of the signature matching along with the AES encryption and the decryption standards because the model that have been proposed was not as effective with the cryptography standards that is required to be taken highly into consideration so as to protect the network security from any kind of data leakage, as per the CIA rule. The classical mining approach SVM classifier has been replaced with the Particle Swarm Optimisation technique because the SVM does not have the ability of performing the matching of the pattern as well as is unable in detecting the new malware that have been released by the adversaries as evidenced in the works of (Hossain et al., 2019). Along with the proposed model, is the beacon signal to capture the malware beaconing activities. The current research study had attempted towards developing an efficient system for detection of the malware beaconing and the same have been discussed in the malware detection methodology, whereby the detection have been made in two phases for identifying the pattern or the behaviour of the malware. The first phase of the detection comprises of the packet level classification that includes the following, as discussed.

3.1 Matching in the flow of network

The flow pool is fundamentally a technique that detects the packets and thereby classifies the same on the basis of the header field that are available on the header parser and thereby adding the matching algorithm along the sender IP address, counters, priorities, fields, cookies, timeouts, instruction and the packet sizes, whereby they are differentiated between the malicious packets and benign packets with the bit size differences. The traffic is monitored based on sending, receiving, forwarding and dropping the packets of the neighbours and then further matching algorithm checks the IP address of source and the packet’s size this filtering will therefore drop the malicious packets.

3.2 Using Advanced Encryption Standard

The technique is more phenomenal as compared to the other encryption protocols and this offers the option of selecting between 128-bit, 192 bit and 256 bits key that makes it stringer as opposed to the 56-bit key that is available. With the application of the technique, the information regarding the valid keys can be acquired. It is known that a handshake connection takes place in the TCP/IP between the sender and the receiver and therefore, it is important to secure the decryption and encryption standards, such that the only reliable source is associated with sending and receiving the data and the same would aid in avoiding the attacker to attack from the middle. With the use of multiple XOR gate along with different keys, diffusion and confusion can be implemented such that the secure information does not get leaked out to the adversaries.

3.3 Optimisation Technique using Particle Swarm Optimisation

Particle Swarm Optimization (PSO), which is considered as a population-based particle algorithm, was developed by Dr Kennedy and Dr Eberhart in the year 1995. Fundamentally, “swarm” in this context refers to the various numbers of prospective solutions for the optimization issue and in this each of the prospective solution is considered to be “particle” (Bonyadi and Michalewicz, 2017). This technique is considered as an iterative method that is based on the inputs and motions. Initially, each of the particle is given preliminary random parameters and the same is allowed to flow in the multidimensional search space. During each of the progressive generation, every particle utilizes the information regarding its earlier best position along with the global best position that maximizes the probability of moving towards a better solution space, which thereby produces better fitness.

3.4 Beacon Signals

The signals that are received provides the information regarding the communication packets at constant intervals of about 1/10 of the seconds, thereby transmitting the unique ID of the device of the receiver as well as the beacons.

4 Design Specification

This section adheres the design of the implementation of model proposed, the model is basically acting as the detection of beaconing behaviour of malwares. The design has the following sections (as depicted in Figure 1) to perform the testing firstly, matching flow pool which will match according to the IP pattern set and size of the packet during the incoming in the wireless network. Secondly, PSO optimization method which calculates the earlier and global best fit position of the particles in motion. Thirdly, AES encryption protocol which is a way to secure the communication in the network this standard is considered due to its maximum size of 256-bits. Fourthly, the beaconing signals are transmitted in microseconds in simulator.

The flow of the malware detection model is as follows, the traffic data is being generated manually, and this generated traffic goes to the matching algorithm in flow pool and the packets gets filtered by shedding malicious pieces and after this the packets travel through AES which is well encrypted standardization with the key already specified in the code file of AES, then this data passes through PSO mining technique which optimizes the data by filtering he redundant and unexpected packets and efficiently gathers the global best position and to this beacon signal is added as per the speed of microseconds.

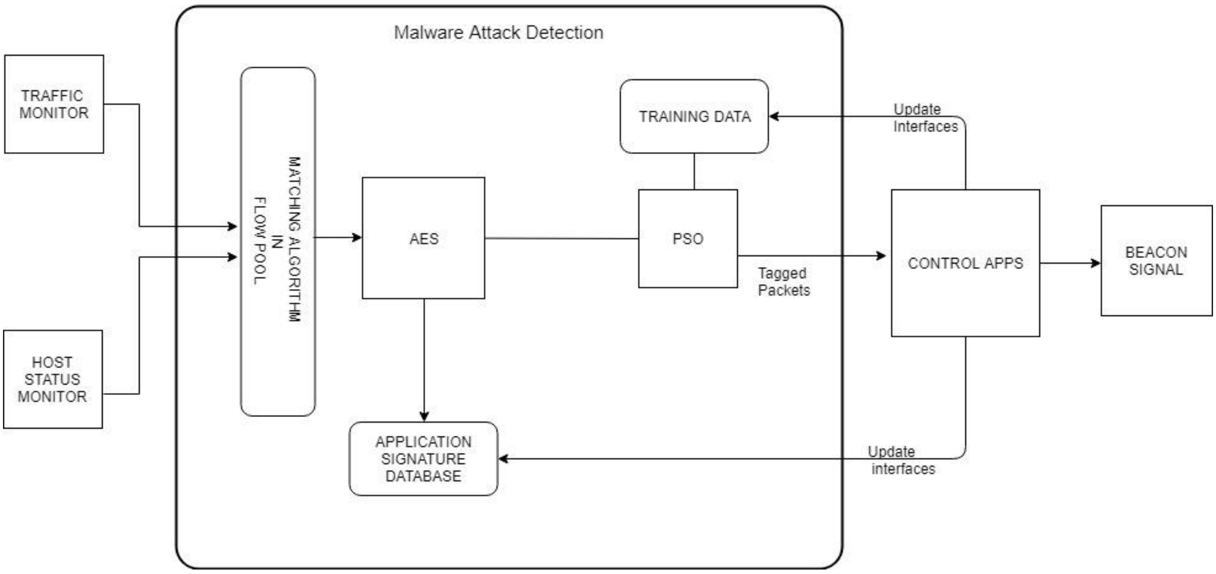


Figure 1. Proposed model for efficient detection of malware

To set up a test bed using network simulator to create a wireless network and manually creating the malicious nodes and given appropriate percentage to those nodes this will create. The AES will add the secret key implemented in the code.

5 Implementation

This section discusses the setup of testbed therefore configuring the virtual box with operating system Ubuntu 14.04 version, choice of this version is that it supported the installation of network simulator and execution of "tcl" scripts.

5.1 Software/Language and Protocol applied:

Virtual Box
The software is installed in personal desktop and this software can be used as the imitation and used as a testbed when it comes to testing the malware as it makes our system to isolated to do such testing. Virtual box is the open source successfully developed by Oracle and it loads several operating system in one platform.

Ubuntu version 14.04
Linux distribution open source which is based on Debian. It is a very popular operating system this platform will help and is adaptive in executing the file scripts for the network simulator execution. For the implementation and testing this proposal we need to focus on this version or below version 16 only, therefore leads to this selection of this version.

Network Simulator (ns2.3.4 version)
It simulates the traffic in the network, developed by two programming languages C++ and TCL. NS2 helps to build the wireless network declaring and setting the values of the basic parameters such as adhocRouting, antType, channelType, phyType, macType, macTrace, etc.

It creates the topology grid and as we are designing wireless network, we need to keep a track of wireless nodes as their distance changes time to time, therefore we create GoD object i.e. General Operations Director.

Network Animator

It is a virtual representation of traces in network simulation. Initially produces the trace files which gives the details of packet traces. The screen of NAM shows virtually the motions of the nodes while playing the animation and other features to fast forward the animation, stop the animation, and to change the step which means the timely animation between neighbouring nodes.

Dynamic Source Routing (DSR)

This routing protocol for the multi-hop in mobile nodes, this allows build own infrastructure and connects with the internet. The DSR includes two vital mechanism which allows to discover nodes i.e. route discovery and maintaining the routes to arbitrary destination i.e. route maintenance in the wireless network (Cs.cmu.edu, 2019).

TCL (Tool Command Language)/ C++

For executing the simulator TCL command is used to write the scripts. Executable in all the platforms, creation of John Ousterhout. The syntax of this language is simple, cross-platform, robust, BSD license.

C++ is crossed platform and popular language developed by Bjarne Stroustrup and therefore due to its efficiency it makes high performances applications and adaptive to multiple platforms.

5.2 Setup and coding in NS2.3.4

Configure “ns-allinone package” as the name depicts it supports all types of packages this is only supported in Unix platform. To start the ns, we need to execute “main.tcl” file. This file will produce output.tr and output.nam

The main file constitute of build-up of wireless network is in “main.tcl” script which has configuration of antenna, wireless channel, routing protocol- DSR number of neighbouring nodes and this file will correspondingly produce after the execution of it in two types of output files as output.tr and output.nam and AES cryptography. In the setup for training dataset in a way (training dataset cannot be done in simulator) we can monitor the data based on packet details of each nodes, match the values and deciding based on threshold as malware or benign.

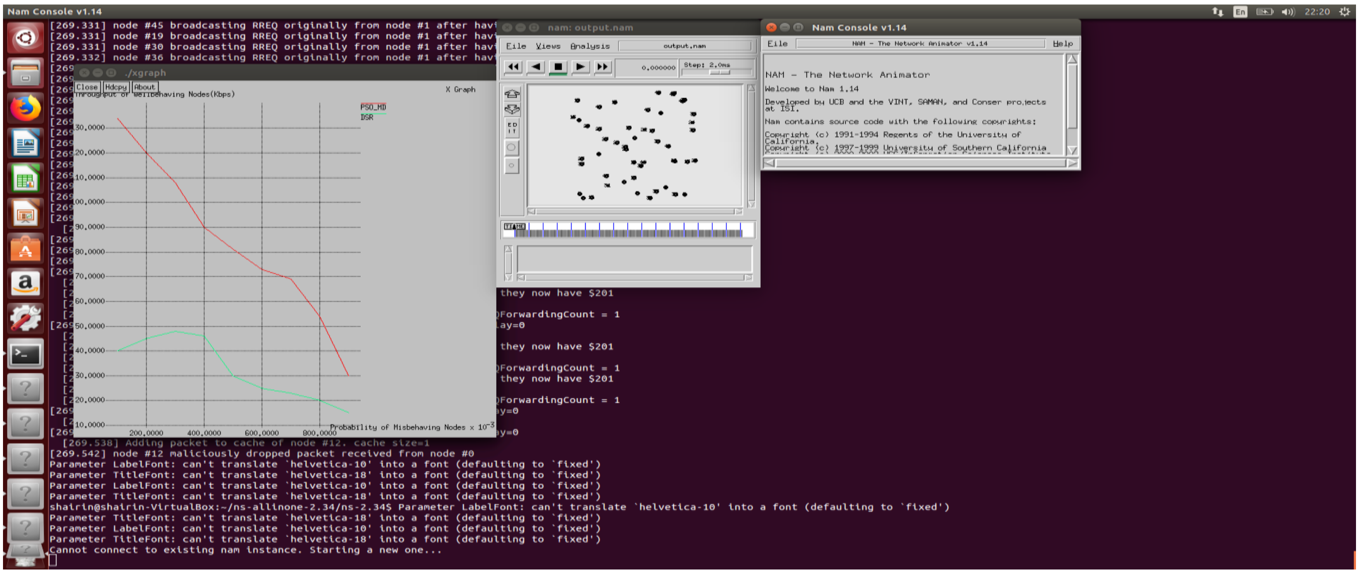
“flowstruct.c” depicts the matching flow as per the rule set for filtering and processing them. Manually setup of attacking with varied percentages of attacks (like one file has 4% of attacking etc.) and to increase the attack we can again configure “attacker.txt”

The DSR file is containing PSO, beacon and matching signals and the attacker.txt file which enables the attacking and produce the resultant data in output, renaming the “attacker.txt” file will disable the attack and will give resultant data as out. The incoming packets monitored based on source, destination and packets are dropped and forwarded.

The result of the design of this implementation will give graph as a result of the cases considered which will be discussed in the below section.

6 Evaluation

The execution of “main.tcl” code (and described in above section) will produce the output on the screen as shown in Figure 2. One is Network animator (Figure 3) which will show the movements of nodes (Figure 4) and second is graphical representation which is being discussed in further section of evaluation.



```
[269.331] node #45 broadcasting RREQ originally from node #1 after havi
[269.331] node #19 broadcasting RREQ originally from node #1 after havi
[269.331] node #30 broadcasting RREQ originally from node #1 after havi
[269.332] node #36 broadcasting RREQ originally from node #1 after havi
```

```
 they now have $201

QForwardingCount = 1
lay=0

 they now have $201

QForwardingCount = 1
 they now have $201

QForwardingCount = 1
ay=0

ay=0
```

```
  [269.538] Adding packet to cache of node #12. cache size=1
[269.542] node #12 maliciously dropped packet received from node #0
Parameter LabelFont: can't translate `helvetica-10' into a font (defaulting to `fixed')
Parameter TitleFont: can't translate `helvetica-18' into a font (defaulting to `fixed')
Parameter LabelFont: can't translate `helvetica-10' into a font (defaulting to `fixed')
Parameter TitleFont: can't translate `helvetica-18' into a font (defaulting to `fixed')
shairin@shairin-VirtualBox:~/ns-allinone-2.34/ns-2.34$ Parameter LabelFont: can't translate `helvetica-10' into a font (defaulting to `fixed')
Parameter TitleFont: can't translate `helvetica-18' into a font (defaulting to `fixed')
Parameter LabelFont: can't translate `helvetica-10' into a font (defaulting to `fixed')
Parameter TitleFont: can't translate `helvetica-18' into a font (defaulting to `fixed')
Cannot connect to existing nam instance. Starting a new one...
```

```
NAM - The Network Animator

Welcome to Nam 1.14

Developed by UCB and the VINT, SAMAN, and Conser projects
at ISI.

Nam contains source code with the following copyrights:

Copyright (c) 1991-1994 Regents of the University of
California.
Copyright (c) 1997-1999 University of Southern California
```

Figure 2. Output after executing input.tcl file

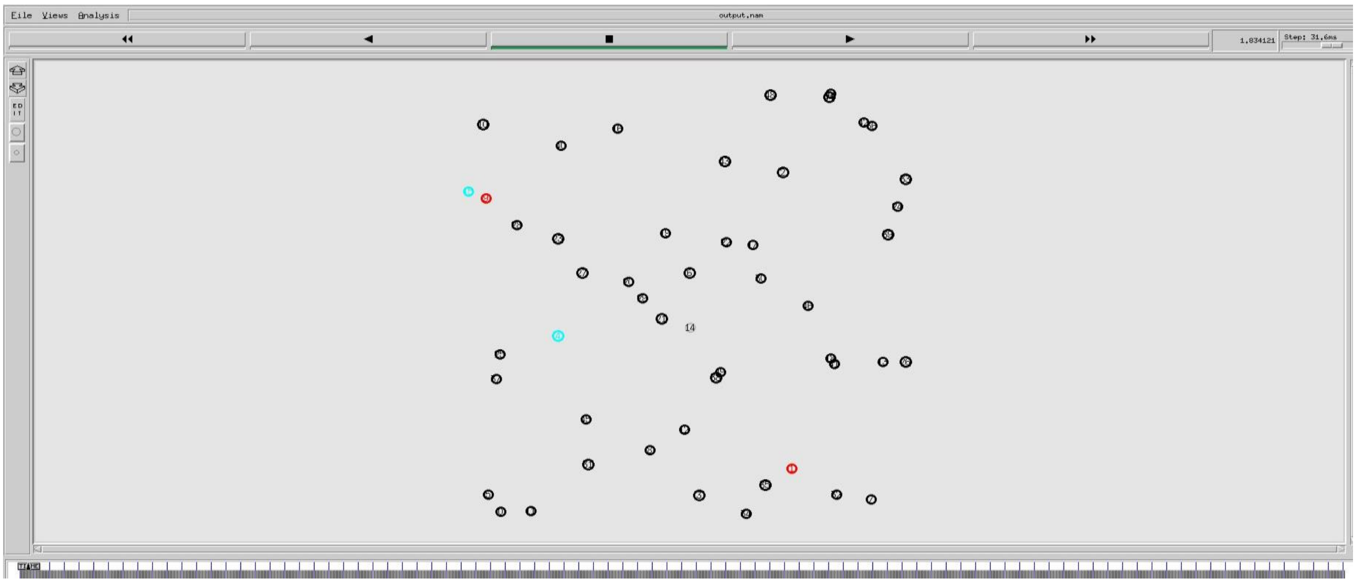


Figure 3. Network Animator

The movement can be seen below when animation is played

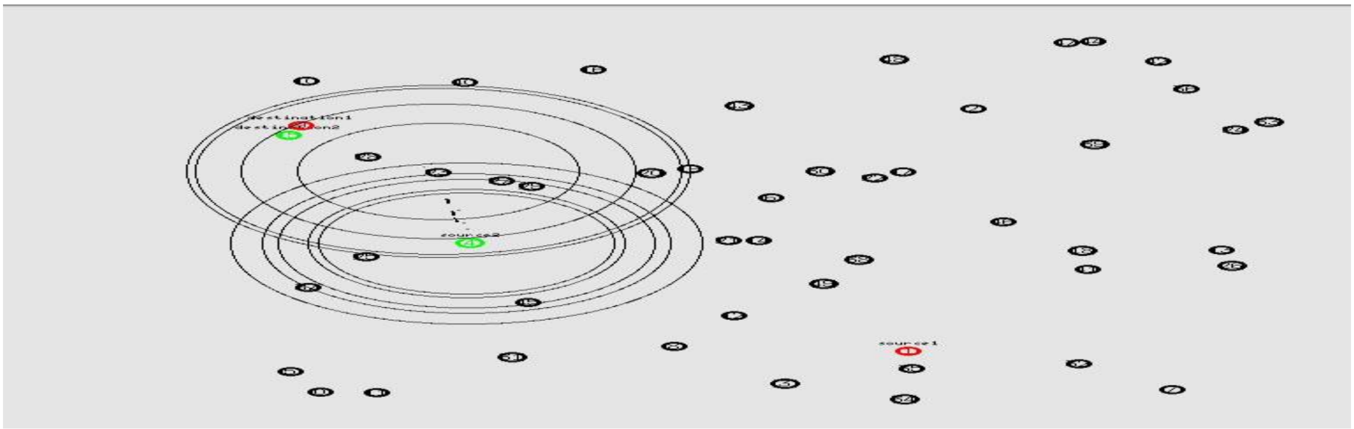


Figure 4. Movements in nodes

Graphical output produced by “xgraph”, while executing the main.tcl file it will produce output.tr and output.nam. In the following graphs the Y- axis varies with the cases and X-axis is constant with the value of misbehaving nodes the red line depicts the PSO_MD and green line depicts the DSR. The output is dependent on 2 possible scenarios that are with the malicious attack and hindrance from malicious attack and last output which will show the number of packets dropped.

6.1 Routing Overhead – PSO vs DSR

The plotting in Figure 5. showcase comparative result of misbehaving nodes (X – Axis) vs routing overhead (Y – axis) between PSO and DSR. The routing overhead shows increase in DSR method on comparison with PSO_MD on nodes 100, 200, 300, 400, 600, 700, 800 and 900.

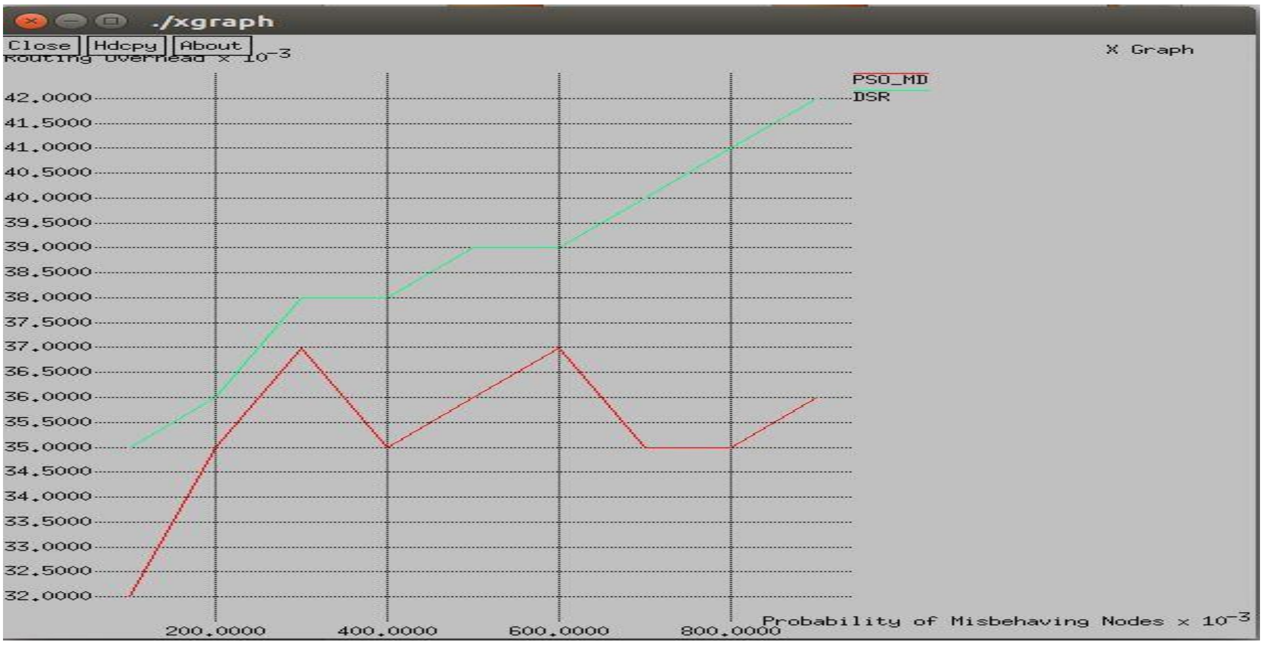


Figure 5. Node vs Routing Overhead

6.2 Packet Success Ratio – PSO vs DSR

In Figure 6. It showcases comparative result of misbehaving nodes (X – Axis) vs packet success ratio (Y – axis) between PSO and DSR. The packet success ratio results in decreasing manner with DSR on comparison with PSO method on nodes 100, 400, 600, 400, 600, 800 and 900.

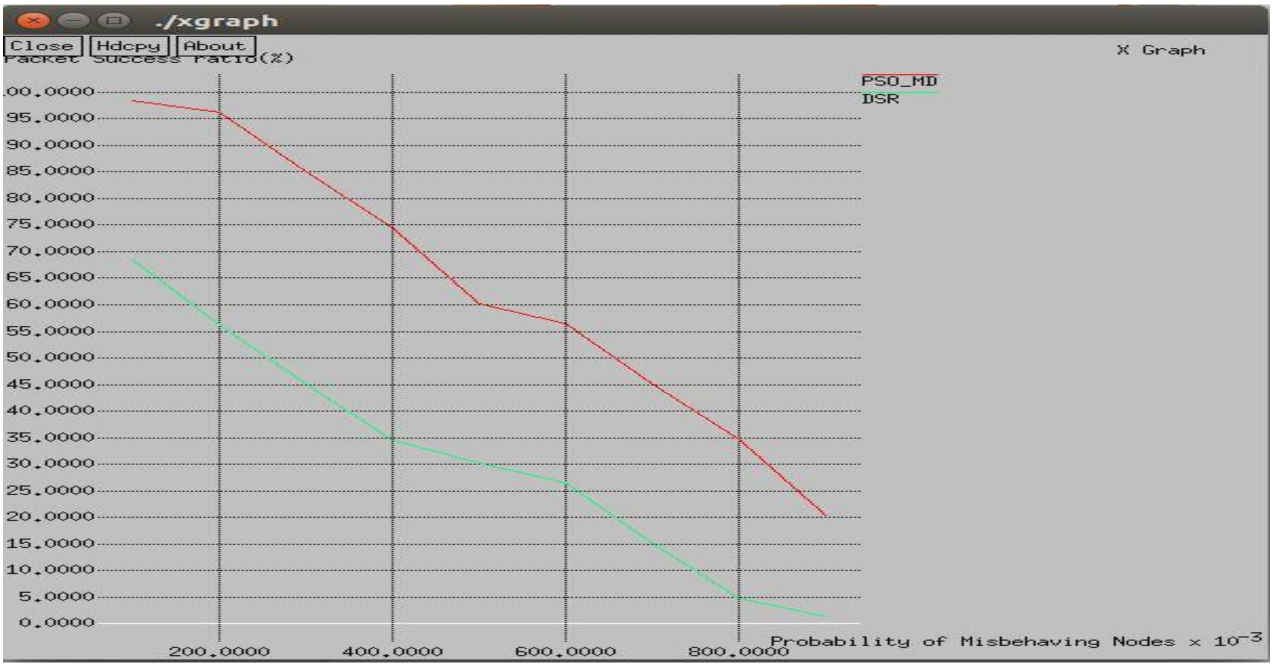


Figure 6. Node vs Packet Success Ratio

6.3 Throughput of Well-behaving Nodes – PSO vs DSR

The plotting in Figure 7. It showcases comparative result the misbehaving nodes (X – Axis) vs throughput of well-behaving nodes (Y – axis) between PSO and DSR. The throughput of well-behaving nodes decreases with DSR as compared to PSO nodes 100, 300, 400,700, 800 and 900.

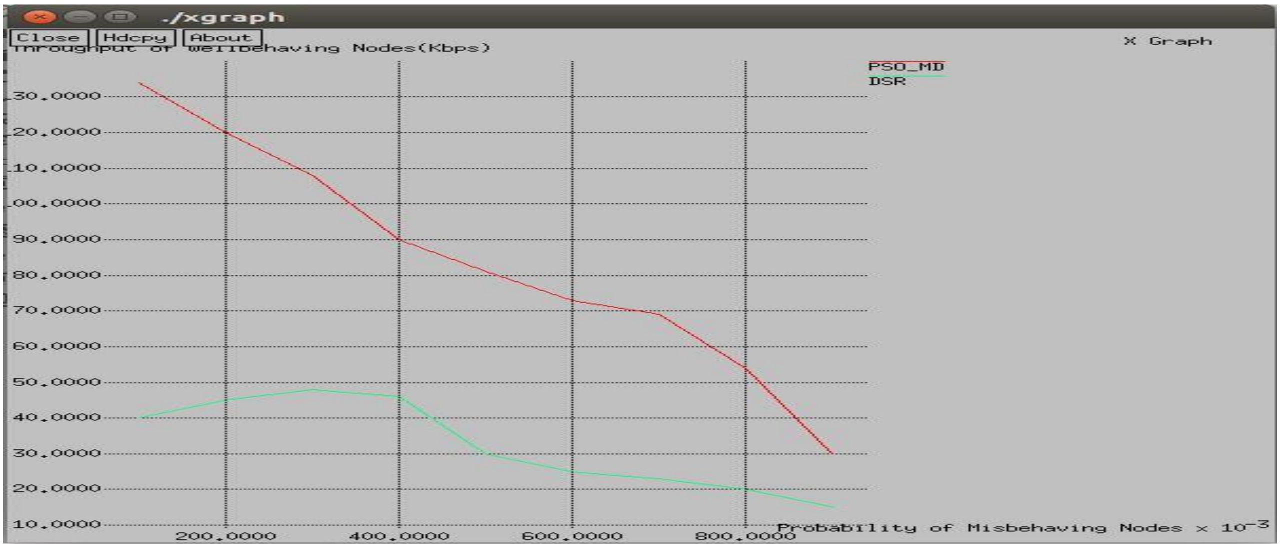


Figure 7. Node vs Throughput of Well Behaving nodes

6.4 Throughput of Misbehaving Nodes – PSO vs DSR

In Figure 8, the graphical representation depicts the comparison of misbehaving nodes (X-axis) and throughput of misbehaving nodes (Y- axis) between PSO method and DSR. The throughput of misbehaving nodes shows increase with PSO_HD on comparison with DSR on nodes 100, 300, 400, 500, 600, 800 and 900.

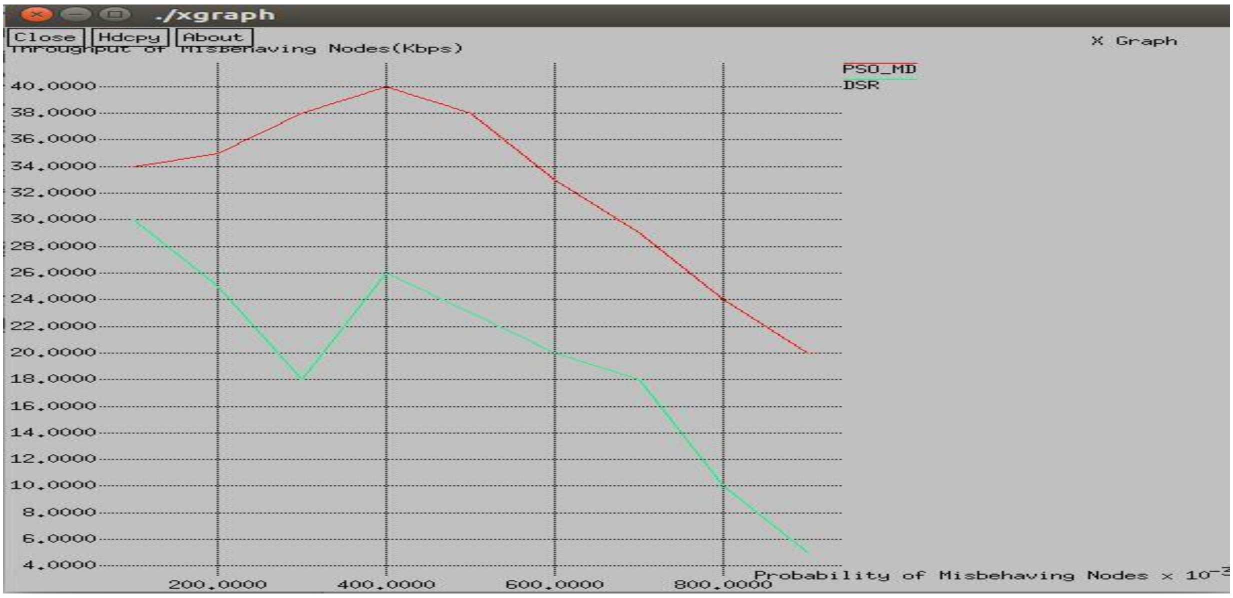
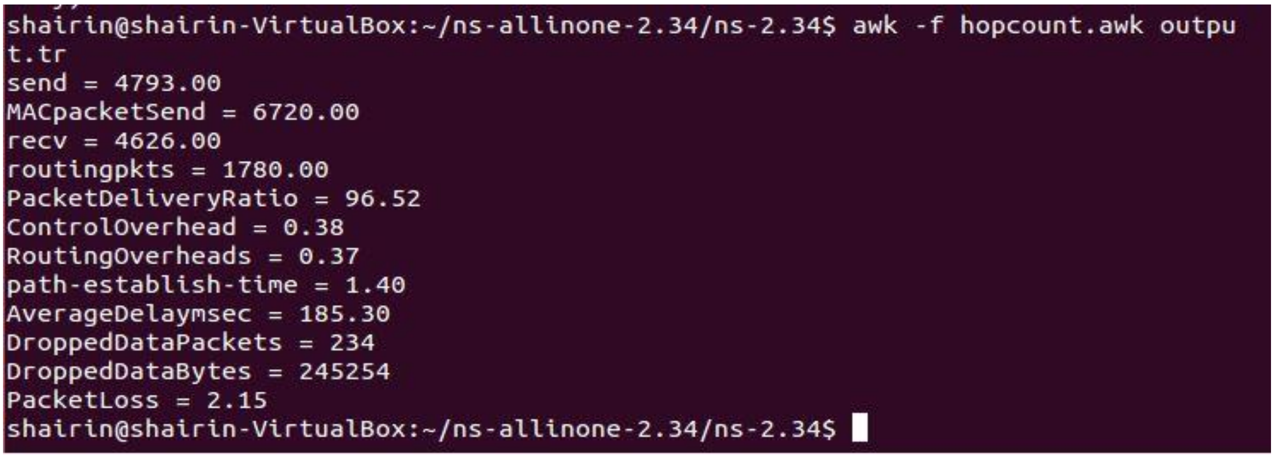


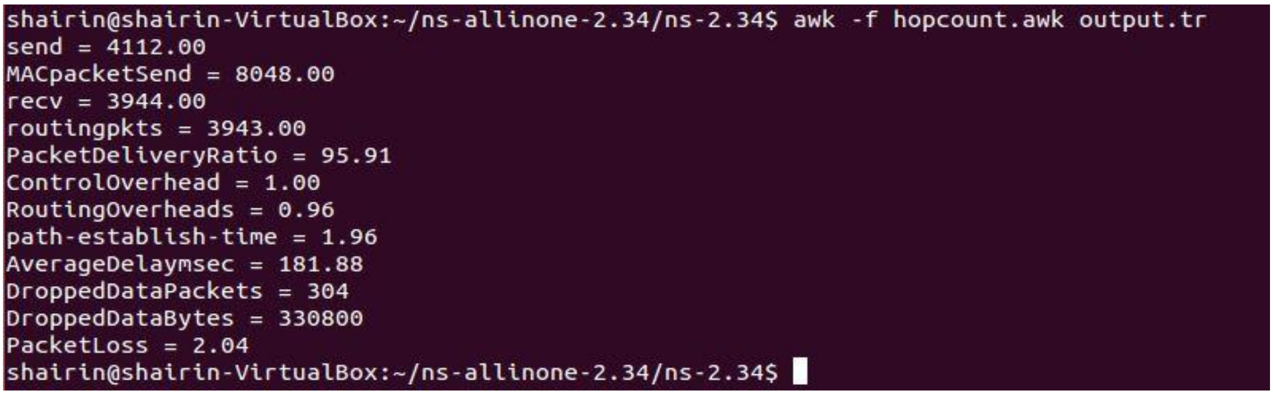
Figure 8. Nodes vs Throughput of Misbehaving nodes

6.5 Hop-count: Most importantly it gives the result of packets sent and received and number of dropped packets and dropped data bytes, depicted in the below Figure 9.and Figure 10. with other outputs as well.



```
shairin@shairin-VirtualBox:~/ns-allinone-2.34/ns-2.34$ awk -f hopcount.awk outpu
t.tr
send = 4793.00
MACpacketSend = 6720.00
recv = 4626.00
routingpkts = 1780.00
PacketDeliveryRatio = 96.52
ControlOverhead = 0.38
RoutingOverheads = 0.37
path-establish-time = 1.40
AverageDelaymsec = 185.30
DroppedDataPackets = 234
DroppedDataBytes = 245254
PacketLoss = 2.15
shairin@shairin-VirtualBox:~/ns-allinone-2.34/ns-2.34$ 
```

Figure 9. Hop-counts when protected from attacks



```
shairin@shairin-VirtualBox:~/ns-allinone-2.34/ns-2.34$ awk -f hopcount.awk output.tr
send = 4112.00
MACpacketSend = 8048.00
recv = 3944.00
routingpkts = 3943.00
PacketDeliveryRatio = 95.91
ControlOverhead = 1.00
RoutingOverheads = 0.96
path-establish-time = 1.96
AverageDelaymsec = 181.88
DroppedDataPackets = 304
DroppedDataBytes = 330800
PacketLoss = 2.04
shairin@shairin-VirtualBox:~/ns-allinone-2.34/ns-2.34$ █
```

Figure 10. Hop-counts when network attacked by attacks

6.6 Discussion

The experiment with or without the attacking showed us significant result of packets dropped ratio in both to see this change is that packets dropped while network is protected from attacks are less as compared to the attacks done, this result therefore shows that the setup in support of research proposal " Efficient detection of malware beaconing" and to accomplish this proposing the use of PSO algorithm , AES encryption standard , beacon signals and matching flow pool was successfully done. The design was normal for the basic testing for manual traffic generation and attack generation just to show the security measured implemented resulted in an appropriately or not but require some major amendments to update this on real network traffics say for example in enterprise level. This design right now does not support heavy load of traffic and attacks in real-time.

As the research papers reviewed and mentioned in related work section were informative to gather techniques for bringing novelty and the outcome gave the positive and indicative results.

7 Conclusion and Future Work

The objective to carry out this research of "efficiently detection of malware beaconing" was not only limited to detection but also how can we reduce the attacks in the network and to this we should apply the encryption standard in the network which therefore lacks in some research papers and using optimization technique give us more real time and accurately detecting the malwares due to the behaviour of malwares that keep on changing with the trends. In future work suggestion can be to use PSO-AES combination makes the network more secure and enhance the detection systems.

The proposal for implementation was successfully done for the research, this project can be done better to support the real-time attacks done on the organisation network. This can be done by using tools like SNORT, R etc. and using dataset available. This can be considered and highly recommended to proceed this in future work.

References

Shalaginov, A., Franke, K. and Huang, X. (2016). Malware Beaconing Detection by Mining Large-scale DNS Logs for Targeted Attack Identification. [online] Semanticscholar.org. Available at: https://www.semanticscholar.org/paper/Malware-Beaconing-Detection-by-

Mining-Large-scale-Shalaginov-Franke/bb67650f0a8d0b8593d7358b38f5afc600bb71c9 [Accessed 9 Aug. 2019].

Patents.justia.com. (2018). US Patent for Malware beaconing detection methods Patent (Patent # 9,979,741 issued May 22, 2018) - Justia Patents Search. [online] Available at: https://patents.justia.com/patent/9979741 [Accessed 9 Aug. 2019].

Duque, S. and Omar, M. (2015). Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS).

Zhang, J., Jang, J., Gu, G., Stoecklin, M. and Hu, X. (2018). Error-Sensor: Mining Information from HTTP Error Traffic for Malware Intelligence.

Hu, X., Jang, J., Stoecklin, M., Wang, T., Schales, D., Kirat, D. and Rao, J. (2016). BAYWATCH: Robust Beaconing Detection to Identify Infected Hosts in Large-Scale Enterprise Networks. 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).

Dengiz, O., Konak, A. and Smith, A. (2011). Connectivity management in mobile ad hoc networks using particle swarm optimization. Ad Hoc Networks, 9(7), pp.1312-1326.

Du, K. and Swamy, M. (2016). Search and optimization by metaheuristics. Ibimapublishing.com. (2019). [online] Available at: https://ibimapublishing.com/articles/JIACS/2018/836339/836339.pdf [Accessed 4 Aug. 2019].

Gao, S., Li, Z., Yao, Y., Xiao, B., Guo, S. and Yang, Y., 2018, May. Software-defined firewall: Enabling malware traffic detection and programmable security control. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security (pp. 413-424). ACM.

AlEroud, A. and Alsmadi, I., 2017. Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach. Journal of Network and Computer Applications, 80, pp.152-164.

Cs.cmu.edu. (2019). Dynamic Source Routing Protocol. [online] Available at: http://www.cs.cmu.edu/~dmaltz/dsr.html [Accessed 9 Aug. 2019].

Hossain, E., Khan, I., Un-Noor, F., Sikander, S. and Sunny, M. (2019). Application of Big Data and Machine Learning in Smart Grid, and Associated Security Concerns: A Review. IEEE Access, 7, pp.13960-13988.

Bonyadi, M. and Michalewicz, Z. (2017). Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. Evolutionary Computation, 25(1), pp.1-54.