Indian Journal of Science and Technology, Vol 8(35), DOI: 10.17485/ijst/2015/v8i35/61671, December 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

# BADUW: Behavioural based Approach for Detecting UDP Worm

**Mohammed Anbar[1]*, Rosni Abdullah[1], Ahmed Manasrah[2], Alhamza Munther[3] and Selvakumar Manickam[1]**

[1]National Advanced IPv6 Centre of Excellence, Universiti Sains Malaysia, Penang, Malaysia; anbar, rosin, selva @nav6.usm.my
[2]Faculty of Information Technology and Computer Sciences, Yarmouk University, Irbid, Jordan; ahmad.a@yu.edu.jo
[3]School of Computer and Communication Engineering, Universiti Malaysia Perlis, Perlis, Malaysia; alhamza80@yahoo.com

## Abstract

A worm is a self-propagating, self-duplicating malicious code that spread without human intervention in computer networks and attacks vulnerable hosts. The severity of network worms depends on the propagation process that degrades the network performance and consume bandwidth and resource (CPU and memory). Thus, this paper presents a behavioral approach for UDP worm (worm uses UDP as transmission mechanism) detection based on scanning and Destination Source Correlation (DSC) behaviors of worm. The proposed approach consists of two sub approaches which are: 1. Statistical Cross-relation Approach for Network Scanning detection (SCANS) approach that is used to detect the presence of network scanning behavior of worm and 2. Worm correlation approach that is used to detect Destination-Source Correlation (DSC) behavior of worm. These behaviors have been chosen among other worm behaviors due to its anomaly behaviors that are clearly exhibit in the network. A salient feature of this approach is that it effective for detecting scanning DSC behaviors of worm with high accuracy. The proposed approach is evaluated with the simulated dataset obtained from Georgia Tech Network Simulator (GTNetS) simulator and confirmed that our approach is efficient in detecting UDP worm than the existing approach.

**Keywords:** Behavioural based Approach, UDP Worm Detection, UDP Worm

## 1. Introduction

Network worms are dangerous threats due to the speed of their propagation. Once a network worm infects a network, it will automatically begin to propagate, which will cause great destruction throughout the network due to network congestion[1,2]. The severity of network worms depends on the propagation process, where in network scanning is initiated to determine the vulnerability of the host and services. Network propagation will degrade network performance and consume bandwidth and resources (CPU and memory) by making the network machines busy due to the voluminous requests that are received and processed, this will create unnecessary traffic, which serves only network worm propagation[3].

The entry point for network worms are the vulnerable hosts and services on the network. To locate vulnerable hosts and services, network worms launch a network scan, which is the first phase of the life cycle of a network worm, this process is followed by the transmission, activation and infection phase[4]. Network scanning enables an attacker to gather information about his or her target, such as the operating systems, system architecture and services that run on each computer. Network scanning is the first step for attackers to gain access to the target network. Identifying the information scanned by attackers can assist system and network administrators to determine the purpose of the attacks. Thus, resources and services can be further protected by patching or installing

*Author for correspondence

security measures, such as firewalls, IDS and computer systems[5–7].

There are two types of network worms which are UDP and TCP worms[8], UDP worms use UDP protocol as transmission mechanism to transfer the malicious code to the victim machine. Meanwhile, TCP worms use TCP protocol as transmission mechanism to transfer the malicious code to the victim. UDP worms are faster than TCP worms in term of propagation and this due to facts that 1. There is no error-checking for packets. 2. UDP header size is 8 bytes while TCP header size is 20 bytes. 3. No acknowledgment must return from destination to let source starts sending packets.

The rest of the paper is organized as follows. Section 2 presents a review of network worm approaches found in the literature. In Section 3, we describe our proposed approach for detecting UDP worms. An evaluation of our approach is presented in Section 4. Section 5 concludes our work, while Section 6 presents possible future work.

## 2. Related Work

There are many approaches that have been proposed for UDP worm detection. In the following, the commonly used behavioral based approaches which are used to detect UDP worms based on an Artificial Neural Network and connection failure approaches are reviewed.

### 2.1 Artificial Neural Network (ANN) based Network Worm Detection

Stopel et al.[9] proposed an approach for detecting infected host by network worms based on ANN. This approach uses the infected host resources such as CPU and memory in the network. In addition, the study utilized feature selection techniques for the dimension reduction; the used selection techniques are as follows: 1. The relation between the inputs and the hidden neuron's relative variance, 2. The Fisher score ranking, 3. Gain Ratio Filter. The average accuracy for the proposed approach was 99.98%. The outputs of the selection techniques are the features that have impact in computer behavior which are infected by network worms. The study evaluated each technique by preprocessing the dataset accordingly and training the ANN model with the preprocessed data. Furthermore, the ability of the model to detect the presence of a new computer network worm was evaluated, in particular, during heavy user activity on the infected computer.

As a result, Stopel et al. enhances the proposed approach in[10] by adopting ANN and two other known classifications techniques, Decision Tree and k-Nearest Neighbours, to observe their ability to classify computer network worms during heavy user activity on the infected computers. In this study, a number of computers infected with a different number of network worms and different parameters distributed in the various measurements such as processor features, TCP layer features, UDP layer features, IP layer features and low Network Interface features. Moreover, the study evaluated each technique by pre-processing the dataset by training the ANN model with the pre-processed data. The average accuracy for the proposed approach was 85.0%.

All in all, the proposed approach in[9,10] detects malicious activity of network worms by looking at the attributes derived from the computer operation parameters such as memory usage, CPU usage and traffic activity. The main drawback of this model was appearing in misclassifications of network worms in the beginning of their activity. Meanwhile, observing all computer features in the network are time and recourse consuming. Moreover, these approaches considered an agent based approach where an agent software has to be deployed into each machine within the monitored network. This makes it tedious to manage a huge number of machines at once.

Therefore, Farag et al.[11] proposes a method for detecting unknown network worms based on local victim information. The proposed method initialized an ANN for classifying network worm / non-network worm traffic in every host. The traffic classification was performed by using two models which are Classification Prediction Combined model (CPC) and Classification Prediction Separated (CPS) model. In CPC the goal was to use ANN to produce two outputs (network worm traffic and percentage of infection). In CPS model, two ANN networks were used to solve the classification problem. To evaluate the proposed approach, a simulated dataset was adopted and the output generated a reliable result with accuracy of 99.96% in detecting the presence of network worm over the network, even for unknown network worms.

The ANN approach has computational advantages when real-time computation is needed and has the potential to detect previously unknown network worms with high level of accuracy. Also, ANN has advantage to reduce the feature dimensionality. However, the two shortcomings for ANN techniques are: 1. Training period (takes

time) and 2. Involvement problem (any changes in target environment will affect the training dataset).

The up mentioned shortages for ANN techniques open the avenues for the researchers to find out other behavioural approaches to overcome the shortages of ANN techniques, such as the Connection Failure Based Network Worm Detection.

## 2.2 Connection Failure based Network Worm Detection

The connection failure in the network appears in the form of ICMP Type 3 (port unreachable), ICMP Type-3 (destination unreachable) and TCP RST packets. The existing of these packets in a high rate means that there are many connection failures which are considered as very strong footprint and symptom for network scanning (first stage in network worm life cycle)[12] . Table 1 shows the packets which generated from connection failure. A global detection algorithm based on Internet Control Message Protocol (ICMP) destination unreachable (ICMP –T3) error messages has been proposed by Berk et al.[13] . The purpose of the ICMP is to provide feedback on problems in IP communication and routing. The proposed algorithm utilized two type of ICMP-T3 which are ICMP-T3 host unreachable and ICMP-T3 network unreachable. ICMP-T3 host unreachable is generated when host send out a TCP-SYN or UDP packet to an inactive destination, the destination will reply with ICMP-T3 host unreachable. Meanwhile, transit level routers generate ICMP-T3 network unreachable.

**Table 1.** Packets which generated from connection failure

| Packet | Reason to generate |
|---|---|
| ICMP Type3 code1 (host unreachable) | Generated when TCP/SYN or UDP packet is sent out to an unused IP address. |
| ICMP Type3 code 3 (port unreachable) | Generated when TCP/SYN or UDP packet is sent out to an existing address but the port closed |
| TCP RST packet | Generated in two cases, when a TCP-SYN packet is sent out to an existing host but the port is closed, and when a TCP-SYN carries forged source IP address that is send to an existing host, the destination host will reply with SYN/ACK packet to the real host, in this case, the TCP RST packet is send from a real IP address to the destination. |

A router can be considered as a transit level when it is not responsible for directly delivering packets to a local network but rather passing on packets to other routers for further routing. By forwarding these messages to a central collection point, an alert can be generated when the number of such error messages reaches a certain threshold.

The drawback of this approach is that the ICMP-T3 (port unreachable) that can help detecting UDP network worm is not taken into consideration that leads to high false positive in term of network worm using UDP transmission schema detection.

Similarly, Jung et al.[14] proposed an algorithm based on how many connection attempts are refused or unanswered. The assumption assumes that, in the normal behaviour the ratio of connection failure is not notable. This is because that most of the targets accessibility are delegated to DNS server. In other words, the user is likely to use web client applications to reach their targets with minimal connection failure. On the other hands, the infected host selects its target randomly without any prior knowledge about the active host or service, so it is likely to get much connection failure. This algorithm identifies the infection host if it makes four or five connection failure, and it does not require training of the system in advance.

The drawback of the proposed approach is that it focuses on detecting TCP traffic only, which makes the presence of UDP network worms ignored.

Schechter et al.[15] introduced another network worm detection method based on the number of failure connections. In order to reduce the number of false positive rates, only the first failed connection sent from the forged source IP address to different destination IP address is recorded and normal network activities are considered as well.

The activities of network worms and normal users can be differentiated from the fact that a network worm usually scan different IP address and shows a larger number of connection failure packets. Usually, normal users produce first failed connection packets. The approach reduces the number of false positive rates, but does not work well on detecting "stealthy" network worm[15].

In general, the approaches that based on connection failures have high false positive for two main reasons:

First, depending only on the number of failure connections for detecting network worms is not accurate

enough since there are many network worm behaviours besides the connection failure.

Second, there are many malicious codes share the network worms with connection failure behaviours, which leads to misclassification and low accuracy in terms of detection.

Consequently, the Destination-Source Correlation detector (DSC) proposed in[16] is based on correlating an incoming connection on a given port with outgoing infections on that port. If the outgoing connection rate (scanning) exceeds certain threshold established during training, the alarm is triggered.

For scanning traffic originating from the network, the appearing of source host is checked in the corresponding filter, if the same host repeatedly sends out more packets and exceeds a trained threshold, an alert will be triggered, for outgoing scanning, a simple anomaly detection heuristics are used to identify this unusual pattern. In practice, the normal profile of the outbound scan rate of services which exhibit infection-like behaviour is created in idle networks or networks that exhibit no fast infection-like behaviour during a training period[16].

The impediments of DSC can be summarized as the follows:

- The correlation between an incoming connection on a given port and outgoing infection to that port may exhibit false positive, since there are existing application exhibit the same mentioned correlation (i.e Gnutella p2p application).
- Using Bloom filter for each destination port is resource consuming (memory and CPU) especially for networks, which have different applications because, each filter must process each packet to find out the corresponding port.
- Using simple heuristics for scanning detection are not efficient in terms of detection accuracy because, simple heuristics will not consider all symptoms of network scanning (such as packets that are generated when inactive hosts or services are being scanning).
- Assigning threshold for each destination port is not practical and time consuming since the individual threshold is obtained from the training process. Moreover, the training process must be repeated if any changes happened in the network services (add/remove service in the network).

Despite the disadvantages mentioned above, a survey investigate different behaviour based approaches for network worm detection conclude that DSC detector consider one of the best behaviour based approaches for network worm detection [3], this is why DSC has been chosen to compared with.

## 3. The Proposed approach

Our approach, Behavioural based Approach for Detecting UDP Worm (BADUW) aims to detect the presence of UDP worms in the network. The proposed approach based on assumption that the first step for network worm to gain access to the network is performing a network scanning to identify the vulnerable hosts and services.

Once a susceptible host or service exists, the malicious code from the sender starts to propagate to its destination. The packets that were used to transfer malicious code from the sender to the destination have specific patterns and noticeable behaviours. After the malicious code infects the destination host, this new host will act in the same manner as the host that infected it (DSC behaviour).

BADUW consist of two sub-approaches 1. A statistical cross-relation approach for network scanning detection (SCANS) which aims to detect the presence of UDP and TCP random and sequential scanning, 2. Worm correlation approach that aims to detect the DSC behaviour of the worm. Figure 1 depicts the proposed architecture of BADUW.

### 3.1 A Statistical Cross-Relation Approach for Network Scanning Detection (SCANS)

SCANS[5] aims to detect both TCP and UDP random and sequential scanning. Most network worms use random

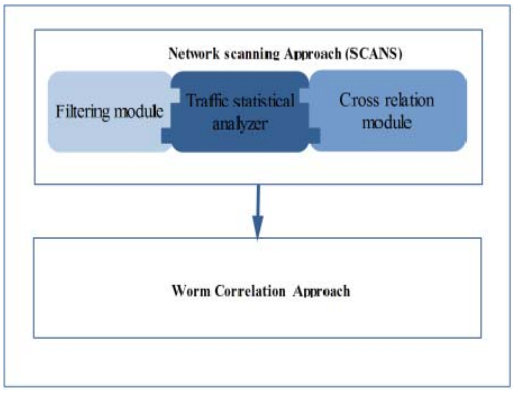


**Figure 1.** The architecture of the BADUW.

scanning to identify vulnerable hosts4, such as the worms Code Red II[17] and Slammer[18]. SCANS is based on a hypothesis that traffic with high rates of ICMP type 3 (destination and port unreachable) and TCP RST packets produces a very strong footprint and provides evidence of network scanning. In this paper, we focus on UDP type of random and sequential scanning because our target is to detect the presence of UDP worm in the network.

UDP scanning used by attackers to determine which services are active on which hosts. UDP scanning can be detected using SCANS as follows:

1. For UDP random scanning, the source IP will be identified as a scanner if:

$$\{\text{ICMP\_Port\_Log}\} \cap \{\text{ICMP\_Host\_Log}\} \neq \phi$$

*where*,

$$\left(\text{Source IP}_i \text{ count}_{\text{ICMP\_Port\_Log}} > \alpha\right) \textit{AND}$$

$$\left(\text{Source IP}_i \text{ count}_{\text{ICMP\_Host\_Log}} > \alpha\right)$$

$\alpha$ is the threshold.

Where source $IP_i$ is the IP address and the count refers to the number of packets (ICMP Type 3 code 3 or ICMP Type 3 code 1) for each source IP in the Log tables.

An alert is triggered if the source IP $IP_i$ receives a number of ICMP Type 3 code 3 packets and that number exceeds the threshold for ICMP Type 3 code 3 packets and the source IP $IP_i$ receives a number of ICMP Type 3 code 1 and that number exceeds the threshold for ICMP Type 3 code 1 packets.

2. Whereas for UDP sequential scanning, the source IP will be identified as a scanner if:

$$\left(\text{Source IP}_i \text{ count}_{\text{ICMP\_Port\_Log}} > \alpha\right)$$

*OR*

$$\left(UDP_{packet\_count}(x) > \alpha\right)$$

Where $x$ is the same source *IP* that is sending the same UDP packet size to different destination IP on the same destination port and $\alpha$ is the threshold.

An alert is triggered if the source IP $IP_i$ receives a number of ICMP Type 3 code 3 packets and that number exceeds the threshold for ICMP Type 3 code 3 packets.

The purpose of ICMP Log Host table is to record all source IPs that is sending out different packets to different destinations targeting inactive host. Meanwhile, the purpose of ICMP type 3, code 3 packets (ICMP Log Port) is to record all source IP addresses that are sending out different packets to different destinations targeting inactive service.

## 3.2 Worm Correlation Approach

Worm correlation approach aims to check whether the scanner IP performs DSC behaviour or not. We claim that the IP which performs scanning and DSC behaviours will be identified as infected IP. Network worm behaviours are usually repetitious and predictable, thereby making it possible to detect them.

Gu et al.[16] defined one of the predictable behaviours for network worms by correlating an incoming connection on a given port with the subsequent on-going infection on that port. This behaviour is called the Destination-Source Correlation (DSC). DSC behaviour is anomalous behaviour that clearly appears during network worm propagation.

The following scenario is an example of DSC behaviour. A host receives a packet on port i, and then starts sending packets to different destinations destined for port i. If the number of sent packets that are targeted for port i exceeds the predefined threshold, the host suspects DSC behaviour, which is also illustrated in Figure 2.

In Figure 2, host F sent out packets targeting port 25 to other hosts A, B, C, D and E. Then, host B sent out the same packet to a number of hosts with the same port number 25 and the number of destination hosts exceeded the threshold. Therefore, host B is a vulnerable host that exhibits DSC behaviour. On the other hand, the other hosts C, E, D and A did not send any packets that targeted port 25, which implies that these hosts did not exhibit DSC behaviour. The vulnerable hosts can be protected by installing new antivirus software, by updating the current antivirus software, or by using other security solutions.

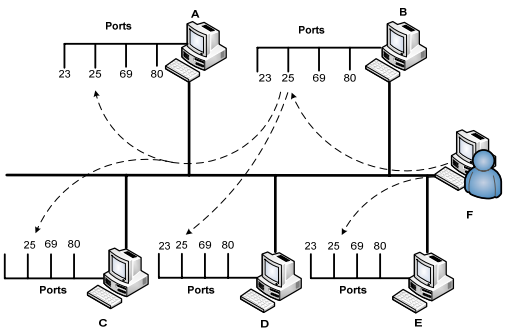


**Figure 2.** DSC behaviours of network worms.

The starting point for the correlation approach begins after the SCANS approach detects the IPs performing the scans. The primarily objective of the correlation approach is to check whether the scanner IPs perform DSC behaviour or not. The correlation approach flow chart is shown in Figure 3.

The incoming traffic must be checked to determine whether it has scanner IPs or not, if the scanner IP is determined then scanner IP must be checked to determine whether it performs DSC behaviour or not, if scanner IP perform DSC behaviour then the counter will increment, an alert will be triggered if the counter exceeded a certain threshold, in case of the scanner IP does not perform DSC behaviour, then the scanner IP will be saved in the log table for further analysis.

The worm correlation approach is proposed for the following main purposes:

1) To detect the IPs that perform both scanning and DSC behaviour (the detected IPs will be considered infected IPs); and
2) To log all the IPs that performs only scanning, but not DSC, behaviour.

Each network worm performs scanning but not every scanner is a network worm. The DSC behaviour, in addition to the scanning behaviour, is considered to detect the presence of network worms in the network. This method, instead of considering the scanning behaviour alone, can increase the accuracy of network worm detection.

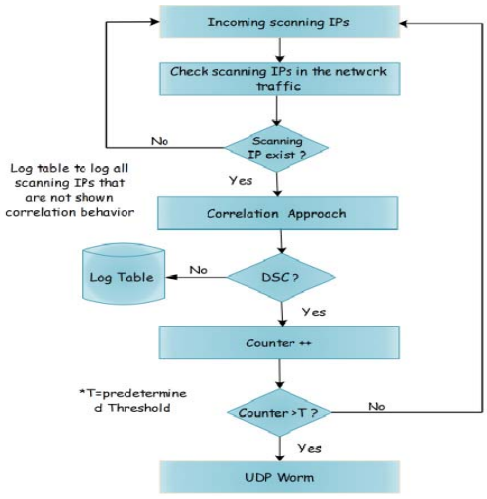


**Figure 3.** The worm correlation approach flow charts.

The scanning log table, which is shown in Figure 3 is created to log all IPs that performs network scanning, but do not shows DSC behaviour for further analysis by a network administrator. In other words, the IPs that exhibit both scanning and DSC behaviour are identified as infected IP. Meanwhile, the IPs that exhibit scanning behaviour alone is logged in the scanning log table. This IPs will be checked by a network administrator to see whether they are triggered from legitimate or non-legitimate sources.

# 4. Evaluations

## 4.1 Simulation Environment (GTNetS Simulator)

The Georgia Tech Network Simulator (GTNetS) is a full-featured network simulation environment that allows researchers in computer networks to study the behavior of moderate to large scale networks, under a variety of conditions. The design philosophy of GTNetS is to create a simulation environment that is structured much like actual networks are structured. For example, in GTNetS, there is clear and distinct separation of protocol stack layers[19].

GTNetS[19] simulator is used to design a network topology that simulates a UDP network worm (i.e., Slammer network worm), GTNetS starts with an infected host that randomly accesses other vulnerable hosts and tries to infect them, all traffic that bypass the network are logged into the log file to be used latter as input for the proposed approach.The designed topologies are created based on the following input parameters:

- **Simulation Duration:** The total time for simulating a target network worm; within this time the network worm must complete its life cycle (target finding, propagation, activation and infection).
- **No. of Nodes:** The number of nodes in the created topology.
- **Scan rate:** The number of packets that are sent out from infected hosts to different distension hosts within specific time window (i.e., seconds).

The reasons of selecting GTNetS simulators that is successfully simulated worm traffic and it is used in different worm detection literatures[7,11].

## 4.2 Dataset

This section explains the reasons for the use a simulated dataset that contains net-work worm traffic. First, there

is a general tendency to register the scarce availability of network worm traces. Moreover, even the traffic traces used in previous analysis/papers such as Slammer[13] and Code-red[12] are typically not made public. Another reason is that the characteristics of the market traces, when too small or when sampled, will be inappropriate for performing traffic analysis/characterization[18].

For example, the National Laboratory for Applied Network analysis (NLANR) collects daily eight traces of ninety seconds every from many backbone links within the USA. Among them, there are traces that capture the times when network worms unfold (Code-red I and II, Slammer). Furthermore, most of the obtainable traces, for example, those from CAIDA[20] and MIT[21] do not contain the remainder of the legitimate traffic flowing on the links.

This limitation is often a result of the observation method, which could be a network telescope or as a result of traces that were deliberately filtered before they were made obtainable. Thus, the process does not enable a satisfactory comparison of the network worm traffic. Therefore, there is legitimate traffic flowing on an identical link at an identical time. Likewise, the approach does not enable studying the impact of network worm traffic on the general combination traffic.

Secondly, to obtain reliable results, traces need sanitization before the analysis. In-deed, when traces are reported as containing only network worm traffic, they are usually filtered by port numbers or other simple indicators. Thus, non-network worm traffic may be present inside the trace.

The captured packet traces can occasionally contain spurious data because of hardware and software errors during data acquisition (as replication of data).

Finally, the DARPA 1999 benchmark data sets were used as input traffic to evaluate the efficiency and accuracy of the IDS, which did not include network worm traffic because of the aforementioned reasons[22,23]. Therefore, network worm traffic was generated using the GTNetS simulator to be used as the input traffic for the proposed approach.

## 4.3 Evaluation Metrics

To evaluate the performance of the BADUW approach, the standard formula (Equation 1) was used to calculate the accuracy of detection, as explained by[11]. The accuracy is the ratio between the summation of the true-positives and the true-negatives that was divided by the summation of the true-positive, true-negative, false-positive, and false-negative values.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Where, True Positive (TP): The capability of the BADUW to correctly detect the infected host. The True-Positive rate can be calculated as shown in Equation 2.

$$TPR = \frac{TP}{(TP + FN)} \quad (2)$$

False Positive (FP): The incorrectly detected hosts as infected hosts with a condition. The False-Positive rate can be calculated as shown in Equation 3.

$$FPR = \frac{FP}{(FP + TN)} \quad (3)$$

False Negative (FN): The number of infected hosts that BADUW could not detect with a condition.

Equation 4 shows the formula that will be used to calculate the network worm detection accuracy of the BADUW over the test bed after the TN is removed.

$$Accuracy = \frac{TP}{TP + FP + FN} \quad (4)$$

## 4.4 UDP Network Worm Scenario

This scenario aims to test the accuracy of BADUW in detecting the presence of UDP network worms in the network. In addition, this scenario aims to compare the accuracy of the BADUW and the DSC in detecting the presence of UDP network worms in the network.The network initially contains one infected host (Table 2 shows details of the infected UDP worm), which primarily explores the network to find vulnerable hosts and services by launching network scanning. Once a vulnerable victim exists, the malicious code from the infected host will be transferred to the target host, and the target host will behave in the same manner as the first infected host. A log file is used to record the traffic that bypassed the network

**Table 2.** Details of the network worms

| Network worm name | Slammer |
|---|---|
| **Target port** | 1413 |
| **Target OS** | Windows |
| **Transmission type** | UDP |
| **Target service MS SQL server** | Target service MS SQL server |

to be used as input for the BADUW. Figure 4 indicates network topology for UDP network worm test consist of a tree topology, four routers, and 12 nodes.

The designed test bed is randomly created by the GTNetS simulator based on the input parameters, as described in Table 3. The dataset for UDP network worm consists of 29703 row packets (these packets are extracted from the simulator log file). Table 4 summarizes the dataset packet distribution.

Table 4 indicates that the existence of ICMP Type 3, code 3 (port unreachable) packets demonstrates a UDP sequential scanning based on SCANS assumption. In addition, the presence of ICMP Type 3, code 1 (host unreachable) packets and ICMP Type 3, code 3 (port unreachable) packets are considered as clear symptoms of UDP random scanning based on SCANS. The existence of random and sequential UDP scanning is caused by the simulated UDP network worm in seeking vulnerable hosts and services. Table 5 shows the infected IPs in the dataset for each second.

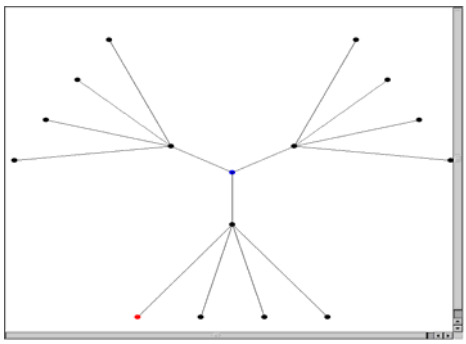

**Figure 4.** Network topology for UDP network worm test (topology created by GTNetS simulator).

**Table 3.** Parameters used in topology (Figure 4)

| Parameter | Value |
|---|---|
| Simulation time | 3 second |
| No of nodes | 12 |
| Scan rate | 50 hosts/seconds |

**Table 4.** The packet distribution for test1 dataset

| UDP | ICMP type3 | |
|---|---|---|
| Number of packets | ICMP code | Count |
| 1268 | 3 | 26637 |
| | 1 | 3066 |

Table 5 indicates 12 infected hosts, which is the actual number in the designed topology. All of these hosts are vulnerable and were therefore infected by the launched network worm. The accuracy detection of the proposed approach was tested using the BADUW on the simulated dataset. Table 6 depicts the high accuracy for UDP network worm detection using BADUW approach.

The high accuracy and therefore, the neat performance measurement ratios are related to the role of SCANS in detecting the UDP scanning and correlation approach that were applied in the traffic for each second. Figures 5 and 6 indicate the IPs conducting random and sequential scanning, respectively, at Second 1 (selected as an example among the timeline). The SCANS approach detected the IPs conducting random and sequential scanning. The detected IPs are used as input for the correlation approach. Figure 7 indicates the IPs that demonstrated scanning and DSC behaviour at Second 1.

Let the threshold for ICMP Port Log = 1 and for ICMP Host Log = 1 at 1 second time window. As depicted in Figure 5, the source IPs (192.168.0.0, 192.168.0.1, 192.168.0.10, 192.168.0.2 and 192.168.0.4) that were detected at second 1 have ICMP Type 3, code 1 and ICMP

**Table 5.** The infected IPs in the dataset for each second

| Second | Infected IPs |
|---|---|
| Second 1 | 192.168.0.0, 192.168.0.4, 192.168.0.1, 192.168.0.2, 192.168.0.10 |
| Second 2 | 192.168.0.0, 192.168.0.4, 192.168.0.1, 192.168.0.2, 192.168.0.10, 192.168.0.3, 192.168.0.7, 192.168.0.6, 192.168.0.11, 192.168.0.5, 192.168.0.9, 192.168.0.8, |
| Second 3 | 192.168.0.0, 192.168.0.4, 192.168.0.1, 192.168.0.2, 192.168.0.10, 192.168.0.3, 192.168.0.7, 192.168.0.6, 192.168.0.11, 192.168.0.5, 192.168.0.9, 192.168.0.8, |

**Table 6.** The evaluation of BADUW approach

| Seconds | 1 | 2 | 3 |
|---|---|---|---|
| No. of infected machine | 2 | 12 | 12 |
| Detected machine | 2 | 12 | 12 |
| False positive | 0 | 0 | 0 |
| False negative | 0 | 0 | 0 |
| True positive | 2 | 12 | 12 |
| Accuracy | 100% | 100% | 100% |

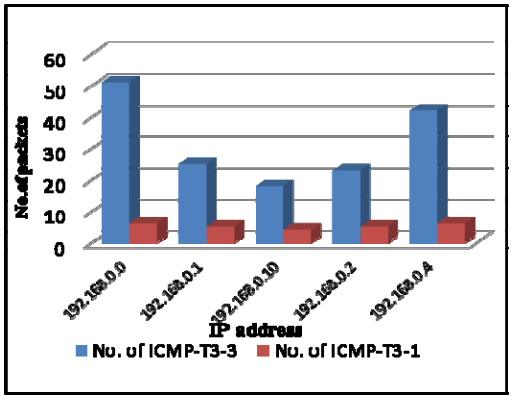


**Figure 5.** The IPs that performing UDP random scanning in the

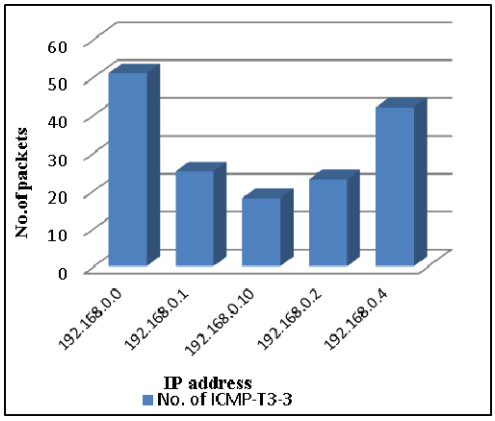


**Figure 6.** The IPs that performing sequential random scanning in the first second.

Type 3, code 3 packets. This means that the detected IPs involved in UDP random scanning in line with the BADUW approach (see Section 3.1). Table 5 indicates that the detected IPs are infected.

Let the threshold for ICMP Port Log = 7 at 1 s time window. As depicted in Figure 6, the source IPs (192.168.0.0, 192.168.0.1, 192.168.0.10, 192.168.0.2 and 192.168.0.4) that were detected within the first second sent a variety of ICMP Type 3, code 3 packets to different destination hosts, which exceeded the threshold. This result indicates that the mentioned IPs is involved in UDP sequential scanning according to the BADUW approach (see Section 3.1). Table 5 indicates that the detected IPs are infected. The SCANS approach detected the IPs that demonstrated scanning behavior (192.168.0.0, 192.168.0.1, 192.168.0.10, 192.168.0.2 and 192.168.0.4), and these are used as inputs for the correlation approach.

Let the threshold for the worm correlation approach be equal to 2, which implies that any host receiving a packet on port X and resending a packet to two additional destination hosts (or more ) on the same port is identified as an IP that demonstrates a DSC behaviour. Figure 7 depicts the IPs that exhibit scanning and DSC behaviours. Thus, the IPs 192.168.0.0, 192.168.0.1, 192.168.0.10, 192.168.0.2 and 192.168.0.4 are declared as infected IPs. Table 5 indicates that the detected IPs are infected.

The same methodology used in the aforementioned approach was applied for the rest of the traffic for each second to obtain the detection.

Based on SCANS[5], the threshold for ICMP Port Log and for ICMP Host Log is equal 1 means that there is at least one ICMP (port unreachable) and ICMP (host unreachable) originated from same source IP exist in both ICMP Port Log and for ICMP Host Log logs table. Meanwhile, the threshold for ICMP Port Log = 7 means that there is an IP sent seven or more UDP packets to different destination targeting inactive service. The use of threshold values in SCANS and worm correlation approaches comes from the observation and analysis of UDP network worm traffic.

## 4.5 Comparison between the BADUW and DSC

This section compares the BADUW and the DSC results to evaluate the detection accuracy of the BADUW. The DSC approach detected the IPs that exhibited DSC behavior, and then checked the outgoing scanning for every IP. If the IP address exceeded the threshold for the scanning rate, then that IP is identified as infected. Table 7 indicates the IPs that exhibited DSC behavior and their scan rate

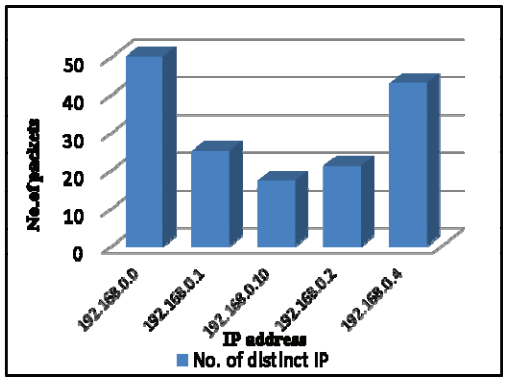


**Figure 7.** The IPs that is exhibiting scanning and DSC behavior in the first second.

**Table 7.** The IPs that is exhibiting DSC and it scan rate for each second

| IPs | No. of packets in second 1 | No. of packets in second 2 | No. of packets in second 3 |
|---|---|---|---|
| 192.168.0.0 | 50 | 50 | 50 |
| 192.168.0.1 | 25 | 50 | 50 |
| 192.168.0.10 | 17 | 50 | 50 |
| 192.168.0.11 | 0 | 43 | 50 |
| 192.168.0.2 | 21 | 50 | 50 |
| 192.168.0.3 | 0 | 46 | 50 |
| 192.168.0.4 | 43 | 50 | 50 |
| 192.168.0.5 | 0 | 36 | 50 |
| 192.168.0.6 | 0 | 45 | 50 |
| 192.168.0.7 | 0 | 46 | 50 |
| 192.168.0.8 | 0 | 22 | 50 |
| 192.168.0.9 | 0 | 24 | 50 |

for each second. In the BADUW, the SCANS approach detected the IPs that demonstrate scanning behavior, then the scanning IPs that demonstrated the DSC behavior are identified as infected.

Table 8 indicates the comparison between the BADUW and the DSC in terms of false positive, false negative, true positive and accuracy for each second.

As shown in Table 8, DSC hits 100% accuracy in second 3, but in second 1 there are five nodes out of 12 nodes (which detected in second 3) are already infected and start to propagate in the network but DSC was unable to detect them in second 1 because their scant rate is below than 50 host/second (refer to Table 3), DSC needs extra two seconds to detect the infected IP in second one.

The reason for extra time required by DSC to detect the infected nodes is the using of the simple heuristics technique which is based on counting the number of outgoing connections originated from infected node, the number of outgoing connections (scanning) originated from infected node will be increased as long as the infected node stays active in the network. In contrast, BAWDSA depends on SCANS to detect the scanning originated from infected node. Figure 8 shows the accuracy percentage for BADUW and the DSC.

As shown in Figure 8 the detection accuracy for BADUW exceeds that of DSC. The low DSC accuracy is caused by the simple heuristics technique that was used for scanning detection. On the other hand, the BADUW

**Table 8.** BADUW vs. DSC in term of false positive, false negative, true positive and accuracy for each second

| Seconds | 1 | | 2 | | 3 | |
|---|---|---|---|---|---|---|
| No. of infected machine | 5 | | 12 | | 12 | |
| Approaches | BADUW | DSC | BADUW | DSC | BADUW | DSC |
| Detected machine | 5 | 1 | 12 | 5 | 12 | 12 |
| False positive | 0 | 0 | 0 | 0 | 0 | 0 |
| False negative | 0 | 4 | 0 | 7 | 0 | 0 |
| True positive | 5 | 1 | 12 | 5 | 12 | 12 |
| Accuracy | 100 % | 20 % | 100 % | 41% | 100 % | 100 % |

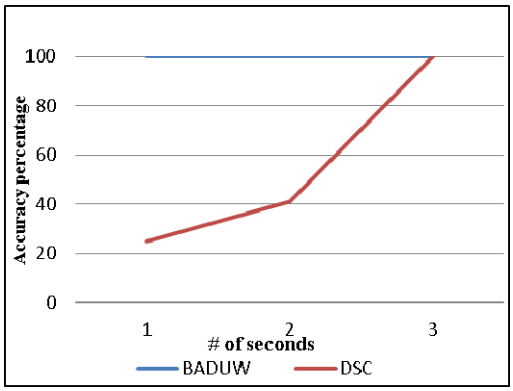


**Figure 8.** Accuracy percentage for BADUW and the DSC.

detected the infected IPs with 100% accuracy, which is often related to the use of the SCANS and worm correlation approaches. SCANS approach is highly contributed in detecting the presence of network scanning which usually used by network worm to find out the vulnerable host and service. Meanwhile, the worm correlation approach verified that the scanner IP is triggered from machine infected by network worm.

## 5. Conclusion

In this paper, we proposed a new approach for detecting UDP worm. The proposed approach can detect UDP worm based on scanning and DSC behavior of UDP worm. SCANS approach is used to detect the UDP random and sequential scanning while the DSC behavior of UDP worm is detected using worm correlation approach. The machines which exhibit scanning activities (detected

by SCANS approach) will be checked weather its perform DSC behavior or not, the machine can be identified as infected machine if it performs scanning and DSC behavior. We demonstrated the effectiveness of our approach for detecting the presence of UDP worm in the network by evaluating it with simulated dataset obtained from GTNets simulator. The results showed that the proposed approach was sufficiently accurate to detect UDP worm in the network.

## 6. Future work

In future work, we aim to extend our proposed approach for detecting different types of network worm. In addition, we will use the proposed approach as part of a Botnet detection engine, because Botnet utilizes worms for Botnet propagation.

## 7. Acknowledgement

This work was supported by National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia.

## 8. References

1. Yini W, Sheng W, Yang X, Wanlei Z. Modeling the propagation of worms in networks: A survey. Communications Surveys and Tutorials., IEEE. 2014; 16(2): 942–60.
2. Yang W, Gao YP, Zhu ZI, Chang GR, Yao Y. Modelling, analysis and containment of passive worms in P2P networks. Int J Internet Protoc Technol. 2014 Dec; 8(2-3):130–42.
3. Stafford S, Li J. Behavior-based worm detectors compared. 2010; 6307:38–57.
4. Strayer WT, Lapsely D, Walsh R, Livadas C. Botnet detection based on network behavior. Botnet Detection. Springer. 2008; 36:1–24.
5. Anbar M, Manasrah A, Manickam S. Statistical cross-relation approach for detecting TCP and UDP random and Sequential Network Scanning (SCANS). International Journal of Computer Mathematics. 2012; 89(15):1952–69.
6. Beng LY, Ramadass S, Manickama S, Fun TS. A survey of intrusion alert correlation and its design considerations. IETE Technical Review. 2014; 31(3):233–40.
7. Amiri R, Rafsanjani MK, Khosravi E. Black hole attacks detection by Invalid IP Addresses in Mobile Ad Hoc Networks. Indian Journal of Science and Technology. 2014 Apr; 7(4):401–8.
8. Moore D, Paxson V, Savage S, Shannon C, Staniford S, Weaver N. Inside the slammer worm. IEEE Security and Privacy. 2003 Jul-Aug; 1(4):33–9.
9. Stopel D, Boger Z, Moskovitch R, Shahar Y, Elovici Y. Application of Artificial Neural Networks techniques to computer worm detection. International Joint Conference on Neural Networks. Vancouver. BC. 2006. p. 2362–9.
10. Stopel D, Boger Z, Moskovitch R, Shahar Y, Elovici Y. Improving worm detection with Artificial Neural Networks through feature selection and temporal analysis techniques. International Journal of Applied Mathematics and Computer Sciences. 2005 Nov; 1.
11. Farag IA, Shouman M, Sobh T, Forces E, El-Fiqi H. Intelligent system for worm detection. 2010. p. 1–10.
12. Anbar M, Ramadass S, Manickam S, Al-Wardi A. Connection failure message-based approach for detecting sequential and random tcp scanning. Indian Journal of Science and Technology. 2014 May; 7(5):628–36.
13. Berk V, Bakos G, Morris R. Designing a framework for active worm detection on global networks. First IEEE International Workshop on Information Assurance. IWIAS 2003. Proceedings; 2003. p. 13–23.
14. Jung J, Paxson V, Berger A, Balakrishnan H. Fast portscan detection using sequential hypothesis testing. 2004 May 9-12. p. 211–25.
15. Schechter SE, Jung J, Berger AW. Fast detection of scanning worm infections.Springer Link; 2004; 3224: p. 59–81.
16. Gu G, Sharif M, Qin X, Dagon D, Lee W, Riley G. Worm detection, early warning and response based on local victim information. 20th Annual Conference on Computer Seurity Applications. IEEE; 2004 Dec 6-10. p. 136–45 .
17. Moore D, Shannon C. Code-Red: A case study on the spread and victims of an Internet worm. 2002 Nov. p. 273–84.
18. Dainotti A, Pescape A, Ventre G. Worm traffic analysis and characterization. International Conference on Communications, 2007. ICC'07. IEEE; Glasgow. 2007 Jun 24-28. p. 1435–42.
19. Riley GF. Simulation of large scale networks II: large-scale network simulations with GTNetS/ ACM; 2003 Dec. p. 676–84.
20. CAIDA. (2014). The Cooperative Association for Internet Data Analysis. Available from: http://www.caida.org/home/
21. L. L. Massachusetts Institute of Technology. Available from: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html
22. Al-Hammadi Y, Leckie C. Anomaly detection for Internet worms. 9th IFIP/IEEE International Symposium on Integrated Network Management; 2005 May 15–19. p. 133–46.
23. Leckie C, Kotagiri R. A probabilistic approach to detecting network scans. Network Operations and Management Symposium,; IEEE/IFIP; 2002. p. 359–72.