Abstract—on July 19th 2001 “Code-Red” was released into the internet after fourteen hours the worm infected 36,000 hosts. Internet worm procedure that spread autonomously from one host to another cause major problem in today’s networks. Current worm defense begins with manual worm detection followed by damage repair. Worms cause damage to the network by consuming its resources such as bandwidth. Our main factor is detecting traffic signature for stealthy worm. In this paper, we detected signature for unknown internet worm by two techniques part. The first part (FCA) of our technique is concerned with detecting the internet worm and stealthy worm in which computer infected by the worm while the second part (TSA) is concerned with detecting traffic signature for the worm. In this paper, we show that our proposed can detect traffic signature for MSBlaster worm.

I. INTRODUCTION

Active worm spread in an automated fashion and can flood the internet in a very short time. Anti-virus is signature-based technology [5]. Anti-virus compares the file structure to the signatures stored in its database. If the file contain same signature, so it is infected by the worm. The anti-virus database must be updated continuously to detect new worms.

There are few answers to solve the worm attack. One of the solutions to update the anti-virus for detects the worms. Anti-virus can not detect the worm due to its spreading speed. Also, anti-virus can’t detect unknown internet worm automatically because it doesn’t depend on the worm behavior but depend on signature to detect it. Therefore, the anti virus can’t detect most of unknown internet worm automatically, routers and firewalls after configured it can block the packets by traffic signatures, but those happen after the worm spread.

Automatic detection is particularly challenging because it is difficult to prophesy what form the next worm will take. However, automatic detection and response is fast becoming an imperative because a newly released (flash or topological) worm can infect millions of hosts in a matter of seconds.

The technology directed to scrutinize the way of the error message, such as RESET in TCP and ICMP (internet controller message protocol) destination unreachable message.

S. Chen et al. [4] built algorithm to detect the worm, but the rate of false alarms take long time to detect the worm. S. E. Schechter et al. [6] presented a hybrid approach for detecting scanning worms that is integrates significant improvements by two techniques: sequential hypothesis testing and connection rate limiting. This algorithm can detect the internet worm but doesn’t work well on detecting stealthy worm. The threshold can’t reach to detect stealthy worm.

X. Yang et al. [7] built algorithm for detecting the worm which has two sub algorithms, the first algorithm “short term algorithm” run well to detect worm, but the second algorithm “longer term algorithm” can’t detect some types of the stealthy worm. The algorithm also cannot hold any equations to determine specification when the equation runs in the algorithm to detect early worm if it has higher rate for value in average of failure connection. Yang algorithm focus just for detected which computer contain the worm.

X. Jiang et al. [12] proposed algorithm to extract a worm’s behavioral footprint from the worm’s traffic traces. X. Jiang evolution is the number of real worms and their variants confirms the existence of worms’ behavioral footprints and demonstrates their effectiveness in worm identification. Jiang algorithm detect known worm by packet filtering.

II. RELATED WORK

Berk et al. [8] proposed a monitoring system by collecting ICMP “Destination Unreachable” messages generated by routers for packets designated to unused IP addresses. Berk method doesn’t use TCP/RESET failure connection technique.

Staniford Chen et al. [9] proposed an intrusion detection system called “GrIDS”, which can detect worm-infected hosts in a local network through building the worm’s infection graph.

Zou et al. [10] presented “trend detection” methodology to detect a worm at its early propagation stage by using “Kalman filter” base detection, which is robust to background noise in the monitored data.

S. Chen et al. [11] proposal was dependent on an effective approach for detecting network and depend based on the
failure of the connection request received by the network routers. S. Chen proposed distributed anti-worm architecture (DAW) that automatically slows or stops the spread of the worm.

D. R. Ellis et al. [2] proposed the existence of behavioral signatures of worm attacks, focuses on the modeling of inter-machine communication patterns for worm activity detection.

III. DESIGN

In the section we have two types of algorithm the first is detect which computer contain the worm and second algorithm to detect the traffic signature from the computer that infected by the worm.

Our design steps to detect the worm signature as follow:

Figure 1. Design Steps

A. Failure Connection Algorithm (FCA)

This algorithm appoints difference between regular connection and worm connection. The worm scans different IP addresses every second. The algorithm depends on the TCP failure and ICMP unreachable connection on different random addresses. There will be in a large number of failures connections if the computer has worm.

An ICMP “Destination Unreachable” returned only when the IP addresses is unused [2]. See figure 2.

Figure 2. ICMP message

When a SYN packet is sent to a used IP address with destination port closed, TCP RESET packet is returned [2]. See figure 3.

Figure 3. RESET message

FCA only records the number of first failed connection packets such as ICMP and TCP RESET packets that returned from the external destination address to the internal forged and monitored source IP address based in the router (see figure 4). Once detecting the first failed connection packets, the algorithm then extracts (the source address, source port, destination address, destination port) from the packet and creates the record.

Figure 4. Error message returned to router

The “counter1” in the short-term algorithm records the first failed connection packets returned from external Destination IP to the internal Source IP during one minute. In order to faster detecting, two cache tables having “short term cache” and “longer term cache” are introduced by the FCA.

The objective of two sub-algorithms is to faster detecting internet worm by “Counter1” and “Counter2” to detect stealthy internet worm. The counters store the failed connection records in shorter time “Counter1” and longer time “Counter2”. When the shorter term set and term cache counter does not exceed the threshold $\beta$ expires the unit time and the counter’s value of an IP address will be insert into the “longer term” cache table to make increasing computation.
The short term algorithm is described as follows:

Function Short Term algorithm()
    While (received failed connection)
        Source_IP Counter1 = Source_IP Counter1 + 1
        if Source_IP Counter1 >= 9 then "warning worm"
    else
        Counter2 = Counter2 - 1
    End if
    End While
End function

The longer-term is calculation of the failure connection and if received normal connection, like TCP SYN/ACK decreased Counter2. Only the first failed connection sent from the forged source IP address to different destination IP address recorded and normal network activities are considered to decrease the counter’s value. Otherwise, the packet should be ignored when the destination IP is recorded into the table.

The worm scanning activity based on the foundation shows that the counter value of an IP address in the table range from less than the threshold. In this situation, the value of the counter record is inserted into the long-term limit under the table to catch once again.

If suppose $\beta = 100$ Then $X = (1 \ldots 99)$ average of failure connection in one minute.

Threshold in "long term algorithm" can be process by this equation:-

Summation of threshold $= 2^{(6.65 + 0.05054 (\beta - X))}$. The equation is depending on the average of failure connection to compute the threshold. FCA can detect the worm early in usual time. But if can’t detect in early stage, the algorithm provide more time to detect the worm.

$T1 = \text{summation of Threshold} / \text{average of failure connection}$

$T2 = \text{time now} - \text{time start of the algorithm}$

The algorithm is more dynamic to detect the worm. The algorithm can detect the worm by compare $T1$, $T2$, if $(T2$ is smaller or equal $T1$) and $(\text{counter2}$ is greater or equal summation of Threshold), if yes detect the worm. Else check $T1$, $T2$, if $(T2$ is greater than $T1$) then going to feed back and increase the average to give other chance for detecting the worm. If $(T1$ is smaller than $T2$), then forward the traffic, when the counter2 value does not exceed the threshold during time cumulative computation phrase, the traffic sent from the corresponding IP address would be forward as normal activity, (see figure 6).

The longer-term algorithm is described as follows:-

Function Longer Term algorithm()
    While (Source_IP received normal connection)
        Source_IP Counter2 = Source_IP Counter2 - 1
    End While
    if (Source_IP is not in record)
        Source_IP Counter2 = Source_IP Counter2 + 1
        Average of failure connection = Source_IP Counter2/minut
        * Step 0 Summation of failure $= 2^{(6.65 + 0.05054 (\beta - \text{Average of failure connection})}$
        Time to detect $= \text{Summation of failure} / \text{Average of failure connection}$
        If (Time now - Time Start <= Time to detect) AND (Counter2 <= Summation of Failure) then "warning Smith worm"
        else if (Time now - Time Start > Time to detect) then
            Average of failure connection $= \text{Average of failure connection} - 1$
            Goto step 0
        End if
    End if
Else forward traffic
End function

Timer for every three days
For source_IP = First source_IP to last source_IP
    source_IP Counter2 = 0
    Next source_IP
End function

Figure 5. Short Term and Long Term algorithms
B. Traffic Signature algorithm (TSA)

We are detecting worm traffic depend on source IP and source port that was returned by router so that, we can collect the packet by using packets monitor because the worm synchronization is like DNA [12], so we depend on the packet filtering using synchronization of internet worm. The algorithm works when the worm is detected by FCA after that, Signature of internet worm can be detected by using monitor traffic signature. The algorithm captures all the packets with successful synchronization which started from source port. If the port used by the worm will be send request and this request is successful some times if the IP is used and the port is open in the computer, then the worm will have a successful synchronization.

The worm has a successful connection, (see figure 7). Where computer B is accepts the worm from computer A, because the worm generates random IP address in computer A and the same IP was used for computer B. The condition is very important to transfer the worm from computer A to computer B when port for computer B is open.

The algorithm focus on this types of successful traffic synchronization and captures all these packets by monitor. After that, the algorithm will check the destination IP of synchronization is in record. If the destination IP is in record that means there is normal connection because the worm generates the IP address randomly. If not successful, the worm searches for other servers to infect it.

The algorithm of traffic signature algorithm (TSA) will compare the packets and take the maximum successful synchronization of the packets that are similar in traffic synchronization, (See figure 8).

The traffic signature algorithm to detect the traffic signature is described as follows:-

```
Function TSAO

1. While (FCA detect which computer infected by worm)
2. Let.Dest.IP = the destination of IP packet
3. Let.Source.IP = computer that infected by worm
4. Let.Source.Port = the port is used by worm
5. Capture = All packets synchronization start from computer (Source.IP,Source.Port) and (the synchronization was success)
6. While (Not finish Capture (k))
7. IF (Capture (k) not in record) then
8. Success_packets (k) = Capture (k) / k+1
9. End if
10. k = k+1
11. End while
12. For (k) to i
13. For j = 1 to i
14. IF Success_packets (j) = Success_packets (i) Then
15. Success_packets_counter (j) = Success_packets_counter (i) + 1
16. End if
17. Next i
18. Next j
19. Worm_Signature = Success_packets (max number (Success_packets_counter))
20. End While
```

Figure 6. The flow chart of the FCA

Figure 7. Sequence of Infected Worm [12]
In this section we experiment by two algorithms the first is FCA and second is TSA to detect traffic signature for internet worm.

A. Detect Internet Worm

Figures 9 and 10 show two types of stealthy worms are detected by FCA. Figure 9 shows the average of failure connection which is 88/minute, and the time process to detect the worm is 103 sec. In figure 10, the average failure connection is 93/minute, and the time process to detect the worm is 82 sec.

Figures 11 and 12 show two types of stealthy worm processed via X. Yang et al. [7] algorithm. Figure 11 shows the average of failure connection is 88/minute, the time process to detect a worm is 34 min 5 sec Figure 12 shows the average of failure connection is 93/minute, and the time process to detect the worm is 32 min 15 sec.

FCA and X. Yang et al. [7] algorithm detected the two worms but FCA is faster than X. Yang.
Figure 14. X. Yang algorithm can't detected the worm after 30 hours

B. Detect Traffic Signature

Show the experiment synchronization of worm was successful through TSA. The traffic signature is detected for MSBlaster worm.

The worm traffic signature as follow:

\[
\begin{align*}
\text{TCP, 4581/infector, 135/victim, SYN} \\
\text{TCP, 135/victim, 4581/infector, SYN, ACK} \\
\text{TCP, 4581/infector, 135/victim, ACK} \\
\text{TCP, 4581/infector, 135/victim, RST} \\
\text{TCP, 4599/infector, 4444/victim, SYN} \\
\text{TCP, 4444/victim, 4599/infector, SYN, ACK} \\
\text{TCP, 4599/infector, 4444/victim, ACK} \\
\text{UDP, 1552/victim, 69/infector} \\
\text{UDP, 69/infector, 1552/victim} \\
\text{TCP, 4599/infector, 4444/victim, RST}
\end{align*}
\]

V. CONCLUSION

The worm is very fast spread and the techniques to detect the internet worms are slow. Our proposed for detecting the worm and generating the signature automatically. Our results show that the algorithm detected the traffic signature for MSBlaster worm. Also our algorithm is detecting and generating traffic signature for stealthy worm.

VI. REFERENCES


[5] How to fight online Identity, searchsecurity.techtarget.com/