خوارزمية اكتشاف لدودة الانترنت الماسحة التي تستخدم بروتوكول الرسائل المسيطرة للانترنت

د.محمد ماهر رشيد
وزارة العلوم والتكنولوجيا / دائرة تكنولوجيا المعلومات

د.نصير علي حسين
جامعة واسط / كلية التربية

الملخص

دودة الإنترنت هو نوع من البرمجيات الخبيثة (البرامج الضارة) التي تكرر نفسها عن طريق توزيع نسخ من نفسها إلى الشبكة. تقوم دودة الإنترنت بهجمات على الضحايا ذات وجهات مختلفة عن طريق بروتوكول الإنترنت والذي يدعم ثلاثة بروتوكولات رئيسية هي TCP و UDP و ICMP حسب بروتوكول النقل و التحكم. هذا البحث يتناول تصميم خوارزمية التي تركز على هجوم المسح ICMP فقط. ان فحص المسح بتركيز على البروتوكولات المختلفة بخوارزمية محددة تجعل من الصعب الكشف عن الدودة، هناك عدة طرق لاكتشاف دودة الإنترنت. معظم هذه الطرق تعتمد على المسح العام للكشف والتي يطلق عليها في هذا البحث مختصر (GPSD)، وهذا البحث ركز على أسلوب السلوكية لدودة الإنترنت عندما تستخدم ICMP للمسح واكتشاف الحاسبة الضحية بتركيز على الفشل الاتصال الذي يحدث عند استخدام هكذا نوع من المسح . ولكن تقنية GPSD ركزت على اتصال الفشل العام التي يتم تلقي من بروتوكولات مختلفة، وعندما تم مقارنة GPSD والخوارزمية المقترحة في هذا البحث تبين ان الخوارزمية المقترحة لكشف المسح باستخدام ICMP هو اسرع من الاسلوب الذي يعتمد على GPSD.

الكلمات المفتاحية: اكتشاف دودة الانترنت، سلوك الدودة، مسح ICMP

Detection Algorithm for Internet Worms Scanning that Used Internet Control Message Protocol

Dr.Mohammad Maher Rasheed
Ministry of Science and Technology/ Information Technology

Dr.Naseer Ali Hussien
Wasit University/ Education College

Abstract

An Internet worm is type of malicious software (malware) that self-replicates and distributes copies of itself to its network. The Internet worm attacks different destination victims by a used Internet protocol that support three main protocols TCP, UDP and ICMP regarding transport and control protocol. This research designed algorithm that focused on ICMP protocol scanning attack only. When the algorithm focuses on the different protocol, the detection will be difficult; there are several detections for internet worm. Most of these detections are depending on General Protocol Scanning Detection (GPSD), our technique focused on behavioral of internet worm when the research used ICMP scanning and failure connection for ICMP protocol. However, GPSD technique focused on general failure connection that received by different protocols. When the research compared GPSD and proposed algorithm. The research found the proposed algorithm is faster detection when the worm used ICMP scanning than technique that depends on GPSD.

Keywords: *Internet worm detection, behavioral worm, ICMP scanning.*

Introduction

Internet worms are programs that self-propagate across a network by exploiting security or policy flaws in widely-used services. It does not need to be part of another program, but self-contained [1]. Internet worms are independent when spreading themselves across the Internet. They break into computers, and copy itself without human assistance and user's knowledge. It infects a new host by searching another vulnerable host in the network. Once a new vulnerable host is found, the worm infects the other host [2]. Many works have been done on detecting unknown worm, but this field is still more challenges [3]. Welchia is Internet worm spread on August 18, 2003 and systems affected are Microsoft IIS, Windows 2000 or XP. It used ICMP request scanning to find the victim, when the victim replies with ICMP reply, after that, the infector machine sent packets by using TCP/ SYN on port 135. Welchia attempts to exploit the previously discussed [4, 5]. The victim IP address is selected by two different ways. The first way, the worm uses the IP from the infected machine W.X.0.0 where W and X from the infector machine IP address. The second way, it will construct a random IP address A.B.0.0, where A and B are randomly generated. After selecting the start address, the worm uses sequential address scanning to find the victim [6].

Most of Internet worms techniques find the victims by depend on blind scanning that is based on chance and has a relatively high failure connection rate. Furthermore, the worm can use *Transmission Control Protocol* (TCP), *User Datagram Protocol* (UDP), and *Internet Control Message Protocol* (ICMP) for scanning.

The ping or ICMP scan sends a single ICMP echo request from infector to the victims. The victim responds from an active device will return an ICMP echo reply, unless the IP address is not available on the network or the ICMP protocol is filtered.

A response from a victim will return an ICMP echo reply as shown in Figure 1, unless the IP address is not available on the network or ICMP is filtered [7].

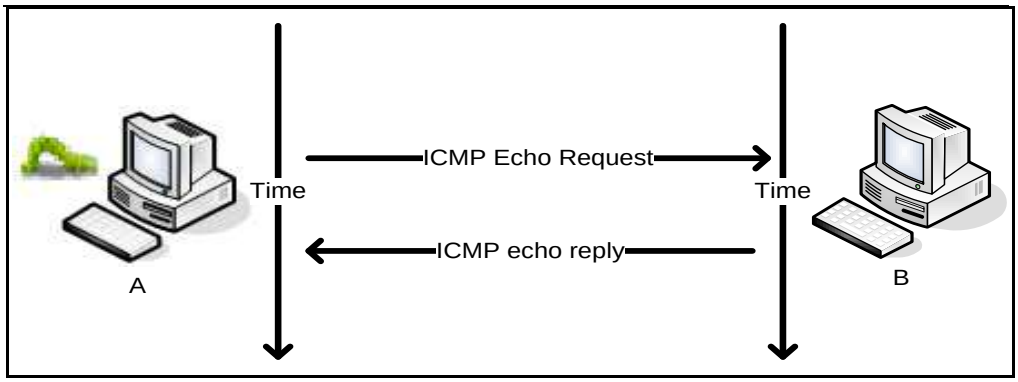


Figure 1: ICMP Echo reply

If the IP of the victim is not available on the network or a packet filter is preventing ICMP packets from passing, there will be no response to the echo frame [7]. See Figure 2.

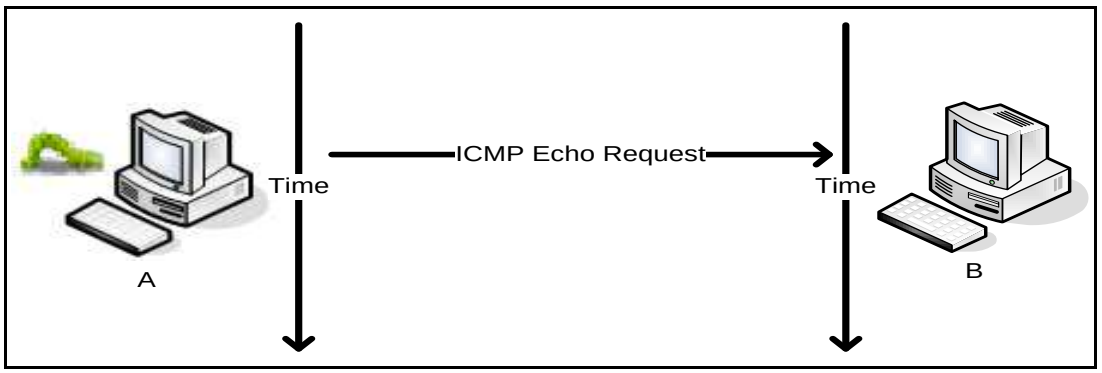


Figure 2: ICMP Echo Request

When the IP is unused in the victim, the router replies ICMP Unreachable Host to the infector machine. See Figure 3.

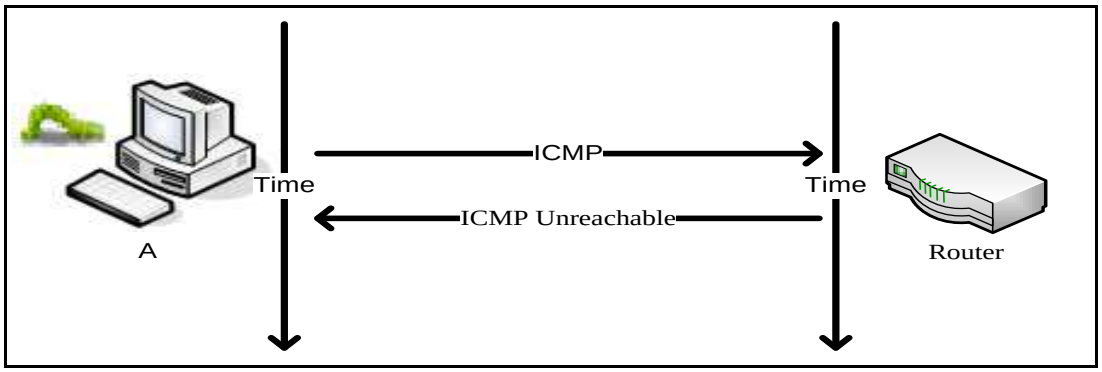


Figure 3: ICMP Request Status When Destination IP is Unused

If a host or router discards a packet due to a time-out, it will generate an ICMP Time Exceeded [7]. See Figure 4.

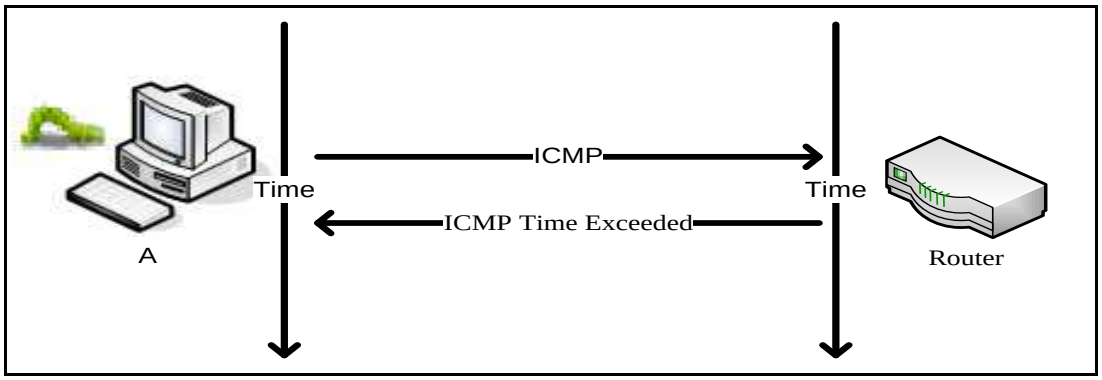
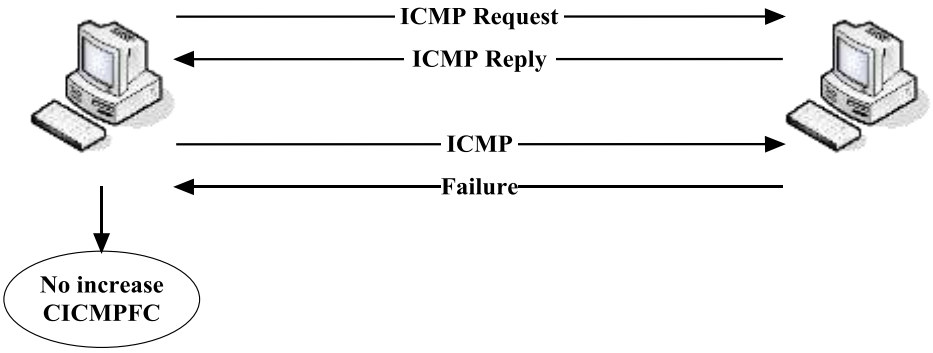


Figure 4: ICMP Request Status When Destination IP is not Responded

The Design of Detection Algorithm for ICMP Scanning Worms

Algorithm Detection for ICMP Scanning Worms (ADICMPSW) works to detect ICMP scanning worm. ADICMPSW works to detect the worm when checking ICMP request from source IP address to the different destination IP address, and the reply is ICMP reply or a failure connection. ADICMPSW depends on failure connections to detect the Internet worm in infected machine. ADICMPSW has a Counter of ICMP Failure Connection (CICMPFC), but the ADICMPSW does not consider all the failure received. ADICMPSW ignores the failure connection when the destination IP address is in the History of ICMP Connections (HICMPC), as shown in Figure 5.



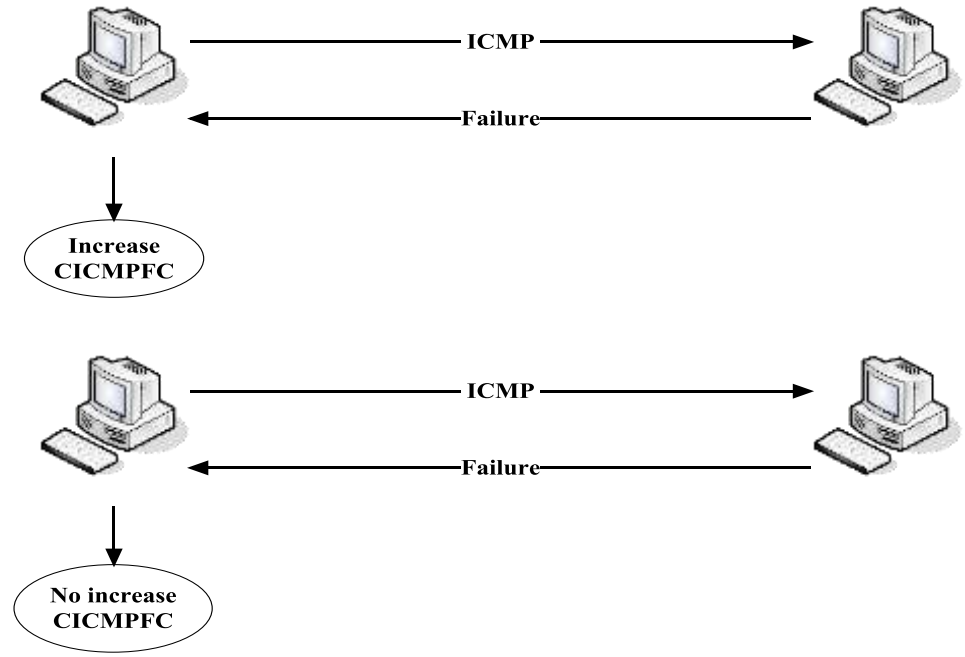


Figure 5: Use Case for ICMP Failure Connection

In Figure 6 shows the sequence diagram for the ICMP requests that are represented in ADICMPSW, the six states are as follows:

Scenario 1: If machine A has a regular connection (ICMP request, ICMP reply) with machine B, in this case, ADICMPSW inserts the IP address for machine B in the HICMPC.

Scenario 2: If machine A receives a failure connection from machine B, and the destination IP address for machine B is in the HICMPC, ADICMPSW does not consider this failure connection.

Scenario 3: If machine A sends a request to machine C and after that receives a failure connection, and the IP address for machine C is not in the HICMPC, ADICMPSW increases CICMPFC and inserts the IP address for machine C in the HICMPC.

Scenario 4: If machine A receives a failure connection from machine C, and the IP address for destination IP address is in the HICMPC, ADICMPSW

does not consider this failure connection. ADICMPSW does not decide any procedure for this failure.

Scenario 5: If the machine A sends ICMP request to machine D, ADICMPSW inserts the destination IP address to the Record of ICMP is Not Responded (RICMPNR) that includes (Destination IP, Source Port and Destination port), while is not responded after three second, ADICMPSW removes the record of destination IP address record from RICMPNR and inserts the destination IP address only in HICMPC with an increasing CICMPFC.

Scenario 6: If the machine A sends ICMP request to machine E and after that sends another ICMP request before three seconds, the first request is inserted to RICMPNR, and the second request is ignored by ADICMPSW. Every ICMP request is saved on RICMPNR when the destination replies. ADICMPSW removes the request from RICMPNR and inserts the destination IP address on the HICMPC.

ADICMPSW does not consider any request that included in HICMPC or RICMPNR except the request that is not included in HICMPC or RICMPNR. Moreover, ADICMPSW considers only one request to the destination IP address and saves it in the RICMPNR.

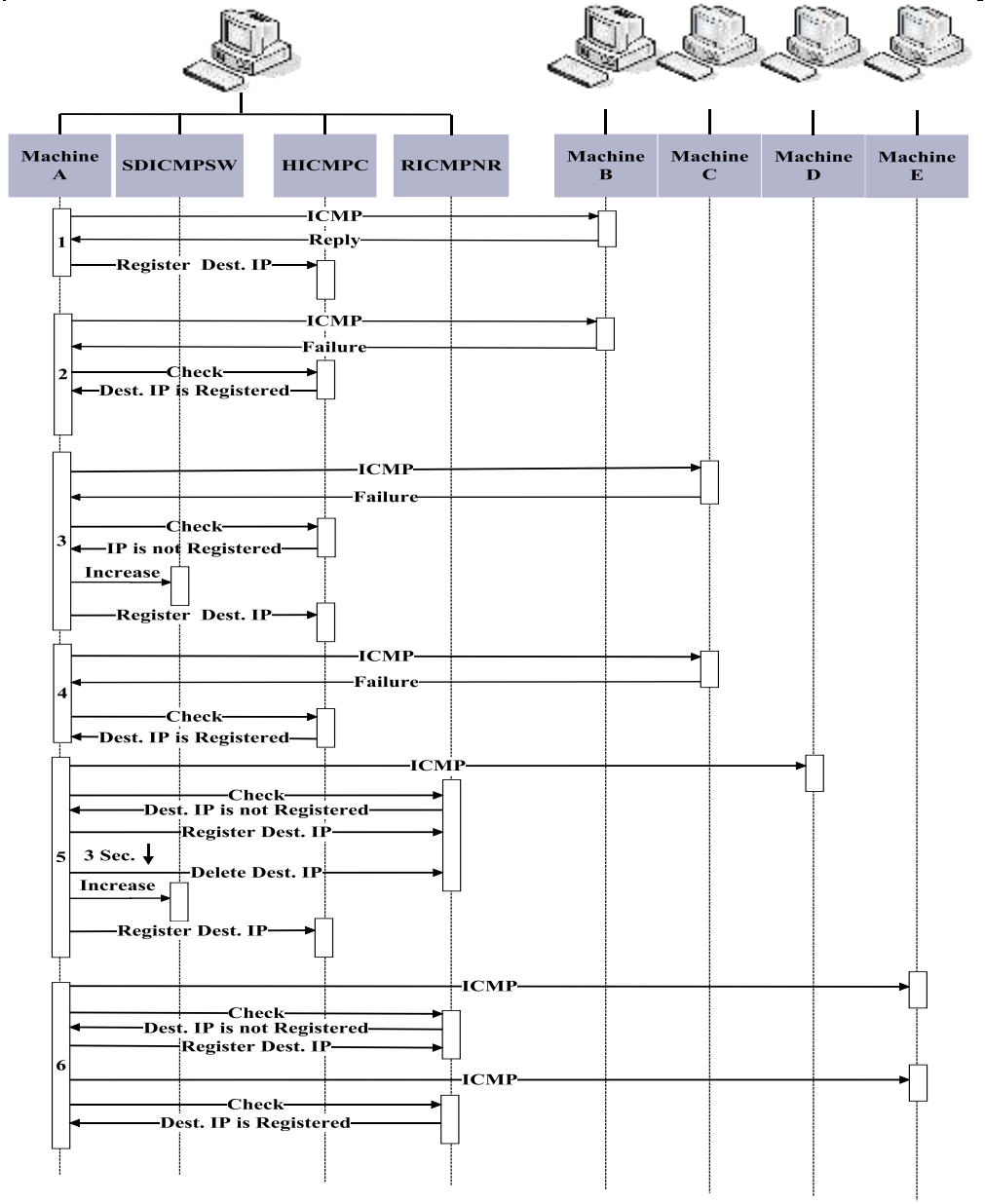


Figure 6: Sequence Diagram of ICMP Failure Connection

Whenever, ADICMPSW reads a packet and ICMP packet replies from destination IP address to source IP address, in this case, ADICMPSW makes sure that the packet came from destination IP address to reply for ICMP request by checking RICMPNR includes the destination IP address equals the source IP for the packet that received. If yes, it is meaning a reply for the request.

After that, ADICMPSW considers four reply messages from the destination when sending ICMP request from the infected machine, one of the connections is a normal connection, which is ICMP Type eight and receives ICMP Type zero, but there are three types of failure connections. The first failure connection is received when the worm sends ICMP request, and the reply is ICMP Unreachable. This reply is received when the destination port is closed or destination IP is unused. The second failure connection receives ICMP Time Exceeded when the destination IP is not responded. The last failure connection is not receiving any a reply from the destination machine when the port of destination IP is filtered, as shown in Figure 7.

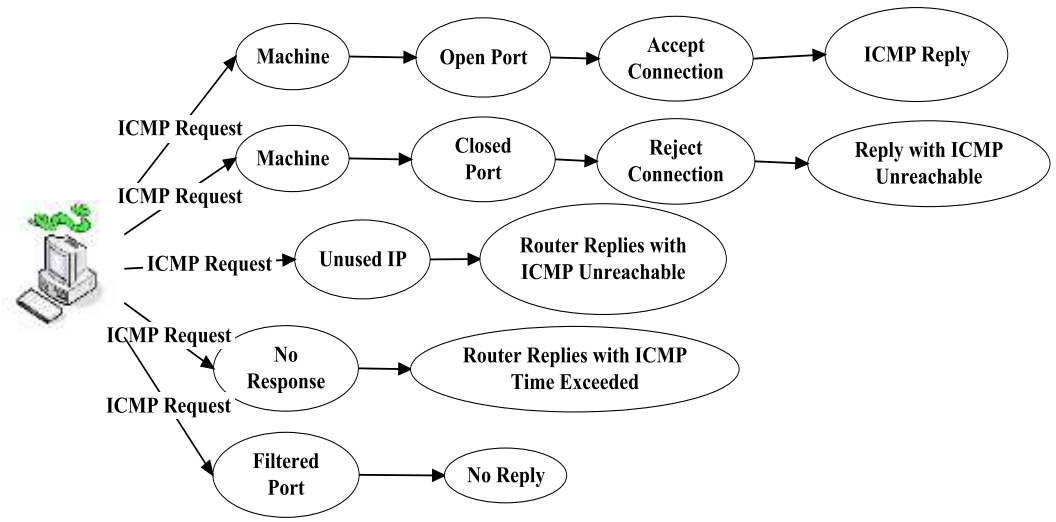


Figure 7: Use Case Diagram for ICMP Worm Request States

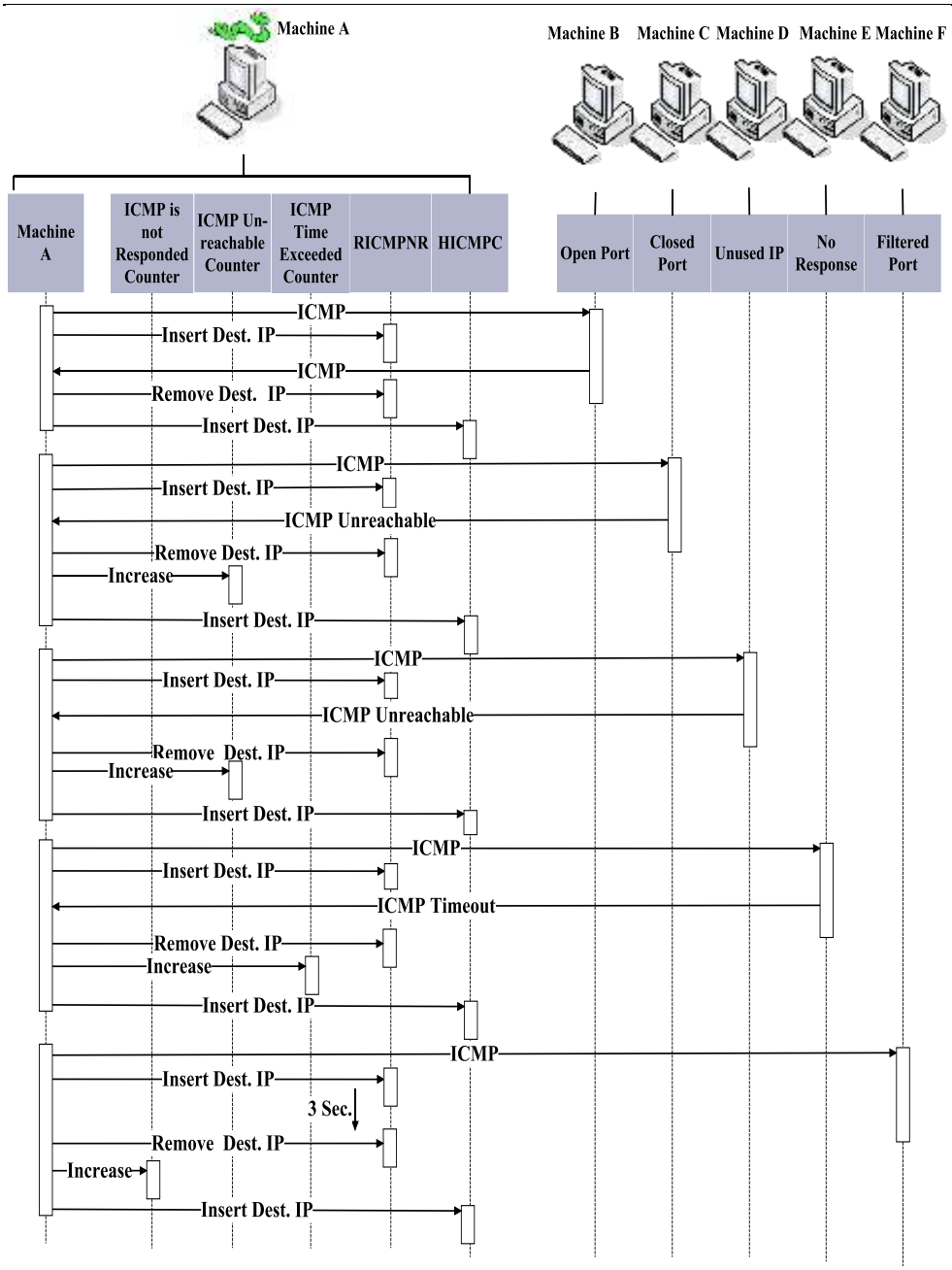


Figure 8 shows the five states for the reply to the request of Internet worm.

State ICMP reply: If the Internet worm in machine A sends ICMP request to machine B, ADICMPSW inserts the destination IP (machine B) to the RICMPNR. The destination IP address is used by machine B and the port for machine B is open, machine B replies ICMP to machine A. ADICMPSW removes the destination IP address record from the RICMPNR and inserts it in the HICMPC.

State ICMP Unreachable: If the Internet worm in machine A sends ICMP request to machine C, ADICMPSW inserts the destination IP (machine C) to the RICMPNR. The IP is used but the port for machine C is closed, machine C replies ICMP Unreachable, after that, ADICMPSW increases ICMP Unreachable counter. ADICMPSW removes the destination IP address record from the RICMPNR and inserts it in the HICMPC. Moreover, if the Internet worm in machine A sends ICMP request to machine D, ADICMPSW inserts the destination IP (machine D) to the RICMPNR. The IP is unused in machine D, after that, machine A receives ICMP Unreachable. ADICMPSW increases ICMP Unreachable counter. ADICMPSW removes the destination IP address record from the RICMPNR and inserts it in the HICMPC.

State ICMP Time Exceeded: If the Internet worm in machine A sends ICMP request to machine E, ADICMPSW inserts the destination IP (machine E) to the RICMPNR. When the port for destination IP is filtered, machine A receives ICMP Time Exceeded from the router when the machine does not reply. ADICMPSW increases ICMP Time Exceeded counter. ADICMPSW removes the destination IP address record from the RICMPNR and inserts it in the HICMPC.

State ICMP is not Responded: If the Internet worm in machine A sends ICMP request to machine F, ADICMPSW inserts the destination IP (machine F) to the RICMPNR. When the port for machine F is filtered, machine A does not receive any reply. After three seconds, ADICMPSW increases 'ICMP is not Responded' counter. ADICMPSW removes the destination IP address record from the RICMPNR and inserts it in the HICMPC. After that, CICMPFC is calculating the total for four counters. The equation is as follows:

TICMPFC= ICMP Unreachable + ICMP Time Exceeded +ICMP is not Responded (1)

TICMPFC checks if the threshold reaches to 101. If true it means the machine is infected as shown in Figure 9.

Figure 8: Sequence Diagram for ICMP Worm Request States

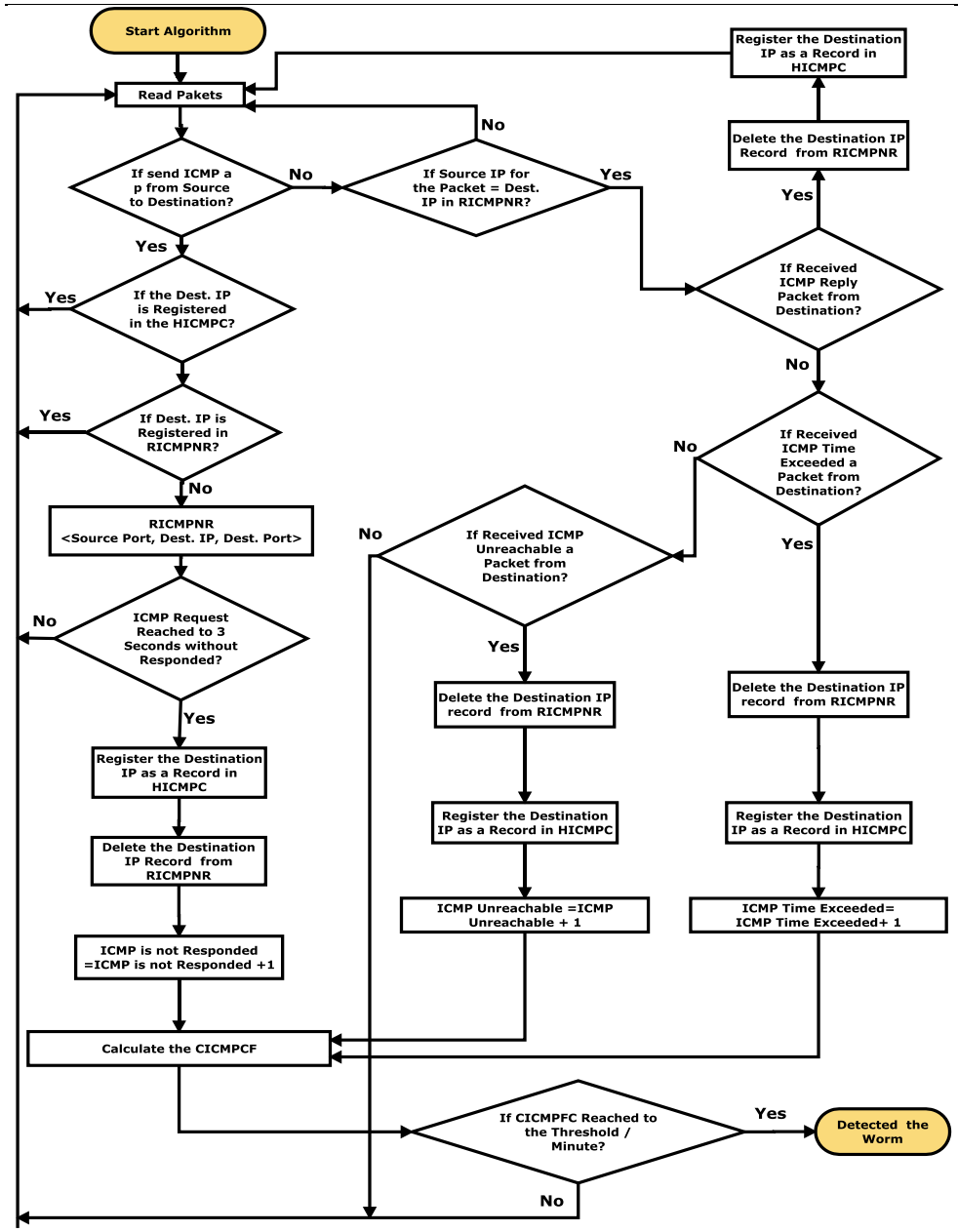


Figure 9: Flowchart Diagram for ADICMPSW

Setup for Evaluation and Validation

For Microsoft's worm setup, the machine operating system used was Microsoft 2000 professional Service Pack 4. Moreover, the machine was connected with a network device by Celcom that supports the Internet by mobile wireless, and the broadband speed was 3.6 MBps, as shown in Figure 10. The infected computer was installed with ADICMPSW. In the evaluation, other techniques that were compared with the proposed technique were also installed on the infected computer.

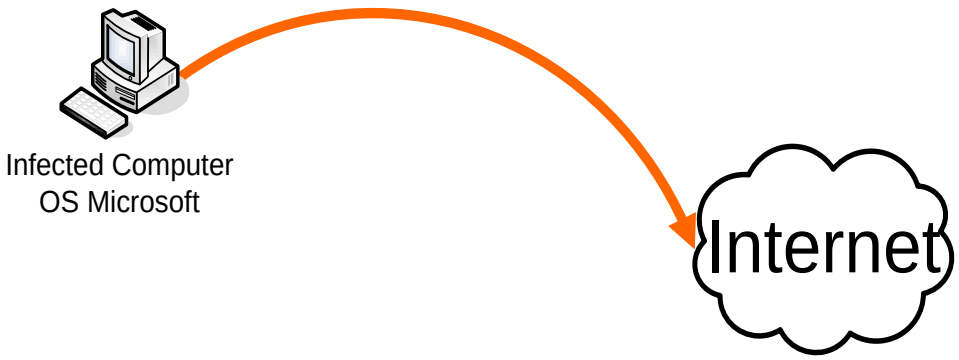


Figure 10: Setup for Detecting Microsoft Scanning Worms

Experiment Results

The study used Xiong et al. [8] algorithm and ADICMPSW to detect a Welchia worm. The maximum failure connection recorded in short term algorithm was 56 failure connections per minute for 684 records, as shown in Figure 11. However, the long term algorithm reached to more than 3000 failure connections after 684 minutes as shown in Figure 12. The Xiong et al. [8] algorithm detected the worm depending on a long term algorithm, but the short term algorithm failed to detect Welchia worm. ADICMPSW detected Welchia worm after eight seconds only, as shown in Figure 13. The So the proposed ADICMPSW algorithm is faster than Xiong et al. [8] algorithm.

Minute

Figure 11: Short Term Algorithm Tried to Detecting Welchia Worm

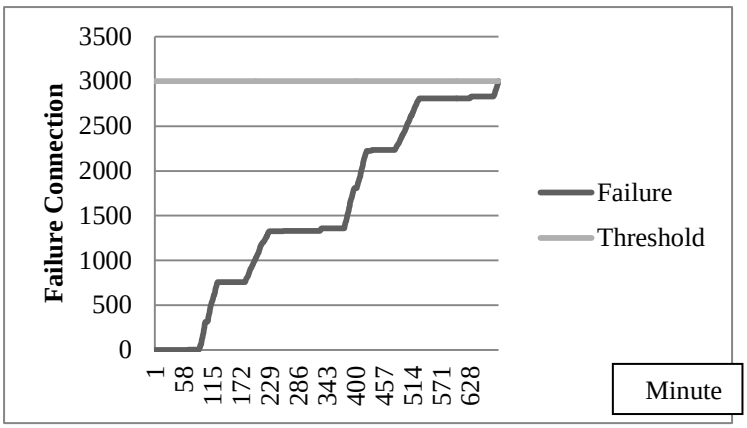


Figure 12: Long Term Algorithm Detected Welchia Worm

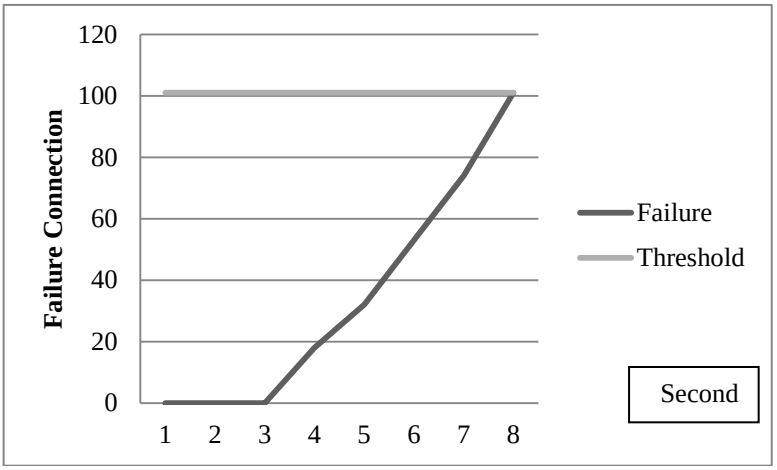


Figure 13: ADICMPSW Detected Welchia Worm

References

[1] *N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, "An Taxonomy of Computer Worms," in Proceedings of the 2003 ACM Workshop on Rapid Malcode, 2003, pp. 11 - 18.*

[2] *D. Seeley, "A Tour of the Worm," in Proceedings of 1989 Winter USENIX Conference, San Diego, 1989, pp. 287-304.*

[3] *L. Pele, M. Salour, and S. Xiao, "A Survey of Internet Worm Detection and Containment," IEEE Communications Surveys and Tutorials, vol. 10, pp. 20-35, 2008.*

[4] *S. E.Eugene, "The MSBlaster Worm: Going from Bad to Worse," Network Security, vol. 2003, pp. 4-8, 2003.*

[5] *C. Wong, S. Bielski, A. Studer, and C. Wang, "On the Effectiveness of Rate Limiting Mechanisms," in 8th International Symposium on Recent Advances in Intrusion Detection 2005.*

[6] *F. Perriot. (2007). W32.Welchia.Worm. Available: http://www.symantec.com/security_response/writeup.jsp?docid=2003-081815-2308-99&tabid=2, [Accessed: 17th January 2011].*

[7] *J. Messer, Secrets of Network Cartography: A Comprehensive Guide to Nmap: http://www.professormesser.com/, 2007.*

[8] *Y. Xiong, L. Jing, Z. Yuguang, and W. Ping, "Simulation and Evaluation of a New Algorithm of Worm Detection and Containment," in Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, 2006, pp. 448-453.*