ALGORITHM FOR DISTRIBUTED AGENT BASED NETWORK INTRUSION DETECTION SYSTEM (NIDS)

Aleksandar Sokolovski
Faculty Of Informatics, European University
Skopje, Macedonia

Saso Gelev
Faculty Of Informatics, European University
Skopje, Macedonia

ABSTRACT

The scope of this research paper is one of the most important aspects nowadays, the security and management of one computer network (methods and procedures to get a stable, reliable and redundant computer network) which is a key issue for any ICT Enterprise in this world of Information Age. This paper attempts to investigate the possible benefits of using the network security methods in combination with medical quarantine procedures, in order to create new algorithm for network intrusion detection system (NIDS).
The proposed algorithm which will be more effective, then the previous NIDS before in stopping multiple attacks/intruders, due to the usage of combined network security, distributed agent based calculation and quarantine. The medical quarantine procedures based on NIH CDS (National Institute for Health and Center for Disease Control in USA) will be used for isolating and identifying the "infected" computer, thus making the algorithm even better. The primary objective is to identify and verifying the best possible integration of network security and quarantine methods into an algorithm for NIDS. The main aim is to test the proposed algorithm for NIDS for efficiency and effectiveness. This will be achieved by testing the algorithm with the collection DARPA DATASET'99.

Keyword: intrusion detection system, network security, agent based security sensors, distributed calculation MPI.NET, medical quarantine procedures.

I. INTRODUCTION TO IDS

An intrusion detection system (IDS) monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. In some cases the IDS may also respond to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network.IDS come in a variety of "flavors" and approach the goal of detecting suspicious traffic in different ways. There categories are: NIDS, HIDS, Signature Based, Anomaly Based. [1]

A. NIDS

Network Intrusion Detection Systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. Ideally you would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall speed of the network.

B. HIDS

Host Intrusion Detection Systems are run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator of suspicious activity is detected.

C. Signature Based

A signature based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to your IDS. During that lag time your IDS would be unable to detect the new threat.

D. Anomaly Based

An IDS which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline will identify what is "normal" for that network- what sort of bandwidth is generally used, what protocols are used, and etc.

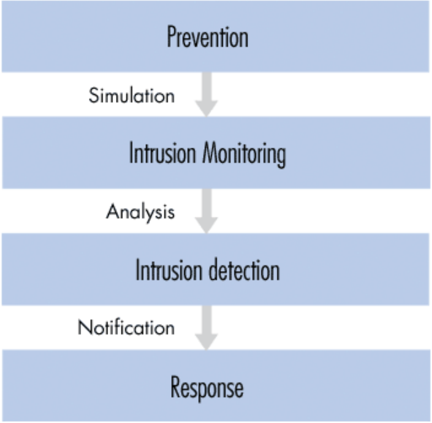


Figure 1. Structure of an Intrusion Detection System

E. Snort

One of the most well-known and widely used intrusion detection systems is the open source, freely available Snort. It is available for a number of platforms and operating systems including both Linux and Windows.
Snort is a signature based IDS, lightweight and very easy to use, the code is 100 KB.

Snort architecture consists of 4 main parts: Packet Decoder, Pre-processing, Detection Engine, Post-Process.

II. THE ALGORITHM FOR THE PROPOSED ISP

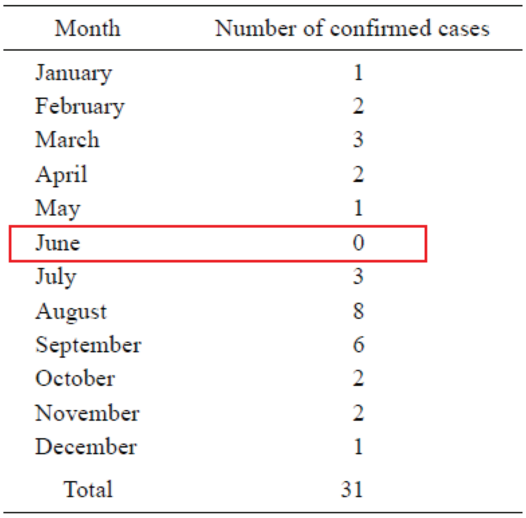
Solutions for IDS are many, our solution is a hybrid version of an, network based / host based IDS and signature based and anomaly based IDS. Our system or the algorithm presented for managing the proposed ISP system on figure 3, is based upon HIN procedure, presented in the next section

A. Procedure for epidemic containment and control created by NIH (National Institute of Health)

Procedure for epidemic containment and control
Quarantine procedure (NIH POLICY MANUAL, 3043-1)

- Dislocating VIP persons from the quarantine zone
- Isolating the sick from the healthy patients
- Immunization of the healthy patients
- Creating Quarantine Zone
- Detecting "patient zero"
- Eliminating the threats

This procedure is used in medicine, and is proven as a very successful procedure in 2009, tested for the H1N1 virus containment in North Carolina, U.S.A. This is also known as procedure CDC H1N1.
Many methods for containment were tested in June the above mention procedure was implemented and the containment of the virus was 100% or 0 newly exposed patients.
The results can be seen on figure 2, results procedure CDC H1N1.



Month	Number of confirmed cases
January	1
February	2
March	3
April	2
May	1
June	0
July	3
August	8
September	6
October	2
November	2
December	1
Total	31

Figure 2. Results procedure CDC H1N1

III. UML DESIGN FOR THE ISP

In this section we will present the implementation of the ISP system, and the proposed algorithm that manages the modules of the ISP and their communication.
The entire ISP is not presented in details, only the parts / modules that will be same for the any implementation to any type of computer network, as a separate system of add-on to an existing one.

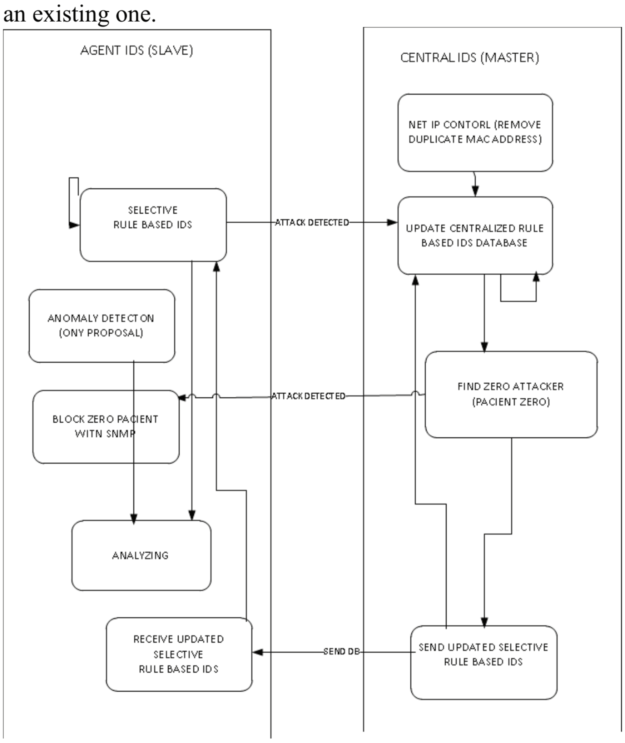

AGENT IDS (SLAVE)
CENTRAL IDS (MASTER)
NET IP CONTORL (REMOVE DUPLICATE MAC ADDRESS)
SELECTIVE RULE BASED IDS
ATTACK DETECTED
UPDATE CENTRALIZED RULE BASED IDS DATABASE
ANOMALY DETECTON (ONY PROPOSAL)
BLOCK ZERO PACIENT WITN SNMP
ATTACK DETECTED
FIND ZERO ATTACKER (PACIENT ZERO)
ANALYZING
RECEIVE UPDATED SELECTIVE RULE BASED IDS
SEND DB
SEND UPDATED SELECTIVE RULE BASED IDS

Figure 3. Elements and modules of the ISP

A. Detecting duplicate MAC address in one LAN network

In one network if there is a duplicate MAC address, it can only be detected if there is one physical network with one VLAN's. If there are multiple VLAN's or multiple address pools with one or many DHCP services, the detection of duplicate MAC addresses cannot be detected.
Our Proposed solution is every IDS slave, (One IDS slave host is placed on every physical switch or VLAN, this is a strategic decision in order to get host and network type of IDS) using the ARP protocol to map all of its neighbor host PC's. The list of all the IDS slaves are send to the master IDS. Then the master makes one jointed list, and the possible duplicate MAC addresses are detected. If One MAC address is in many network than that is an virtual interface made by some type of WORM. Form this list the preserved IP/MAC addresses are ignored, like broadcast.
Then the MAC address is blocked on ALL VLAN's or address pool's, using SNMP TRAP.

B. Selective DB with attacks

One of the universal or unique solutions of this paper is the selective Database (DB) with attacks. The entire DB with signatures of attacks is kept at the master IDS.

Locally at all the IDS slaves only a selective DB are being kept. The algorithm for exchange is LRU.

The network traffic is recorded in by the slave IDS, the packet are copied, and are converted in understandable form for the our ISP. The data is first converted from HEX to ANCII, by using HEX to ANCII decoder. Later on the value of every variable from the IP header for each packet is written in a predefined class IP packet or an object is derived from that class IP packet.

If the packet matches one of the rules / signatures of attacks from the selective DB in the slave IDS.

If the packet does not match any of the signatures of attacks from the selective DB in the slave IDS, the packet is then send to the slave IDS for more detailed analyses.

The difference is that when the IDS slave detects an attack, it can be block there is time for that action, therefore the slave IDS is the Intrusion Prevention System.

Using analogy from the medical quarantine procedure (mentioned in this paper above) the slave IDS make so called vaccination of their host neighbors, this is an important task in order to prevent the network attack from spreading in other parts of the network.

The detected signature in the slave IDS is put on the top of the signature based DB and send update for the number of detected attacks from those signature to the master IDS, and the master IDS updates the selective DB of all the slave IDS with the new information.

The used algorithm for exchange of rules in LRU least recently used, the reason for choosing this algorithm and not FIFO or LIFO, are explained in more details in appendix 4 of [2].

The master IDS compare the packet with all of the signatures in the database, so the attacks will be detected after it will have happened. Using sort function, the packet are sorted according to the logic first the LAN, and then WAN because the LAN packet are “faster” then the WAN packets. With this methodology and with replacing all the public IP addresses with the LAN IP address of the default gateway using (Reverse NAT), the attacks are detected faster for the fastest packets.

The experiment for the sort algorithm and which sort algorithm for this module of the ISP is chosen is explained in more details in Appendixes [2], and in section 5 (Experiment) in this paper.

C. Finding Patient Zero

In scenario of multiple attacks or medically an “epidemic outbreak”, the module for finding patient zero is used; the detailed explanation for this part is presented in this section.

From implementation aspect every switch has a slave IDS and analyses the traffic only for the host on the same switch. With this implementation the number of computers / host in the network will not affect computational speed of the IDS, this is solution for the problem of any network based IDS, that have slower computational speed with the increase number of new hosts in the network.

Our proposed algorithm for the ISP also does not the disadvantage of any Signature based IDS, by using a selective database for the signature based attacks in the slave IDS part of the ISP.

In a classical signature based IDS with time the number of rules in the signature DB increases and with that the computational speed/time of the packets, and so the possible attacks are detected more and more later. So with time the classical signature bases IDS are losing their efficiency.

Our algorithm in the slave IDS part of the ISP uses fixed number of signature.

The algorithm for exchange of rules between the master and slave IDS’s is Least Recently Used, explained in more details in appendix 4 of [2]. In section 5 (Experiment) of this paper, an experiment is presented which determines the right or optimal number of rules that should be kept in the selective signature DB.

Using this DB the slave IDS’s compare the packet for possible attacks.

The packet marked as attacks by the slave IDS are send with TIME STAMP of the detected attacks are then send to the master IDS for detailed analyses.

All the same attacks with different IP sender and IP receiver packers are put into a single array, the array is sorted from smaller to larger according to the value of the variable TIME STAMP. The IP sender address of first element in the array is the zero patient or the first attacker.

That IP address is BLOCK from the network and its associating MAC address, this tasks is performed using SNMP TRAP.

IV. DARPA DATASET’99

In 1998 and 1999 The Information Systems Technology Group of MIT Lincoln Laboratory [3] with the support of the Defense Advanced Research Projects Agency [4] and the Air Force Research Laboratory (all from the USA), had worked on a new innovative experiment in the field of intrusion detection systems.

They had done a cutting edge experiment for the time, creating an Intrusion detection system that monitors the state of an active computer network, looking for some form of attack like denial of service, form of abuse like unauthorized usage, or rear and strange behavior like some forms of so called anomalous behavior.

The experiment was set in a real military base with real computers, but the attack were simulate (it was known what was attack what was a normal connection, this was used later on to evaluate the effectiveness).

The experiment in 1999 (1999 DARPA Intrusion Detection Evaluation Data Set [5]) was small improvement on the experiment done in 1998 (1998 DARPA Intrusion Detection Evaluation Data Set), the main difference is that the 1999 data set contains 56 type of attacks and the 1999 data set only 24 (types of attacks).

In the 1999 the “simulated” attacks lasted 5 weeks, the first and third week was normal traffic, the second week

Contained Labeled Attacks. The attacks were divided into five main categories: Denial of Service Attacks, User of Root Attacks, and Remote to Local Attacks, Probes and Data. The full list of attacks is presented on [6].

Then the system was tested with random network packets (some attacks, some normal traffic), there were 201 instances of about 56 types of attacks distributed throughout these two weeks. At the time the main purpose of the experiment was creating the intrusion detection system, but the real “hided” value of this experiment was the 1998/1999 DARPA Intrusion Detection Evaluation Data Set. The collected data (audit data for many operating systems including Windows, Linux and Sun Solaris, and TCP dump data) from this were made available for all the researchers that needed a test data set for their intrusion detection system. This data set had made possible the creation of many future intrusion detection systems. Proof of the value of this data set is the number of publications using this data set in their research project, like publications [7].

This is the reason why we intended to use the 1998/1999 DARPA Intrusion Detection Evaluation Data Set.

The 1998/1999 DARPA Intrusion Detection Evaluation Data Set, is a very data set containing around 4-5 GB of data, our main purpose was testing our intrusion detection genetic algorithm, so in order to minimize the time for analyzing the data set and maximizing the testing type, we used the optimized versions of the DARPA data set, the KDD CUP 99 Data Set. Detailed analyses of the KDD CUP 99 Data Set, is presented in [8].

KDD CUP 99 Data Set is compiled from the 1998 DARPA Intrusion Detection Evaluation Data Set, but optimized for the Third International Knowledge Discovery and Data Mining Tools Competition, which was held in combination with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. The main task of the competition (Third International Knowledge Discovery and Data Mining Tools Competition) was to create a network intrusion detector, a so called predictive model which will be capable of making a difference between a network attacks and normal network connections. The database contains normal connections and 24 types of attacks, the types of attacks are presented on [9] (the database is based on the data from the 1998 DARPA Intrusion Detection Evaluation Data Set, as mentioned previously).

Evaluation on the KDD CUP 99 Data Set and Summary report with type of attacks and number of connections for each type of attacks is presented on the table in [10]. For this paper and our algorithm it is very important to present the KDD CUP 99 Data Set Schema properly and precisely, for this we will use the tables from the tasks for the KDD CUP 1999 [11-12] (table 1 presented in [2]).

V. EXPERIMENT

The main purpose of this experiment is to test the effectiveness of the prososed ISP algorithm, using dynamic number of selective rules in the IDS sensors (intelligent agent IDS).

A. *Materials used*

Hardware: 1 desktop PC with the following components, 1 virtual machine on the same PC (using from the Host Machine: 1 core of the CPU, 1 GB RAM)

- 3 GHz CPU (Intel Core 2 Duo E7500)
- 3GB RAM DDR2 (400 MHz),
- Motherboard Intel P43, System Bus 800 MHz
- HDD 160 GB (SATA 2, @7200 RPM)

B. *Methods*

First the DARPA DATASET’99 training collection is used to define the rule set of attacks in the central IDS, then using the distributed agents and the DARPA DATASET’99 training collection the algorithm is tested for the most optimal number of attack rules (minimal number, maximum number of detected attacks). The test is reputed multiple times using different number of attack rules in the distributed database (the attack rules are replaces using LRU).As a reference for indenting is the packet an attack or not, we use SNORT as a 99, 9 % effective signature based IDS system.

C. *Data and Results*

Number of attack rules in Distributed Agents	20	40	60	80	100	120	140
% of attacks detected by the proposed algorithm for IDS	11, 25 %	36, 41 %	46,71%	63,57%	65,39%	68,92%	69,12%

Table 1: Number of rules in Distributed Agents, % of attacks detected

D. Discussion or Analysis

From the analyses we can see that the most optimal Number of rules in Distributed Agents for this dataset is 80, because below 80 the % of detected attacks is very small, and above 80% of detected attacks increase slowly.

VI. Conclusions and Future Work

These results are only valid for this dataset, for different dataset it might be that different optimal Number of attack rules in Distributed Agents is needed.

References

[1] Andrew S. Tanenbaum, *Computer Networks,* 4th Edition

[2] Aleksandar Sokolovski, Master Thesis "Algoritthm for dymanic management of multiple attacks scenarios in Local Computer Network", 2011 Faculty of Informatics, European University – Skopje Republic of Macedonia.

[3] MIT Lincoln Laboratory, http://www.ll.mit.edu/about/about.html,

[visited: 2010/06/18]

[4] Defense Advanced Research Projects Agency, http://www.darpa.mil/, [visited: 2010/06/18]

[5] 1998/1999 DARPA Intrusion Detection Evaluation Data Set,
http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1999data.html,
[visited: 2010/06/18]

[6] Intrusion Detection Attacks Database,
http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attackDB.html,
[visited: 2010/06/18]

[7] MIT Lincoln Laboratory, Cyber Systems And Technology Publications,
http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/pubs.html,
[visited: 2010/06/18]

[8] KDD CUP 99 Data Set, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html,
[visited: 2010/06/18]

[9] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani,
A Detailed Analysis of the KDD CUP 99 Data Set, Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)

[10] KDD CUP 1999 Training Attack Types,
http://www.sigkdd.org/kddcup/site/1999/files/training_attack_types,
[visited: 2010/06/18]

[11] KDD-Cup 1999 Data Evaluation,
http://matauranga.wordpress.com/rana/kdd-cup-1999-data-evaluation/,
[visited: 2010/06/18]

[12] KDD Cup 1999: Tasks,
http://www.sigkdd.org/kddcup/index.php?section=1999&method=task,
[visited: 2010/06/18]