# AN ALGORITHM TO IMPLEMENT DYNAMIC ACCESS CONTROL USING ANOMALY BASED DETECTION WITH VLAN STEERING

Shalvi Dave, Dr. Bhushan Trivedi

## ABSTRACT

Intrusion Detection and Prevention Systems, IDPS, are mature network level defenses deployed in thousands of computer networks worldwide. The basic difference between detection and prevention technique lies in how it provides protection for network environments. An IDS monitors logged data and compares it with attack signatures to detect unwanted access. For such identification, IDS normally uses signatures or any unique characteristics of such attacks.

In this paper, we have designed an algorithm to achieve dynamic access control. Dynamic access control requires implementation of three functionalities: traffic monitoring, validation and policy enforcement. In this algorithm, traffic monitoring and validation is done using anomaly based detection during access. For policy enforcement and preventing attacks, we have chosen VLAN Steering method. The reason for choosing VLAN steering is that it can be used with both out-of-band approach as well as in-band approach also. We need to implement both approaches to achieve access control dynamically. It helps to prevent insider as well as outsider attacks to a network. To prove the concept of blocking a malicious host after it is successfully admitted in a network, we present an example and a working algorithm for anomaly based detection. This algorithm uses IDS logged data from database for traffic monitoring and validation. It also updates signatures stored in signature database. An IPS sensor helps perform VLAN Steering in our system for quarantining suspicious hosts.

## 1. INTRODUCTION

We need to monitor Network Traffic properly during admission of a host in a network and when host access anything from the network. It is clear that the monitoring process must be dynamic to enable monitoring process to consider latest network status. For such continuous monitoring need, our traffic monitoring system should also be dynamic. TCP/IP supports multiple headers. This header information is normally used to keep track state of network connections. Therefore, in case of network connection, one needs to validate whole stream instead of packets. This kind of validation is used to search protocol non-compliance and intrusions using predefined criteria. For traffic monitoring and stream validation conventionally detection methods fall under the following two categories [2]:

Anomaly based Detection

Misuse/Signature based Detection.

Signature based detection depends on pre-defined signatures. These signatures

are reactive. It is because a signature cannot be developed until a threat is known and a remedy exists. Therefore, network can be attacked before signature is created. Anomaly-based detection establishes a baseline of normal network activity and responds to any traffic that appears unusual. It can detect attacks as soon as they take place. For example, if your computer never uses TELNET and suddenly some threat tries to open a TELNET connection from your computer, IDPS would detect this as anomalous activity. After detection, IDPS performs policy enforcement.

The paper describes our own implementation of anomaly detection process and a module to implement dynamic access control and policy enforcement. We use IDS logs to feed into the anomaly detection routine. The IDS log is stored in a MySQL database from where the anomaly detection routing reads the information. The information contains fields used in anomaly detection including signatures. The algorithm, upon receiving the information, tries to figure out if the input indicates some anomaly. If there is an anomaly, it does three things. First, it updates the signature database to inculcate new anomalies; second, it raises an alert to send an indication to IPS Sensor that an attack has occurred. Third, it blocks the attacking host for saving from further attacks.

We have divided our algorithm into three main functionalities: Stream formation using TCP Reassembly, Stream validation against stored signatures and Detection of anomalies. After our algorithm analyzes and validates network traffic, IPS sensor performs prevention using VLAN steering. VLANs segment networks into logical zones, and steering moves hosts onto particular VLANs. Steering happens by leveraging a switch's native VLAN management system through other protocols, like SNMP or CGI scripts. In our system, IPS sensor uses SNMP/CGI script to quarantine malicious host. Virtually any port can be in any VLAN, so mobility is easily accomplished.

We use VLAN Steering to steer a malicious host to quarantine VLAN. The host remains in the quarantine area and it is monitored for further attacks. After its behavior is adjudged as normal, once again it is shifted back to normal LAN. In this way, until the host stops sending malicious data, the host remains in quarantine VLAN and is monitored for any abnormal activity.

## 2. RELATED WORK

Anderson [Anderson 1980] defines an intrusion as any unauthorized attempt to access, manipulate, modify or destroy information or to render a system unreliable or unusable. Since then, lot of evasion techniques have been invented which are proving very effective against existing IDS. In this section, we describe two systems using anomaly-based approach: Snort [4] and Threat-aware anomaly based IDS.

Snort: After Snort detects an attack or anomaly, the preventive actions are drop, drop and log, etc. However, these preventive actions do not provide a solution to block a malicious host permanently. These attacks can recur repeatedly over a specified period. Since Snort only prevents attacks rather than the host who is the cause of this attack.

Threat-aware anomaly based IDS: This system introduces the concept of Threat-Awareness for anomaly based network IDS that periodically learns the changing threats in a network and enhance the capability of traditional anomaly based IDS to obtain network specific useful alarms [5]. It uses this knowledge to generate network specific alarms in real-time. However, this system does not block these threats permanently. There is no provision to prevent these attacks from recurring in future.

## 3. POLICY ENFORCEMENT FOR DYNAMIC ACCESS CONTROL

*Reason for selecting Anomaly Detection:*

Dynamic Access control focuses on Traffic Monitoring. Since Signature-based detection can only detect attacks against a fixed set of signatures, it cannot be used for dynamic access control. As opposed to this, anomaly based detection recognizes previously unknown attacks. Since it is impossible for the attacker to know, what activity generates an alarm, they cannot assume that any particular action will go undetected. Our algorithm utilizes this capability of anomaly detection to achieve dynamic access control. Dynamic access control deals with attacks inside the network after a host is successfully admitted.

Our algorithm achieves Dynamic Access Control using anomaly detection and VLAN Steering to solve the above-mentioned problem. Instead of Host Assessment using pre-defined policies, Dynamic Access Control focuses on traffic monitoring, stream validation, and policy enforcement. The major difference lies in the fact that host assessment and validation can be done only once during network admission, whereas traffic monitoring can be done even when host access network resources after successful admission. Our algorithm achieves three things, traffic monitoring, and validation of the incoming and outgoing stream and Enforcement of Policy for dynamic access control.

### ALGORITHM FOR TRAFFIC MONITORING USING ANOMALY BASED DETECTION

Traffic monitoring observes the host for bad behavior like port scanning or worm infection. It performs intrusion detection and monitors authentication requests and responses. It detects malicious behavior regardless of a host's condition. It also offers real-time detection of noncompliant activity.

Our algorithm, based on Anomaly Detection, focuses on detecting anomalous activities. It monitors the network for deviation of behavioral patterns from

normal behavior. For this, our algorithm does stream analysis and protocol decodes to ensure stream data adhere to the protocol. We have divided the entire procedure for dynamic access control in following four steps:

Step-1. TCP reassembly for stream formation.
Step-2. Stream data decoding and analysis using existing IDS logged data.
Step-3. Stream data validation by checking for deviation from normal behavior.
Step-4. VLAN Steering for policy enforcement and prevention

Our algorithm performs the first three steps. VLAN Steering is done by IPS sensor in the network. The following diagram shows the exact data flow:

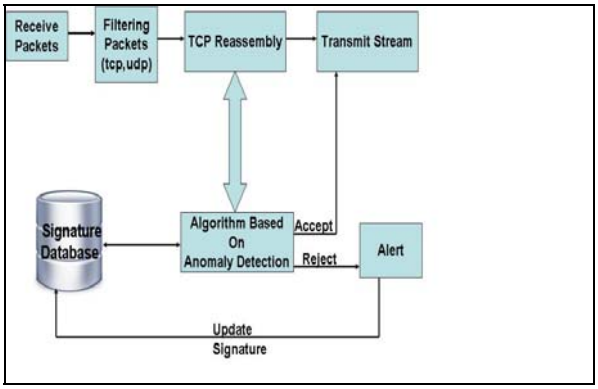


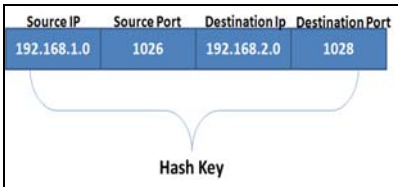

**Fig. 1: Data Flow across different Functions in the Algorithm.**

## A. STREAM FORMATION USING TCP REASSEMBLY

TCP reassembly uses different data structure to store the data it receives from the network. In this paper, we use hash table for maintaining TCP connection data. Hash is computed on the following quadruple:

Source IP Address
Source Port
Destination IP Address
Destination Port

Our hash table has two main fields: key and data. Combination of different hash data values generates a single unique key. For example, if we have a 16-bit hash, the hash table has 64K different entries. As each entry takes 4 bytes for data-part, the total size of hash table is 256KB. For each TCP connection, we store data in hash table.




| Field | Size (Byte) |
|---|---|
| Source IP | 4 |
| Destination IP | 4 |
| Source Port | 2 |
| Destination Port | 2 |
| TCP Sequence Number of next packet in order | 4 |

**Algorithm for anomaly detection using TCP Reassembly**

Step 1) Make Hash Key for Unique identification of packet
Step 2) If Packet arrives
2.1) if entry exists in the hash table
Drop Packet and goto Begin
Else
Create a new entry in the hash table
Hash add Item (Source IP address, Source port, Destination IP address, Destination Port)
Set Flag variable 0
End if
Else
If Entry exist in the hash table
Continue with TCP stream. If next packet is available
If yes
Continue
Else
Store the stream in particular data structure with other packet goto Begin
Else
Store that Hash key in another temporary database. Monitor that particular IP address and Check for Anomaly.
If anomaly is detected
Alert and Update Signature.
Else
Continue with next Stream data analysis
End If
End if
End if

Verify AnomalyDetect ():

1. Call the function Signature Detection

If Signature verify from the entry in data structure
Clear all the TCP entry in data structure
Drop the Stream data
End if

We have implemented the above algorithm using JAVA on Windows platform. For stream filtering and validation, we have used Jpcap package. Though it does not provide complete functionality as compared to C packet capture library, it does provide sufficient functionalities required to achieve our results. We store signatures and algorithm results in MYSQL database. For experimentation, we have configured two VLAN's: One VLAN retains hosts with normal behavior. The other is Quarantine VLAN, where administrator monitors quarantined hosts, until they regain normal activity.

First, algorithm receives and filters stream using TCP reassembly concept. Then the algorithm creates hash key for unique identification of the packet. The hash key is inserted in hash table and the hash key and data part will match with signature database by calling the function Signature detection. When next stream arrives, then first algorithm matches with the existing hash key entries in the hash table. If the entry already exists in the hash table, then the hash key is stored in temporary database. Now main purpose of anomaly detection function is monitoring that particular IP address and quarantine it using VLAN Steering. If algorithm detects an anomaly then it forwards an alert and the anomaly function updates that information in signature database. If it does not detect any anomaly, then it continues with next stream.

After stream analysis, algorithm performs the task of validation. It validates the traffic based on pre-defined policies, to see whether the stream adheres to the protocol rules or not. If it detects some deviation from normal behavior, it reports an attack.

### B. STREAM DECODING & VALIDATION USING POLICY BASED ANALYSIS

The function AnomalyDetect () of our Algorithm defines and uses policies for detecting any sort of deviation from normal behavior. It categorizes the traffic into various categories based on normal behavior of network and hosts, and maintains log of suspicious and harmful threats based on any deviation from this baseline defined. In our system, we have pre-defined this baseline by monitoring network behavior offline. We also modify this baseline for normal behavior online while monitoring network traffic. In this function, we apply the traffic analysis using signature database. According to database, we apply condition for false positive and break the connection. The profile database is updated after new threat is detect. On the negative side, because of the analytical nature of its

model, algorithm is bound to raise a number of false positives, and the value of the threshold actually determines a compromise between the number of false positives and the number of false negatives the IT security personnel is willing to accept. Fig.2. shows the output produced for various anomalies experimented in our network:

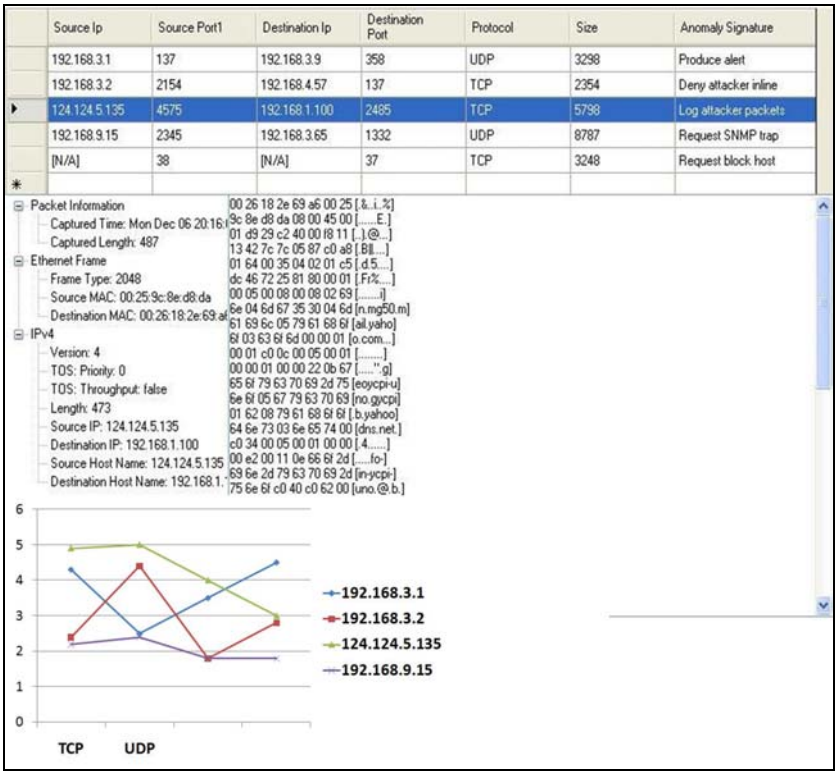



**Fig.2: Sample Anomalies detected by algorithm**

Example:

On port 80 only HTTP protocol should be allowed. Any other traffic is to be blocked since it shows deviation from normal behavior for that port. HTTP traffic has a POST state where a client is sending data to the web-server (can be binary data or options for the website). If none of the web server's pages are using such protocol option it is suspicious to see such anomalous traffic going to the web server.

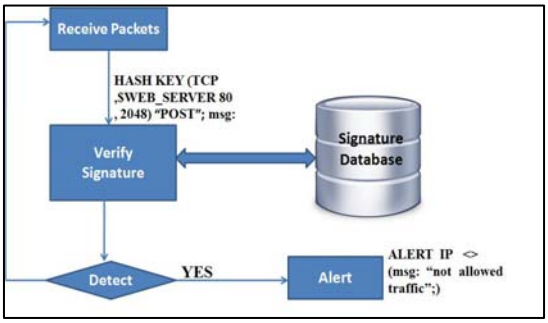



**Fig. 3**

### C. POLICY ENFORCEMENT

Users cannot be isolated from each other, except broad classes such as unauthenticated from authenticated, and even this is only enforced pre-connect. Once a user is placed in an authorized VLAN, the security device cannot monitor that user's activities, and so that user can mount attacks against whomever they wish in the authorized VLAN or on the rest of the LAN, for that matter.

Combining the algorithm with VLAN Steering, we can detect outsider as well as insider attacks dynamically and prevent it.

### D. DYNAMIC ACCESS CONTROL USING VLAN STEERING

All existing IDS including Snort, use static policy enforcement techniques such as 802.1x, DHCP, ARP management etc. These techniques use many enforcement policies to drop malicious traffic. These policies are DROP Packet, DROP Connections, BLOCK Sender IP, etc. All these enforcement techniques prevent attacks by blocking further traffic flowing from one zone to another zone (i.e. LAN to WAN or LAN to DMZ). However, there are chances that if these infected sources are in LAN then they are infecting other hosts too in the network. They might also be generating unwanted traffic in network and choke up the internal bandwidth. Behavior of such maliciously infected hosts in the LAN can disturb the overall service and utilization of resources.

#### Why VLAN Steering?

VLAN Steering deals with enforcement of policy to implement dynamic access control by which such hosts can be isolated from LAN and cannot affect the service and availability of resources. Therefore, we have chosen VLAN Steering in our system.

After our algorithm using anomaly detection deals with Traffic Monitoring & Stream Validation, IPS sensor in our system performs VLAN Steering for policy enforcement to implement dynamic access control by which such hosts can be isolated from LAN and cannot affect the service and availability of resources. When routine generates an alert, IPS Sensor assigns a quarantine or production VLAN to the access switch ports. Managed Access switches whose profiles are pre-defined in the IPS Sensor are called distribution switches. The profiles enable the Sensor to use protocols such as SNMP, Telnet, or SSH to manage the switches for tasks such as shutting down the port, changing VLANs and so on.

Fig. 4. shows how our IPS sensor enforces policy using heuristic approach. Using NAC at the time of network admission, network administrator assigns each host a VLAN ID. Port 2 is used for Port Mirroring. We have defined a quarantine area separately for malicious hosts. Static access control is performed using 802.1x. For Dynamic Access control, whenever the IPS Sensor detects malicious attack from any of the host on VLAN, it uses CLI / SNMP for shifting any suspected host in the quarantine area.

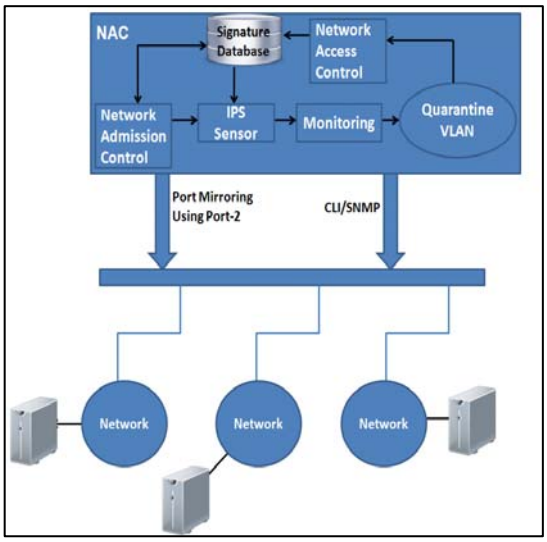



**Fig. 4. Process of VLAN Steering to quarantine a malicious host**

***Policy Enforcement during Admission Control using VLAN Steering***

When a host plugs into the switch port, the switch sends this event to the IPS Sensor, which assesses the host for one of the criterion defined and grants access accordingly. If an unhealthy host plugs into a port, IPS Sensor assigns quarantine VLAN to that port. Otherwise, it will assign production VLAN.

***Policy Enforcement during Access Control using VLAN Steering***

When the IPS Sensor detects attacks from a host on its configured monitoring port, it can put infected host in a separate VLAN i.e. Quarantine VLAN. Such quarantine VLAN does not have any access of the network other than walled-garden area. Thus, quarantine action prevents non-compliant hosts from harming other network systems, by isolating them from the network for a specified period. When this host turns healthy, the IPS Sensor assigns the production VLAN to the same port.

## 4. CONCLUSIONS & FUTURE EXTENSIONS

From the algorithm, example and results shown in this paper, we conclude that in order to achieve access control after a host is successfully admitted to a network, we need to combine anomaly detection with VLAN Steering. The capability of our algorithm using anomaly detection to detect anomalies enables

us to monitor traffic rather than host for malicious behavior. VLAN Steering blocks the host permanently to avoid further anomalies until host reverts to normal behavior.

Key to full-proof dynamic access prevention is proper traffic monitoring. Signature based traffic monitoring is not efficient. Therefore, one has to look into anomaly-based detection. However, the drawback of anomaly-based detection is that it generated large number of false positives. In addition, it is more complicated and hard to understand. Building and updating profiles also require extensive work. Existing anomaly based detection techniques must be extended to cope up with future attacks.

**REFERENCES:**

1. Dain, O. and Cunningham, R 2001.Fusing a heterogeneous alert stream into scenarios. In proceedings of the 2001 workshop on Data Mining for Security Applications. 1-13
2. Kumar, S and Spafford, E.H. 1994. A pattern matching model for misuse intrusion detection. In proceedings of the 17'Th National Computer Security Conference 11-21.
3. Peng Ning, Yun CUI, Douglas Reeves and Dingbang XU, 2004 ACM Transactions on Information and Security, Techniques and Tools for analyzing intrusion alerts.
4. www.snort.org
5. Subramanian Neelakantan & Shrisha Rao, A threat aware anomaly-based IDS for obtaining network specific useful alarms.