Synopsis

Ms. Shalvi Dave

Batch: 2011

Enrollment No: 119997493014

Guide : Dr. Bhushan Trivedi

Thesis Title: Network Access and Admission Restriction Using Traffic Analysis and Vulnerability Detection.

Statement of Purpose: To simplify alert aggregation by application profiling, reduce false positive alerts and generate set of precise advices for attack prevention in real time.

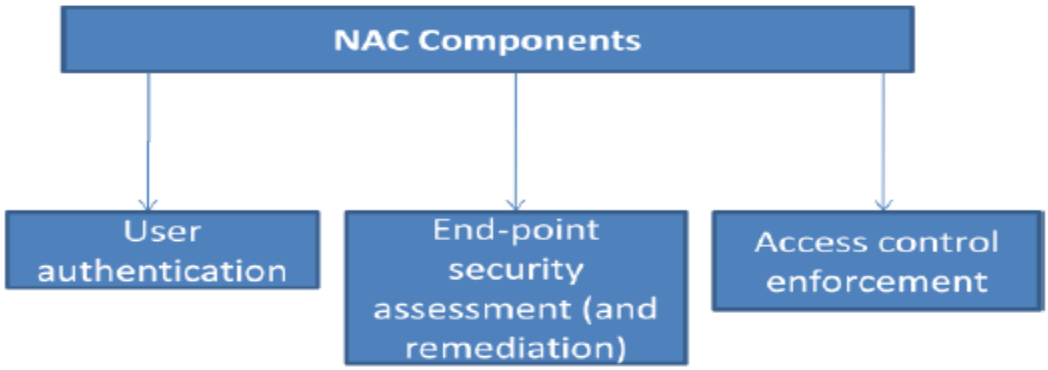
Abstract: Network is a mix of healthy, potentially vulnerable and already infected hosts. One would always want to keep the already infected as well as potentially vulnerable hosts separate from healthy hosts. The objective behind this is to stop transitive spread of vulnerability from one infected host to another and also prevent potential attack on other shared resources. Network admission solutions try to solve the problem of keeping such hosts in a separate functioning section from clean hosts. Network administrators normally prevent the vulnerable and infected hosts from accessing resources outside the network periphery, in lieu of the fact that they may launch an attack on outside resources which may backfire on the network performance itself.

Along with applying these rectifying methods, one would also want to prevent such incidents from happening by protecting potentially vulnerable but not infected applications either by restricting access. Most of the solutions implement this security by denying access of admission of such potentially vulnerable or infected hosts into network and by restricting overall network as well as Internet access for them. This implementation may effectively work if we considered every potentially vulnerable host as infected hosts. We categorize potentially vulnerable host and an infected one as: If a windows machine running with lower service update version is considered as potentially vulnerable but a machine having a virus is an infected host. Tight security restrictions in organization may lead to productivity loss if absence of security update prevents a user to access official mails.

Our research is a concentrated effort to maximize alert capturing, simplify alert aggregation without compromising with network performance. We propose a model that can be as secure as conventional solutions without compromising network productivity. The proposed model advocates the fact that not the machine but the application running on that host is either potentially vulnerable or infected so the access control policy focuses on the process and not at a gross level of host. The model is designed to keep the compromised applications separate from clean applications. It also builds up the run time data information for such applications running in the network to maintain the global black list of such

applications with highest confidence level. Our research also generates precise set of advices to network administrator for attack prevention in real time.

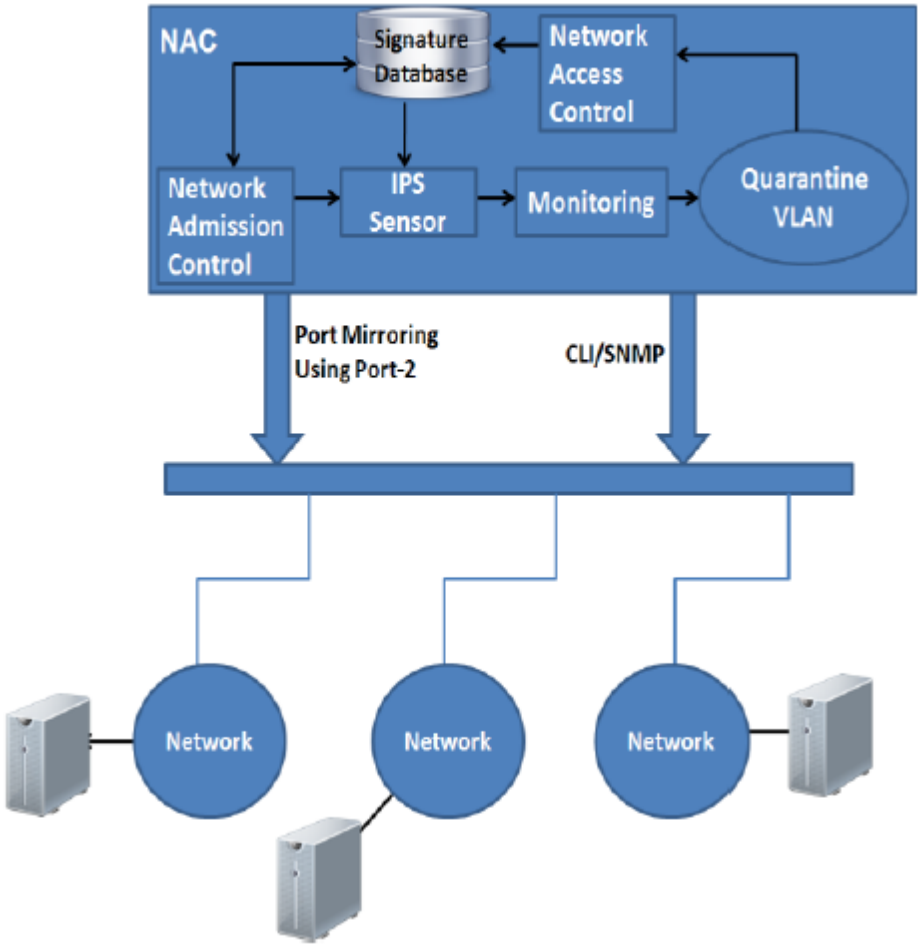
Brief description on the state of the art of the research topic: Network, wired or wireless, is an amalgamation of healthy hosts and shared resources, accessed by these host devices. At the same time, sharing resources in a network invites problems in the form of clogged bandwidth, spread of virus and data leakage [14][15]. This happens due to host malfunctioning, application malfunctioning, or corrupt kernel routines. Network Admission Control (NAC**)** is a combination of technologies and defined processes, whose aim is to control access to the network resources allowing only authorized and compliant host to access and operate on a network. The following figure showcases basic components of NAC:



NAC is a static policy implementation, which is achieved with the following functional specification:

- Periodic security assessment and remediation of each host which is part of network: Security assessment of every host involves periodic inspection of the state of the host. State of a host is defined as the processes / applications running on the host and the level of access each of them is allowed for shared network resources[4][5]. Using agent-based or agent-less systems for assessment, NAC provides capabilities to identify the security bearing of connected devices.
- Access Control Enforcement: Security assessment and remediation of host is followed by access control enforcement also known as policy enforcement. Policy enforcement is implemented as: Pre-connect mechanism and Post-connect mechanism. Pre-connect policies focus on host compliance and user authentication [6][7]. Post-connect policy enforcement employs techniques such as traffic monitoring, intrusion detection using signatures or behavioural anomaly indicators and activity monitoring.

Policy Enforcement of restriction for host is achieved normally using VLAN steering. VLANs or virtual local area networks, segments network 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 [1]. VLAN steering requires VLAN enabled switch with facility of executing command from remote systems. Different methodologies exist to achieve the VLAN steering using SNMP or Telnet or SSH [2]. It blocks the entry of unauthorized or unhealthy host in the network. The following figure demonstrates the process of VLAN steering:



Our research on NAC identified the following problem area [11][12][13] in the existing NAC solutions:

- Security assessment of host periodically checks for currently installed programs / software patches applicable, operating system and kernel routines. This assessment is very

time and resource consuming. These assessments also have a tendency to generate many false alarms.

- Access control enforcement usually quarantines the potentially vulnerable host / infected host after assessment and validation fails. This affects the overall productivity of network and also hampers network performance.

Various types of Intrusion Detection and prevention systems (IDPS) such as host-based (HIDPS), network-based (NIDPS) or perimeter-based IDPS have been designed and implemented to achieve and enhance NAC capabilities.

Host based Intrusion prevention systems (HIDPS) also aim to solve above-mentioned NAC problems in a different way. HIDPS engines running on host machines keep a watch on system level routines, applications, etc. and try to prevent unauthorized or unusual system access by either dropping the packets which belong to that connection or aborting that application itself. In comparison to NAC, it can give information in real time and have a control at all the levels on Host so it can control the access to infected application. However HIDPS performance is strongly dependent on previous learning as well as system information. Since operating systems have grown tremendously complex, extensive monitoring at host-level becomes all the more difficult. Apart from this, any attempt by an attacker, who has access to network resources, leading to application-level exploits, goes unnoticed by an HIDPS [8][9][10].

Network based Intrusion prevention system (NIDPS) aim to solve the network-level exploitation problem. It normally gets deployed on gateway and scans all the traffic going through it for any known protocol vulnerabilities. NIDPS works best in case of protecting internal services exposed to Internet from known application & protocol vulnerabilities. However, a major bottleneck for NIDPS is false alarm rate. It is unable to generate precise alert information as it lacks granular information of applications or services running on a particular host machine. This information is readily available in case of traditional NAC solutions or HIDPS.

As our initial research indicates [2][3], existing Network Access and Admission control systems tries to do Host assessment in a similar fashion as Host based Intrusion detection systems. Therefore, it relies on effectiveness of HIDPS assessment.

These findings helped us to lead a research and survey for the second point, Identification of clean traffic and giving access to network resources. Traffic Monitoring and vulnerability detection is a major functionality of an Intrusion detection and prevention systems (IDPS) for clean traffic identification and network access control. Traffic monitoring observes the host for bad behaviour like port scanning or worm infection. It performs intrusion detection and monitors authentication requests and responses. It detects malicious behaviour regardless of a host's condition. It also offers real-time detection of any dissenting activity.

We have surveyed the working of various popular Intrusion Detection and Prevention systems (IDPS) based on following questions:

- How IDPS control Network admission and Access?
- How Traffic is monitored at the network periphery and internally?
- What policy is implemented once a host is found vulnerable?

The above questions led to the following parameters for survey of IDPS:

- Different ways of deploying IDPS [23] [24]
- Architecture of IDPS [1][2]
- Source of Information available
- Relevance of attack

Deployment and Architecture of IDPS addresses security coverage of network and effectiveness of IDPS security where as Source of Information and Attack relevance addresses false positives issues. Deployment of IDPS is majorly divided in three categories. Host based, Network based and Perimeter based. Some of the study of host-based assessment has been carried out in our initial papers [3] [4] and we summarize the study as follows:

Host based systems advocates and relies more on an operating system and network based systems relies more on a network traffic aggregated at central location. Network based IDPS can give more accurate and real time information about attacks while Host based IDPS can give more information about operating system parameters like vulnerable application information on top of alert. The following major issues with individual approach are:

- Since operating systems are more complex, the complexity makes traffic monitoring difficult at host-level. It also influences and sometimes also hampers overall performance of the host.
- Network-level monitoring is difficult when data is encrypted. The problem is manifested in IPv6 whose main goal is authentication and confidentiality of data.

So a combined approach of hybrid IDPS is proposed by us who utilizes the best of both deployment approaches. It works on network traffic to give real time alerts but with more granular information about application such as version and name, that is provided by Host based IDPS.

Architecture of IDPS is also called topology of IDPS, which can be either centralized or distributed. We can categorize whether an IDPS is either centralized or distributed according to the location at which it monitors traffic, alert log after alerts are generated and alert log aggregation takes place [3].Distributed monitoring, logging and aggregation can reduce load on central system but it lacks correlation of attacks if any. Putting everything on central server is also not a feasible solution as central server might go out of resources in processing alerts. Therefore, we decided to create a central server, which will receive, aggregated and distributed correlated alerts and further correlation of events can be done centrally if required.

Source of information and attack relevance is more about accuracy of finding the attack and tracing it with actual incident or application. We studied different approaches for the same. This lead to the following conclusion: [4]

- 92.85% of false alerts are false positives (FP) and 7.15% are false negatives.
- Out of these FPs, 91% of FPs occur only because policy configuration and not due to any security issue. It is also observed that all such FPs majorly occur due to traffic similarities between protocols.

Once we have an attack alert log, the next step is to aggregate the log in such a way that helps the administrator to take correct preventive measures. To correctly correlate mass alerts, data mining techniques such as building classification models from identified attacks or using rules to associate same attack instances are generally used[22][25]. A detailed study of approaches [17][18][19][20][21] for alert aggregation led to the following conclusion:

- Wrongly adjusted time-windows lead to more no of false positives.
- Aggregation done on source and destination IP/Port ignore the fact that similar traffic between different protocols may again lead to more no of false positives.
- These approaches also fail to classify an attack as Inbound/Outbound, which is very important for an administrator to take decisions during prevention [28][29[30].

Therefore, we decided to link the alerts and signature rules with the application information and attack direction. It gives a basic generalization and we can get rid of false alarms. Our initial survey on Architecture and Deployment helped us in achieving application profiled rules and alerts as the architecture deployment proposal advocates host based IDPS which is working on network data. [2][3]

Definition of Problem: To generate highly optimized aggregated and correlated raw alerts, mask false positives with highest possible accuracy, using traffic monitoring & vulnerability detection, without compromising network productivity.

Objective of Work: The overall objectives of our research are summarized as:

- Use a hybrid deployment approach for combining advantages of host and network based IDPS
- Identify attacks in real-time
- To use alert aggregation for eliminating false positives.
- To use correlation to group multiple alert instances to form a single threat scenario with highest confidence level and accuracy.
- Quarantine applications instead of hosts for better network productivity and throughput.

Scope of work:

- Simulation & utilization of raw alert dataset with standard parameters and our own configured ones
- Bifurcation of self-determining alerts and related alerts from raw alert log.
- Highly optimized aggregation and correlation of raw alerts into meaningful alert context

- False positives masked with high accuracy and confidence level
- Quarantine applications to achieve optimal network productivity
- Generate preventive advice for network administrator for preventive decisions

Original contribution by the thesis:

Although multitude researchers have worked towards enhancing existing Intrusion detection and prevention systems (IDPS), major breakthroughs in terms of prevention of attacks on network resources is not yet accomplished. Critical issues hammering the network security experts are: Increasing rate of false positives and negatives, smart crafting of known attacks in unknown forms, and identifying and analyzing unknown attacks [26][27][28]. Attack detection and prevention is not the only remedy. It is equally important to analyze the strategy behind these attacks, relate similar attack instances and understand the motive behind these attacks. Almost all IDPS generate alerts after attack detection and do not concentrate on the analysis of those attack scenarios, which is the need of the hour. Correct aggregation and analysis results in proper preventive decisions and assists network administrators to react in response to attacks in real time.

All of the already mentioned solutions have their own pros and cons. So we propose a combined approach by bringing best of all together. Summarizing the research gaps, we aim to achieve the following enhancements with respect to existing IDPS approaches:

- To restrict the admission & access of Host in the network like NAC but only when it is required.
- Isolate and restrict the access of applications running on the host when found potentially vulnerable or infected.
- Make our proposed model more practically implementable by working at network traffic level but at the same time provide a solution to classic NIDPS problems like False alarms.

As our solution is based on IDPS technology, our solution also inherits the problems of IDPS technology. We propose a more efficient approach to reduce raw alert log size with maximum accuracy and high confidence level. To achieve this, we add semantic information to alert generated by an IDPS such as application name, version, severity of alert and timestamp. This information leads us to actual source of attack with application name and version with attack severity and timestamp. Our results shown at the end proves our claim. In

testing, we have simulated raw alert dataset and shown how this semantic information helps our aggregation and correlation methodology. It reduces and classifies the raw alert data to provide precise preventive advice to administrator for prevention of attacks at run-time. The result also indicates that we take a much smaller granularity by addressing processes on the hosts and not the hosts themselves. Thus our proposed solution achieves run-time attack prevention without compromising network performance. It is more in close proximity to IDPS technology but we are enhancing the same, which can be applicable to solve the problems of network admission and access restriction.

Hypothesis:

a) **Null Hypothesis:** (i) In case of attack in the network, log the attack information and quarantine the host, if within the network periphery, or block the host, if otherwise.

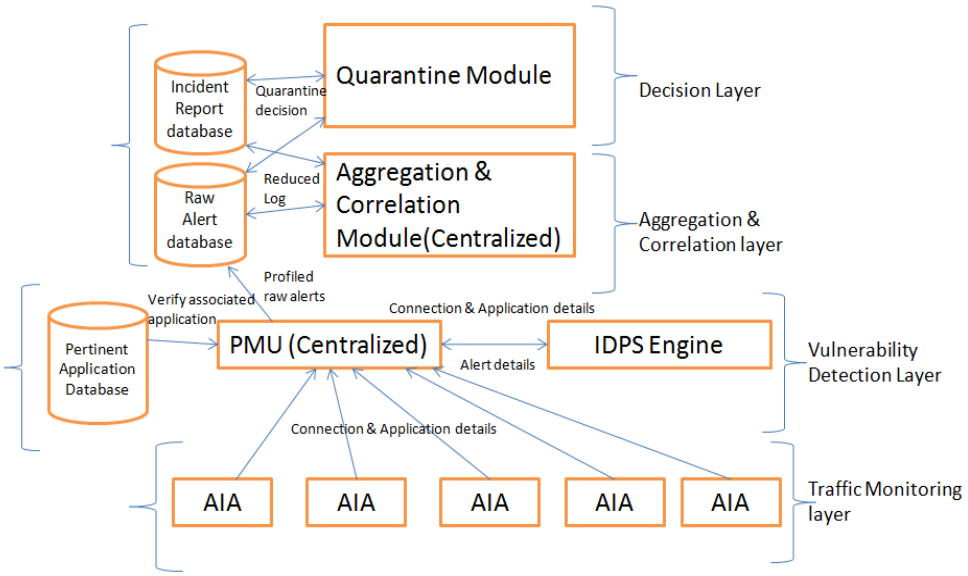
b) **Research Hypothesis:** (i) In case of attack in the network, log the attack information and quarantine the vulnerable application responsible for attack occurrence.

Methodology of Research, Results / Comparisons:

- Our Research is:
    - **Qualitative** since we continuously strive to maintain optimal balance between process of aggregation and network productivity, without compromising any of the performance measures.
    - **Experimental** since our proposed model follows hybrid approach for deployment, which we have used for proof of concept.
    - **Simulation based** because we have simulated our own utility to generate live data sets, parser for rule-set modification and different data stores for storing experimentation data.

Proposed Architecture:

We have designed and presented following subsystems, which constitute our proposed architecture:



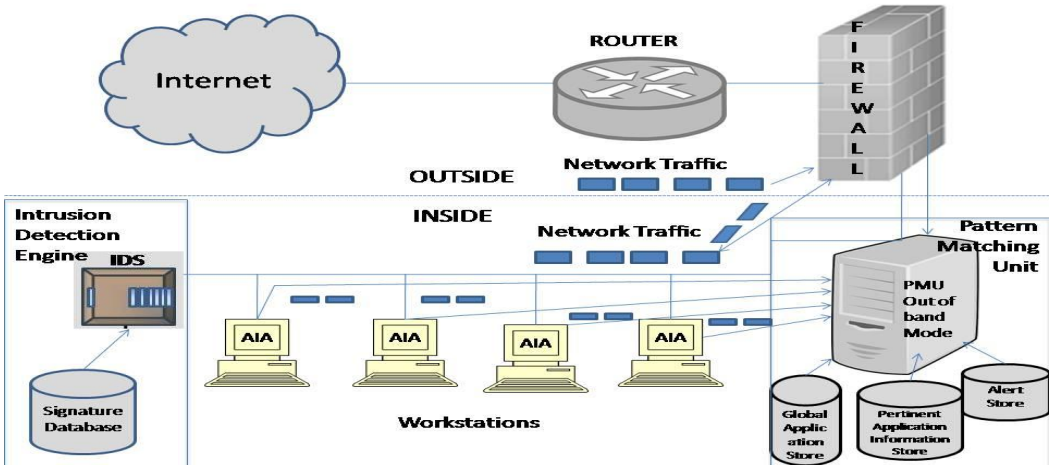
- **Traffic monitoring layer**: This layer monitors inflow and outflow of network traffic. For each connection initiated, Application Intercepting Agent (AIA) [1] reads initiator application details from by hooks in socket calls. It forwards these details with network packet payload to Packet Matching Unit (PMU).

- **Vulnerability detection layer**: Two components PMU and IDPS engine (any standard IDPS such as Snort) work in sync to detect vulnerabilities in packet sent by AIA. An alert sent by IDPS contains "sid" of the signature which matched the vulnerability. PMU performs lookup for this "sid" in Pertinent Application data store (data stores are described in next section). If a match is found, it stores profiled alerts (raw alerts with application details) in raw alert database.

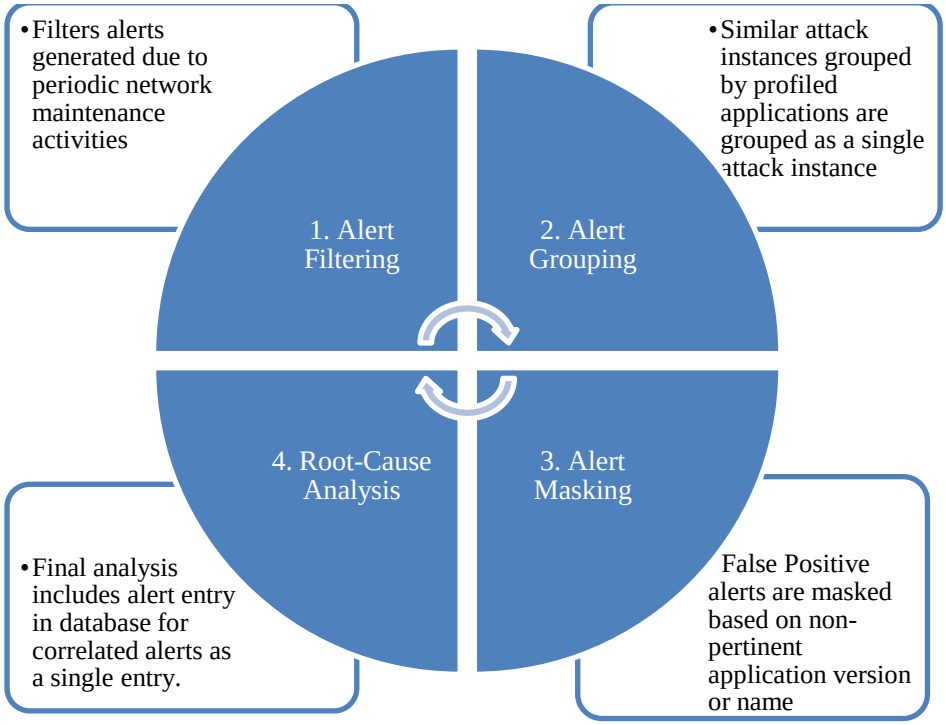
- **Aggregation & Correlation layer**: This layer is responsible for false positive reduction. Aggregation and correlation module at this layer reads raw alerts and masks false alerts, filters alerts for routine network maintenance activities, and group alerts which are part of single threat scenario with reason for grouping.

- **Decision layer**: Application Quarantine Module (AQM) at this layer reads the correlated log, and quarantines the applications. It also marks them as "quarantined" in the alert database.

Our application intercepting agents are installed on every host in the network and pattern matching unit is installed centrally on a server; thus our approach combines local and central processing into one solution. The following diagram shows our test setup:



Our alert aggregation module is described below:



System Flow Chart:

The following flow chart describes the complete system flow of our experimental model:

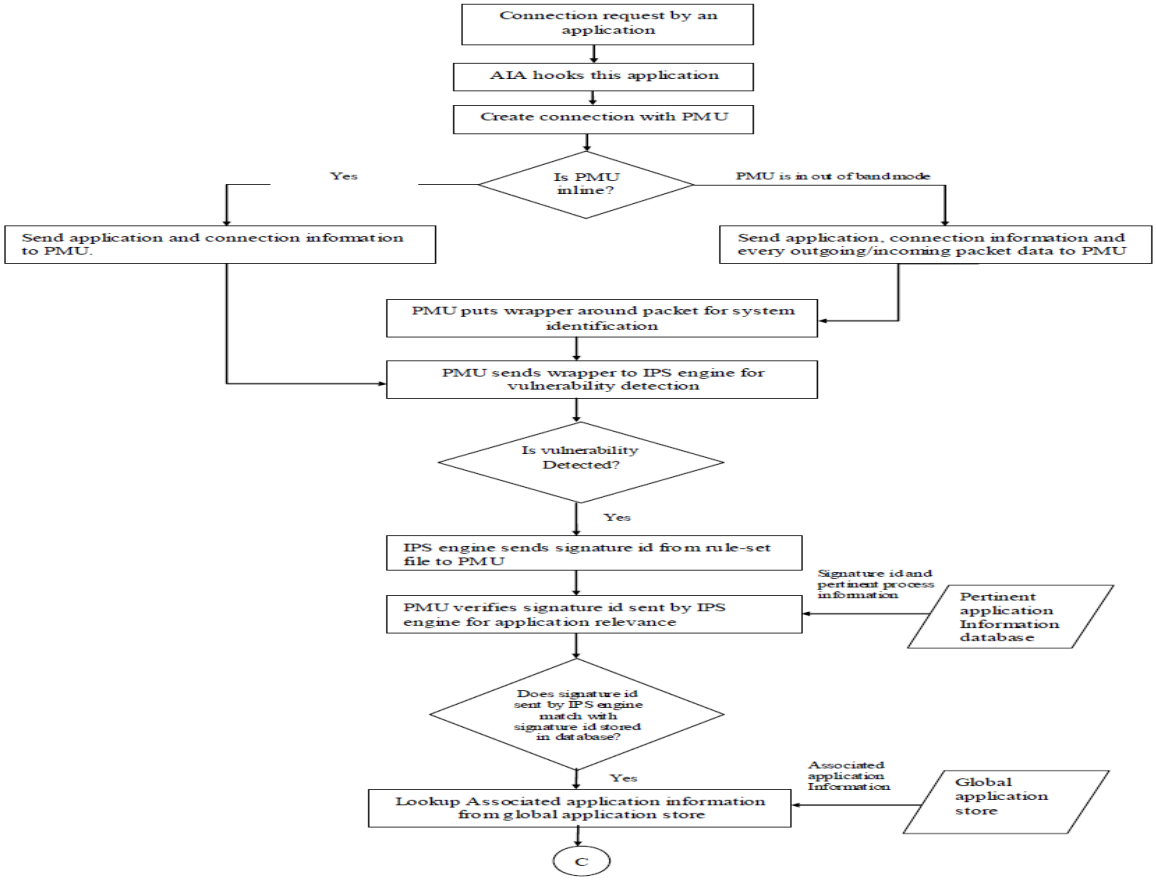
Acronyms: AIA (Application Intercepting Agent: deployed at each network node)

IDS (Any third party intrusion detection system)

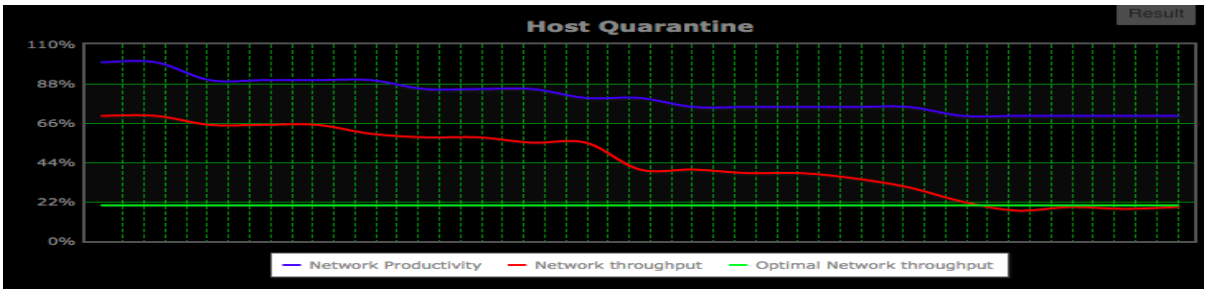
PMU (Pattern matching unit: deployed at central server)

AQM (Application Quarantine Module: deployed at central server)

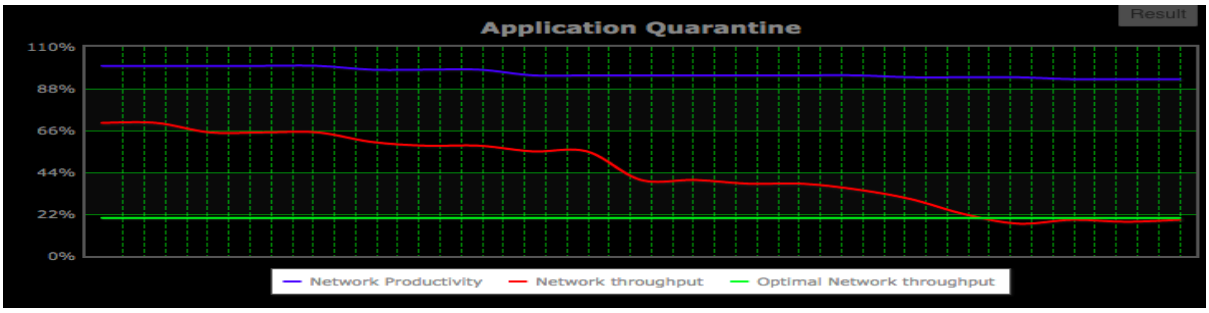
ACM (Aggregation and Correlation Module: deployed at central server)



After experimentation, our original dataset was reduced by almost 60%. We then configured same dataset for Snort and Fortinet and monitored their performance. We also continuously monitored network performance by the standard approach (for Snort and Fortinet-host quarantine) vs. our approach. For this, we developed a utility which would let us continuously monitor network throughput and productivity. The following is a snapshot of the Host quarantine approach by Snort:



The following diagram shows effect of our application quarantine approach:



Performance Comparison with other popular models: For comparing our proposed system with other popular IDPS available, we observed the outcomes of our approach. Then we observed the performance of Snort, on our dataset (since it is open-source, it was readily available and configurable) in our network. We also configured Forti-Analyzer (Fortinet IDPS), in our network. For comparison with other products CISCO and Juniper, we studied their logged documents (since they were costly for us to procure for live experiments). We now conclude our observations as follows:

Comparison Parameters	**Snort**	**CISCO**	**Juniper**	**Fortinet**	**Our approach**
Deployment (host-based, perimeter-based, network-based, hybrid)	Network-based	Hybrid	Network-based	Hybrid	Hybrid
Architecture (local, centralized, hybrid)	Centralized	Hybrid	Centralized	Hybrid	Hybrid

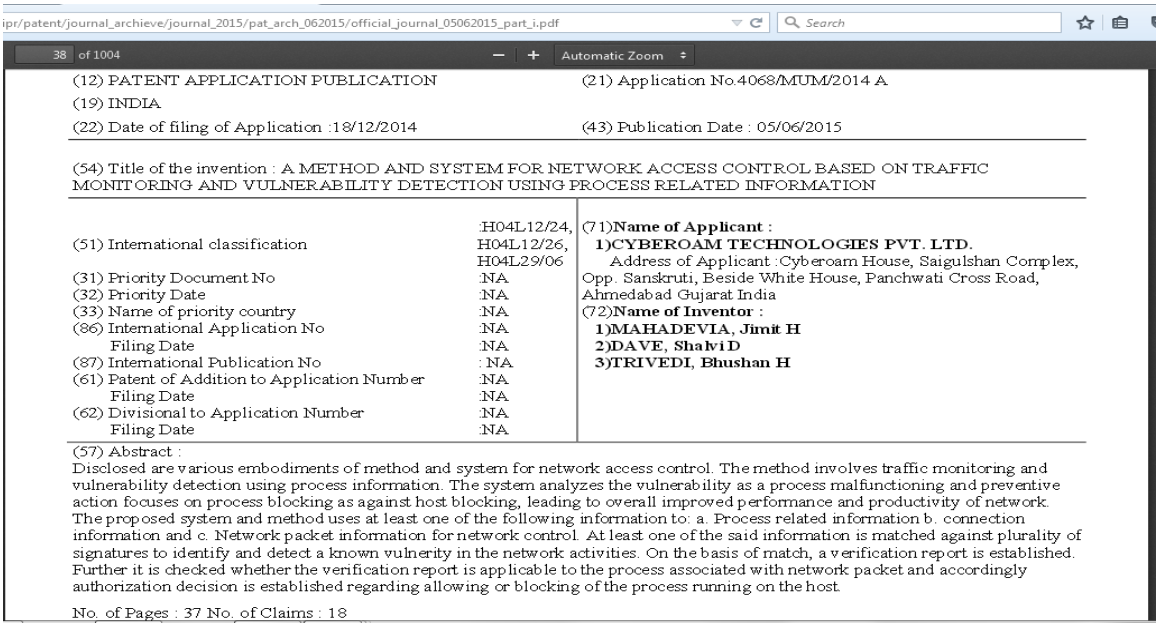
Comparison Parameters	Snort	CISCO	Juniper	Fortinet	Our approach
Real-time attack identification (true/false)	False	False	False	False	True
Attack identity (victim/attacker)	False	False	False	False	True
Attack boundary (from/to inside/outside the network)	False	True	False	False	True
Attack Relevance (is aggregation required?)	False	True	False	True	True
Aggregation approach	None	Does not work for applications using dynamic ports.	Does not correlate alerts	Aggregation is done on IP & Port. Cannot detect attacks which are distributed or correlated.	Aggregation is performed for alerts which are part of single distributed attack. Correlation is done to form a single threat scenario.
Quarantine approach	No database to store alerts generated. So need to be configured externally	Host blocking	None	Host blocking	Application is quarantined, host functions normally
Network throughput & Productivity	Considerable decrease in throughput because of centralized approach.	Because of host blocking, it can be affected whenever rate of alerts is high.	Does not affect much since no quarantine approach is applied	Network productivity down by 30% and throughput down by 40%.	Almost 80% achieved against optimal benchmark

Achievements with respect to objectives: We have achieved the following results through simulation of our proposed model with semantic raw alert dataset:

- False positives reduced with maximum accuracy as compared to other popular IDPS
- Precise correlation and aggregation of attack scenarios with confidence level specified
- Application quarantine approach leads to optimal network productivity and throughout
- Accurate preventive advice for real time attack prevention

Conclusion: We hereby conclude that our approach for aggregation and correlation with semantic raw alerts and application profiling leads to highly reduced and precise attack scenarios. The confidence level of our resultant alert database helps administrator take precise preventive measures. Our approach achieves all objectives without compromising network productivity and throughput.

Patents: We have filed complete specification for our research with Indian Patent Office and the patent has been published. The details for the patent are as follows:



ipr/patent/journal_archieve/journal_2015/pat_arch_062015/official_journal_05062015_part_i.pdf

38 of 1004 — Automatic Zoom

(12) PATENT APPLICATION PUBLICATION (21) Application No.4068/MUM/2014 A

(19) INDIA

(22) Date of filing of Application :18/12/2014 (43) Publication Date : 05/06/2015

(54) Title of the invention : A METHOD AND SYSTEM FOR NETWORK ACCESS CONTROL BASED ON TRAFFIC MONITORING AND VULNERABILITY DETECTION USING PROCESS RELATED INFORMATION

(51) International classification	:H04L12/24, H04L12/26, H04L29/06	(71)**Name of Applicant** : **1)CYBEROAM TECHNOLOGIES PVT. LTD.** Address of Applicant :Cyberoam House, Saigulshan Complex,
(31) Priority Document No	:NA	Opp. Sanskruti, Beside White House, Panchwati Cross Road,
(32) Priority Date	:NA	Ahmedabad Gujarat India
(33) Name of priority country	:NA	(72)**Name of Inventor** :
(86) International Application No	:NA	**1)MAHADEVIA, Jimit H**
Filing Date	:NA	**2)DAVE, Shalvi D**
(87) International Publication No	: NA	**3)TRIVEDI, Bhushan H**
(61) Patent of Addition to Application Number	:NA	
Filing Date	:NA	
(62) Divisional to Application Number	:NA	
Filing Date	:NA	

(57) Abstract :

Disclosed are various embodiments of method and system for network access control. The method involves traffic monitoring and vulnerability detection using process information. The system analyzes the vulnerability as a process malfunctioning and preventive action focuses on process blocking as against host blocking, leading to overall improved performance and productivity of network. The proposed system and method uses at least one of the following information to: a. Process related information b. connection information and c. Network packet information for network control. At least one of the said information is matched against plurality of signatures to identify and detect a known vulnerity in the network activities. On the basis of match, a verification report is established. Further it is checked whether the verification report is applicable to the process associated with network packet and accordingly authorization decision is established regarding allowing or blocking of the process running on the host.

No. of Pages : 37 No. of Claims : 18

Patent Link:

http://www.ipindia.nic.in/ipr/patent/journal_archieve/journal_2015/pat_arch_062015/official_journal_05062015_part_i.pdf , Page: 38

References:

[1] Shalvi Dave, Bhushan Trivedi, Jimit Mahadevia ,"Windows based Application Aware Network Interceptor", International Journal of Enterprise Computing and Business Systems, Vol 2, Issue1, 2012

[2] Shalvi Dave, Bhushan Trivedi, Jimit Mahadevia, "Parameterized Analysis of Intrusion Detection and Prevention Systems and their Implications on Attack Alert and Event Co-relation", International Journal Of Computer Applications, Harvard University, Vol 65, No.9, 2013

[3] Shalvi Dave, Bhushan Trivedi, Jimit Mahadevia , "Efficacy of attack detection capability of IDPS based on its deployment in wired and Wireless environment", International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.2, March 2013

[4] Peng Ning, Yun CUI, Douglas Reeves and Dingbang XU, 2004 ACM Transactions on Information and Security, Techniques and Tools for analyzing intrusion alerts

[5] F. Valeur, G. Vigna, C. Kru¨ gel, and R.A. Kemmerer, "A Comprehensive Approach to Intrusion Detection Alert Correlation," IEEE Trans. Dependable and Secure Computing, vol. 1, no. 3,pp. 146-169, July-Sept. 2004.

[6] F. Cuppens, "Managing Alerts in a Multi-Intrusion Detection Environment," Proc. 17th Ann. Computer Security Applications Conf. (ACSAC "01), pp. 22-31, 2001.

[7] K. Julisch, "Using Root Cause Analysis to Handle Intrusion Detection Alarms," PhD dissertation, Universitat Dortmund, 2003.

[8] Kai Hwang, Min Cai, Ying Chen, Min Qin, "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes," IEEE Transactions on Dependable and Secure Computing, vol. 4, no. 1, pp. 41-55, Jan.-March 2007, doi:10.1109/TDSC.2007.9

[9] Cheng-Yuan Ho; Yuan-Cheng Lai; I-Wei Chen; Fu-Yu Wang; Wei-Hsuan Tai; , "Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems," Communications Magazine, IEEE , vol.50, no.3, pp.146-154, March 2012

[10] Ramana Rao Kompella; Sumeet Singh; George Varghese;, "On Scalable Attack Detection in the Network," IEEE/ACM Transactions on Networking, vol.15, no.1, pp .14-25,Feb.2007 doi: 10.1109/TNET.2006.890115

[11] Herve Debar, Andreas Wespi, "Aggregation and correlation of intrusion detection alerts", ACM dl, RAID '00 Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection, pg 85-103, springer-verlag,london

[12] Alsubhi, K.; Al-Shaer, E.; Boutaba, R., "Alert prioritization in Intrusion Detection Systems," Network Operations and Management Symposium, 2008. NOMS 2008. IEEE , vol., no., pp.33,40, 7-11 April 2008 doi: 10.1109/NOMS.2008.4575114

[13] N. Anitha, S.Anitha, B.Anitha, A Heuristic approach for alert aggregation in intrusion detection system, Journal of Computer Applications, Vol-5, Issue 3,2012

[14] Autrel, F. and Cuppens. (2005). Using an intrusion detection alert similarity operator to aggregate and fuse alerts, 4th Conference on Security and Network Architecture Bat sur Mer, France.

[15] DARPA dataset found at http://www.ll.mit.edu/ideval/data/1999data.html

[16] KDD cup dataset found at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[17] Alexander Hofmann, Bernhard Sick, "Online Intrusion Alert Aggregation with Generative Data Stream Modeling," IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 2, pp. 282-294, March-April 2011, doi:10.1109/TDSC.2009.36

[18] F. Valeur, G. Vigna, C. Kru¨ gel, and R.A. Kemmerer, "A Comprehensive Approach to Intrusion Detection Alert Correlation," IEEE Trans. Dependable and Secure Computing, vol. 1, no. 3,pp. 146-169, July-Sept. 2004.

[19] F. Cuppens, "Managing Alerts in a Multi-Intrusion Detection Environment," Proc. 17th Ann. Computer Security Applications Conf. (ACSAC "01), pp. 22-31, 2001.

[20] K. Julisch, "Using Root Cause Analysis to Handle Intrusion Detection Alarms," PhD dissertation, Universitat Dortmund, 2003.

[21] Kai Hwang, Min Cai, Ying Chen, Min Qin, "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes," IEEE Transactions on Dependable and Secure Computing, vol. 4, no. 1, pp. 41-55, Jan.-March 2007, doi:10.1109/TDSC.2007.9

[22] Cheng-Yuan Ho; Yuan-Cheng Lai; I-Wei Chen; Fu-Yu Wang; Wei-Hsuan Tai; , "Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems," Communications Magazine, IEEE , vol.50, no.3, pp.146-154, March 2012

[23] Shalvi Dave, Bhushan Trivedi, Dashang Trivedi, "Simulation Of Security Agent Using Anomaly Based Detection and VLAN Steering", 2011 3rd International Conference on Computer Modeling and Simulation (ICCMS 2011)

[24] Shalvi dave, Jimit Mahadevia, Bhushan Trivedi, "Application Aware Event Logger", International Journal of Computing, Vol-1,Issue-2,April 2011,pg-201-208

[25] Ramana Rao Kompella; Sumeet Singh; George Varghese;, "On Scalable Attack Detection in the Network," IEEE/ACM Transactions on Networking, vol.15, no.1, pp .14-25,Feb.2007 doi: 10.1109/TNET.2006.890115

[26] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” Proc. Third SIAM Conf. Data Mining, 2003, http://www.users.cs.umn.edu/~kumar/papers.

[27] W. Lee, S.J. Stolfo, and K. Mok, “Adaptive Intrusion Detection: A Data Mining Approach,” Artificial Intelligence Rev., vol. 14, no. 6,pp. 533-567, Kluwer Academic Publishers, Dec. 2000.

[28] W. Lee and S. Stolfo, “A Framework for Constructing Features and Models for Intrusion Detection Systems,” ACM Trans. Information and System Security (TISSec), 2000.

[29] M. Qin and K. Hwang, “Frequent Episode Rules for Internet Traffic Analysis and Anomaly Detection,” Proc. IEEE Network Computing and Applications (NAC ‟04), Sept. 2004.

[30] L. Ertoz, E. Eilertson, A. Lazarevic, P. Tan, J. Srivastava, V. Kumar, and P. Dokas, “The MINDS—Minnesota Intrusion Detection System,” Next Generation Data Mining, MIT Press, 2004 .