Survey On Integrated Threat Management (ITM)

Ranjit Shrirang Nimbalkar
Department of Computer Science, VeermataJijabai Technological Institute, India

Dr. B. B. Meshram
HOD Department of Computer Science,VeermataJijabai Technological Institute, Mumbai -400019,

Abstract

Since emerged in 2004, Unified Threat Management (ITM) has been used widely to enhance network security protection. Typical ITM device integrates multiple security technologies, therefore its control and management involves various interfaces, message formats, communication protocols, and security policies and so on. Therefore, it is a big challenge to design and implement the configuration and management of security technologies in ITM. To address this issue, this paper proposes a practical ITM control mechanism that features ease-to-use, scalability, interoperability, high efficiency and reliability.

1. Introduction

With rapid development of the Internet, network security has become an intractable issue, and the security threats present the trend of diversity with quick variation velocity. As a primary network gateway defense solution, Unified Threat Management (ITM) emerged in 2004, and has been used widely. ITM is a special equipment consisting of hardware, software and network protection technologies in one single appliance, such as firewall, intrusion prevention system (IPS), gateway antivirus
(AV), VPN, content filter and so on .

ITM can be divided into two types: tightly-coupled ITM and loosely-coupled ITM. For tightly-coupled ITM, all the security functions are developed by one vendor, which is easy-to-use and has advanced Technologies. However, loosely-ITM integrates existing security products from various vendors. The interoperability and interaction between different security technologies is a matter of especial importance to configuration and management. Anyhow, the real ITM constructs a standard management platform to provide strong defense capabilities. To achieve the goal, we propose the solution to control mechanism for ITM with ease-to-use, scalability, interoperability, high-efficiency and reliability.

2. Related work

With network functionality growing increasingly complex, as discussed in [1-2], network configuration management becomes a challenge and many management methods or platforms have emerged. Some related management solutions are introduced in this section.

1) SNMP. Simple Network Management Protocol (SNMP) was created to solve the problem of the management of the distribution network based on TCP/IP protocols. SNMP is a UDP-based protocol. Although many successes may have been made in using SNMP in different domains, as shown in [3-4],it is not a perfect choice for ITM which is a integration management solution. In other words, SNMP couldn't reach the goal of interaction and interoperability.

2) TOPSEC &OPSEC. TOPSEC's Talent Open Platform for Security (TOPSEC) is a unified and scalable security platform, which integrates various network security technologies and excellent network security productions, realizing the interoperability and interaction between security products [5]. Check Point's Open Platform for Security (OPSEC) integrates and manages various network securities through an open, extensible framework [6] . Both of the technologies can configure and manage various network securities, but nearly all the security applications can't interact with either of TOPSEC and OPSEC unless they strictly follow the protocols created by the two technologies and even have been authenticated by them. For each vendor designs and develops their application programming interface (APIs) under certain environment, the two technologies cause the problem of compatibility. An open, standard and scalable mechanism is required to configure and manage various security applications.

IJERT

3) NETCONF. The NETCONF protocol defines a simple mechanism through which a network device can be managed, configuration information can be retrieved, and new configuration data can be uploaded and manipulated . The NETCONF protocol plays a great role in a system of automated configuration. However, the control mechanism of ITM is not just referring to configuration. For example, interoperability and interaction between different security devices are out of reach of NETCONF [7].

3. Solutions for the ITM

3.1 ATCA-based ITM

3.1.1 ATCA-based ITM overview

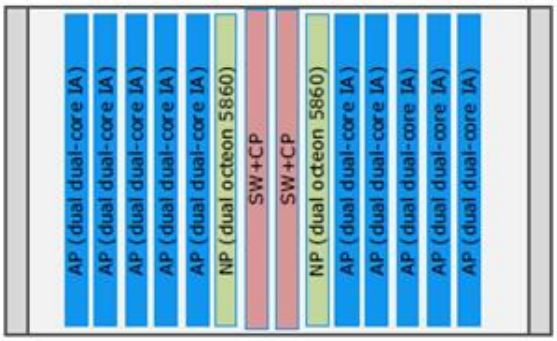

Figure 1. Infrastructure of ATCA-based ITM

As shown in Figure 1, ITM adopts the Advanced Telecommunication Computing Architecture (ATCA) and has 14 slots where various blades can be inserted according to applications and user's requirements. The configurable blades include:

1) Switch blade (SW) has the capabilities of fast switching and high capacity. There are two switching mode, 1GE Base switching for control plane and 10GE Fabric switching for data plane.
2) Network Processing blade (NP) deals with network layer and implements load balance. It plays the role of a firewall at the same time. NP adopts Multi-MIPS core architecture.
3) Application Processing blade (AP), based on multi-IA architecture, is used for deep inspection of network traffic with the functions of IPS, AV, etc. Single blade has the processing capacity of 500Mbps~1Gbps, and multiple blades can coordinating work.
4) Control Processing blade (CP) controls communication and interaction between various blades. And it also stores both configuration information and applications, and makes sure the hot standby of critical components.

3.1.2 Goals of Control mechanism

Requirements that a control mechanism of ITM must supply are listed below.
1*) Easy to use*. The development trend of network security is active-safety and auto-defense which can reduce the error and bad response by man-made factors. Since ITM integrates various security technologies (such as firewall, VPN gateway, Antivirus, IPS, etc.), users don't need to buy these security technologies separately, and network managers never are required to learn the configuration and management of all the equipments from different vendors. Therefore, one of the vital requirements of configuration and management of ITM is easy-to-use and comprehensive.
2) *Scalability*. To enforce protective capability, more and more security technologies are being integrated into ITM. ITM should be configured and managed well even after integrating new security technology. The configuration and management of ITM shall be scalable to meet the requirement.
3) *Interoperability*. As applications of networks are becoming more and more complex, the security has been increasing unexpectedly, and the mode and targets of attack vary quickly. An enterprise maybe has a series of security products (Anti-virus, Firewall, IPS, etc.). However, these security products generate a great deal of safety information with different forms, so it's almost impossible to make them cooperate and interact well. ITM, which can provide three-dimensional protection, must have the management ability of interoperability and interaction.
4) *High-efficiency and reliability*. ITM continues to evolve to offer multiple security services, integrates equipment from multiple vendors, and conducts continuous performance and feature tuning. Configuration errors have a significant portion of operator errors, and are the largest contributor to failures and repair time. About 60% of network downtime is owing to human configuration errors, and more than 80% of IT budgets are allocated towards maintaining the status quo [3]. This is why the configuration and management of ITM must have features of high-efficiency and reliability.

3.1.3 Architecture of Control Mechanism

For achieving the design goals, layer architecture to category the different functionality of ITM and abstract different layer's main function. The architecture can be used as a guideline for our system design and implementation.

IJERT

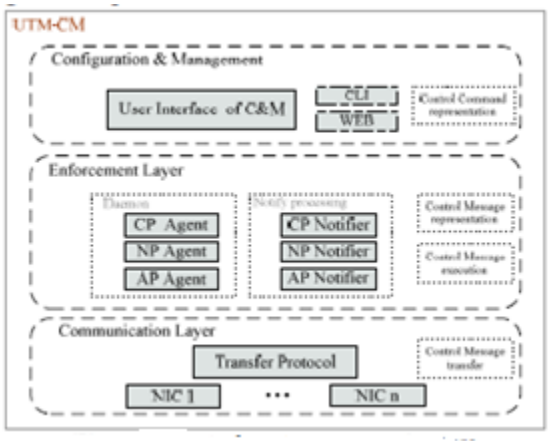

Figure 2. control mechanism architecture

Following is an overview of control mechanism architecture. It is described on a high-level overview in this section. following sections will discuss the details and implementation of several components . architecture divide it into three layers:

1) Configuration & management. It is human-to - machine layer and deals with control command Representation .
2) Enforcement layer. The layer processes the control messages, such as control messages representation and their execution.
3) Communication layer. Communication layer, machine-to-machine layer, controls the communication and interaction between different components (CP, NP and AP). We name the control mechanism as ITM Configuration and Management (ITM-CM). It includes two components:
1) ITM Configuration (ITM-C) configures the components of ITM;
2) ITM Management (ITM-M) manages the ITM.

3.2 Dual-core ITM

3.2.1 Typical technologies in the inspection

3.2.1.1 Traditional inspection based on TLU

As to the server offering specific service the directory server in the centralized P2P network, the super node in the mixed P2P network and some hostile opposite customer, it is possible to inspect the application of the IP package through identifying the IP address [8]. Most common protocol types have fixed port numbers and a lot of network applications achieve the definite functions with default port numbers. TLU (table lookup unit) is used by network protocol stacks to store and retrieve data in tables. To retrieve data, the TLU searches tables using a search key which is normally derived from one or more fields of a packet being processed, and returns the data associated with that key. Classification is used to determine such things as whether to forward or discard an incoming packet, what class of service to provide, or how to charge for it. As is shown in the Figure 3, the 5-tuple (including source IP, destination IP, source port, destination port and the protocol number) is put into the key to search for the result and then traditional inspection is achieved.

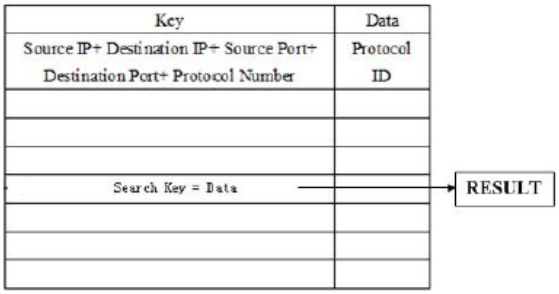

Figure 3.Table search operation

3.2.1.2 DFI (Deep flow inspection)

As to the two- direction communication data belonging to thesame session flow, if certain direction of the data is identified as an application, then its reverse direction will also be attributed to the same application [9].

3.2.1.3 DPI (Deep packet inspection) based on PME

DPI technology is a strict inspection method, which not only analyzes the head of the data package, but also involves the application payload. The fixed data field is analyzed and picked up in the application layer, then defined as character information and saved in the character database. Compare the traffic package with the character information and the matched package will be identified to certain application type. DPI technology could attain high precision of inspection through optimizing the character database [10]. The built-in Pattern Match Engine (PME), with its features and operational characteristics, is particularly conducive to providing high performance and accuracy simultaneously in UTM deep packet inspection:

- Regex allows the creation of sophisticated signatures that are fingerprints of various forms of undesirable content
- Stateful rule enables even more accurate signatures by tracking application protocol exchange
- Set and subsets enable only relevant portion of the total signature be used in a context-sensitive manner

IJERT

- Matching patterns across packet boundaries to increase accuracy by matching application messages that span packet boundaries

3.2.1.4 QoS (Quality of service)

QoS is the idea that transmission rates, error rates, and other characteristics can be measured, improved, and, to some extent, guaranteed in advance. QoS is of particular concern for the continuous transmission of high-bandwidth video and multimedia information. Transmitting this kind of content dependably is difficult in public networks using ordinary "best effort" protocols. Using the Internet's Resource Reservation Protocol (RSVP), packets passing through a gateway host can be expedited based on policy and reservation criteria arranged in advance.

3.2.1.5 Integrated protocol processing

Protocol processing in UTMs involves repeated application of several basic operations, mainly including packet classification for firewalling and protocol analysis for intrusion prevention [11]. In order to get the maximum gain in overall performance, we need to design the protocol processing applications in a very efficient manner. However, because each processing engine need to manipulate multiple fields of packet header, they have to load the packet header from the off-chip memory to the local cache and then carry out modification and classification, and finally write the new header back into the external memory. Such read-write operations are very time-consuming, and hence greatly impede the overall processing speed .

3.2.2 Implementation of UTM system model

3.2.2.1 Dual-core usage

The dual-core NP can be used in the Symmetric Multi-Processing (SMP) or the Asymmetric Multi-Processing (AMP) mode. From Figure 4, we can easily see how UTM operations can be implemented in the AMP mode. In essence,

- **Core2**, working in conjunction with the Pattern Matcher, is used for the CPU-intensive application protocol and content processing, including matching suitable parts of the packet flow payload against intrusion, spam and virus signatures.
- **Core1**, working in conjunction with eTSECs and TLUs, is used for the packet data path, such as packet I/O, forwarding, controlling QoS and updating statistics.

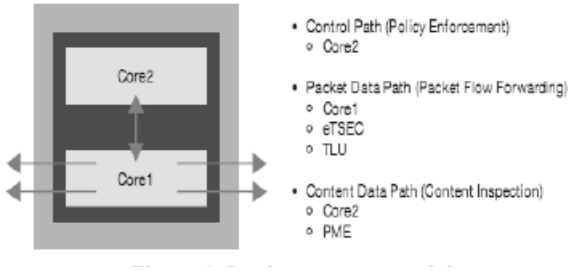

Figure 4.dual core ITM

3.2.2.2 Implementation

Form Figure 5, we can clearly see the total process of the implementation of the UTM system model. Firstly, the Ethernet controller puts received packet into appropriate queue in memory and interrupts Core1. Then, Core1 analyzes the network packet and extracts 5-tuple key from the packet header. After that, Core1 writes key to TLU to lookup flow table and reads back results (Protocol_ID) of the lookup.
Then the content of received packets are put in to the PME. After the deep packet inspection, the system reads back the Patten_ID. Focus on the Protocol_ID and Patten_ID, the final result is established. At the same time, Core1 instructs appropriate Ethernet ontroller to transmit packet and the Ethernet controller transmits packet, applying QoS rules as required. There is a PC server on which SQL server 2003 is installed in the system model. The configuration information is saved on this server, including TLU, PME and QoS rules. We can upgrade the configuration of TLU (including Source IP, Destination IP, Source Port, Destination Port and Protocol Number) and PME (including Regex, Stateful Rules, Sets and Subsets) on this server; what's more, QoS rules should be configured and upgraded on this server

IJERT

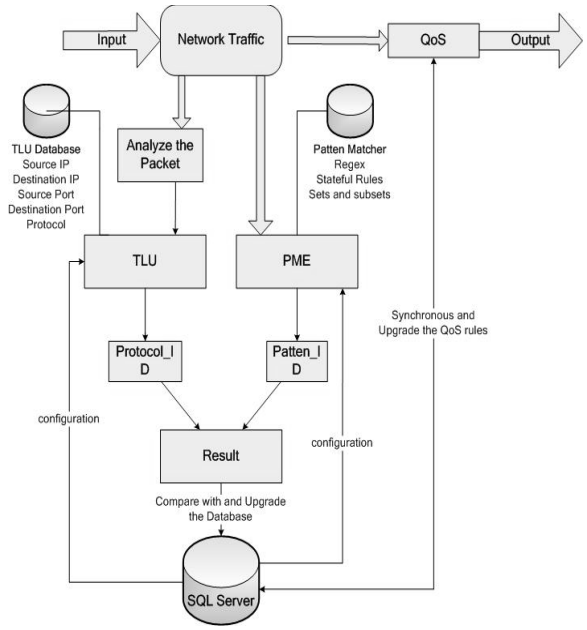

Figure 5.system model implementation

3.3 NetChannel ITM

Under the MultiCore/MultiProcessor architecture, according to Amdahl Law, the improved program of the system quality acquired by a certain part from the system in a certain executive way of higher speed depends on the use frequency of this way or the ratio of the total executive time.with the increase of Core/ Processor number, the quality increases in a much-more- like linear way while the cost system increases in a squared way. Such cost as the share or mutual exclusion of lots of resources and the resource management will lead to the improving space[4] loss of the system quality with the increase of the processors. Under the traditional X86 architecture given in the literature[1], the resource use during running NetPerf by MonoProcessor and MultiProcessor. Based on the above analysis, there is a flaw with MultiCore/MultiProcessor architecture: the increase of the Core/ Processor number does not necessarily bring about the improvement of the system quality in a linear way. The reason is that with the increase of the processing units extra operation like lots of breaks, lock mechanism and context exchange will consume much resource. What is more, copy operation for many times particularly in the case of a mass of small byte data packet, the system cost will be much higher. Therefore, NetChannel idea can help to solve the bottle neck problem of the UTM architecture quality based on MultiCore/MultiProcessor. Integrating the seven functions of UTM and the characteristics of Netchannel, the whole system can be divided into two major parts on the data level:

1) the data processing on the traditional ether layer and network layer: the processing of VPN/address conversion/cause can be completed in the inner core space because this processing demands instant response and transparent information of the users.

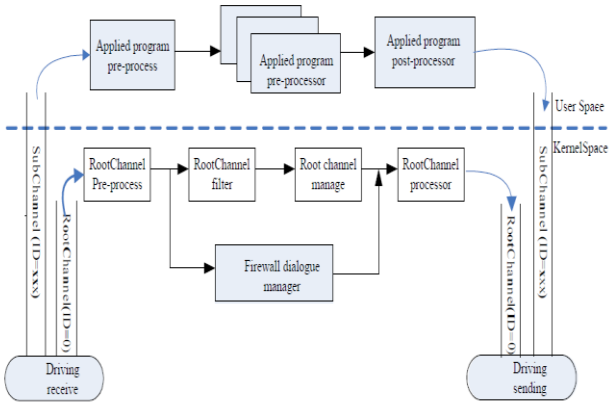

Figure 6.NetChannel ITM

2) the data on the applied layer: such operations as dealing with the virus check, attack check, anti-waste check can be completed in the user space because these operations consume much more resources and are relatively be dangerous to these data; at the same time, because different users needs should be taken into consideration, open applied layer interface is provided and mode matching task can be completed by use of part hard ware. Based on the kernel mode, the classification of the data packet, visit control, VPN, address conversion, establishment/destroying SubChannel and firewall dialogue table, etc can be finished. Father course KernelRootThreadof the permanent system has two registered ID of 0 RootChannel in each network card driving program: Tx-RootChannel and Rcv-RootChannel. The network card first receives data packet and look for whether there is registered SubChannel according to five basic units of the data packet. If registered SubChannel is found, the data packet will be pushed to the certain SubChannel and be processed in the applied layer; if it is not found, the data packet will be pushed to RootChannel to wait for being processed. This system architecture is shown in Figure 6, in which each mode functions are as follows:

1) RootChannel pre-processor: the data can be obtained from Channel whose ID is 0, then these data will be processed in the following manner: if this data packet is a encoded one, it will be decoded; if it is a sliced one, it will be re-organized according to the different parts,

IJERT

then look for the corresponding dialogue item in the firewall dialogue table according to the five basic units information of the data packet(original IP address, target address, protocol, original port, target port), if the corresponding dialogue item is found, the data packet will be sent to the firewall dialogue manager for processing. If not found, the data packet will enter RootChannel filter.

2) RootChannel filter: the data packet will be validated regularly according to the regular collocation of users, and illegal data packet will be discarded, while legal data packet will undergo such operations as address conversion, path searching and the firewall dialogue table will be set up.

3) RootChannel manager: according to the output of RootChannel filter, SubChannel from original interface to the target interface will be set up and the information of the corresponding firewall dialogue will be added to SubChannel. If the data packet needs to be processed on the applied layer, rcv-SubChannel and tx-SubChannel from drive to applied layer will be set up. These two Channels share one ChannelID, SubChannelID is the information of the data packet on transport layer such as 80/TCP.

4) NetChannel processor: according to firewall dialogue information, users' collocation information and NetChannel information, such operation as the data packet modification and check sum will be finished. If it is VPN data packet, it will be encrypted first; if the data packet needs to be sliced, it will be processed partly, then the processed data packet will be pushed to tx-RootChannel; the data packet followed from SubChannel to the applied layer carried with it firewall dialogue information. This data packet may be encrypted, sliced, needed to be regulated by sequence. So, after applied layer polls data packet from SubChannel, the integrated data packet will need to be checked and decoded. The flow of each mode on the applied layer is shown in the following descriptions.

5) Applied program pre-processor: the data packet will be sent from drive to applied layer directly. Then the data packet will be checked by applied program pre-processor according to the firewall information carried by data packet and SubChannel. If they are not matched, the data packet will be discarded. Afterwards if the data packet is sliced, it will be reorganized. If the data packet is in disorder, it will be reordered first, and then enter the applied program processing mode.

6) Applied program processor: operations like URL stopping, virus defending, waste mail resisting, and Fishing resisting will be completed according to users equipment (including in the firewall dialogue information). If the data packet is found illegal, it will be discarded and EMS memory will be released. The accelerating card of the outer applied .

4. System-level Optimization for High performance integrated Threat Management

4.1 Protocol Processing Algorithm

Although the theoretical bounds make it impossible to design a single algorithm that performs well for all cases, fortunately, real-life rulesets have some inherent characteristics that can be exploited to reduce the complexity both in search time and memory space. A variety of characteristics of real-life rulesets have been presented and exploited and some best known algorithms like HiCuts [13], HyperCuts [14], RFC [15], and HSM [16] achieve encouraging improvements in packet classification performance Generally, these algorithms can be classified into two categories [12]:

Decision tree search algorithms: HiCuts and HyperCuts belong to this category using decision trees for packet search. Generally, decision tree algorithms require less memory. However, the complexity of the decision trees often leads to implicit worst-case bounds and thus cannot ensure a stable worst-case classification speed.

Parallel search algorithms: RFC and HSM perform independent parallel searches on indexed tables; the results of the table searches are combined in multiple phases to yield the final classification result. Algorithms using parallel search are very fast in term of classification speed while they require larger memory to store the crossproducting tables. In this paper, we choose HSM to be the basis of the integrated protocol processing scheme, because HSM
algorithm provides both fast search speed and modest memory usage compared to other algorithms.

4.2 Integrated Protocol Processing

Protocol processing in UTMs involves repeated application of several basic operations, mainly including packet classification for firewalling and protocol analysis for intrusion prevention. In order to
get the maximum gain in overall performance, we need to design the protocol processing applications in a very efficient manner. However, because each processing ngine need to manipulate multiple fields of packet header, they have to load the packet header from the off-chip memory to the local cache and then carry out modification and classification, and finally write the new header back into the external memory. Such read-modify-write operations are very time-consuming, and hence greatly impede the overall processing speed [16].This section, describes an integrated protocol processing algorithm that efficiently combines the protocol processing functionalities of different security

applications such as firewalling and IDS/IPS. Our algorithm handles multiple independent protocol processing rule sets by a compact data-structure using a HSM scheme.

4.1.1 Independent Protocol Processing

HSM performs parallel binary searches on multidimensional packet header, the result of the binary searches are combined in multiple *phases*. In the first phase, *F* fields of the packet header are **segmented** according to unique rule-projection intervals into multiple sub-spaces that are used to index into multiple memories. For example, in Figure 7, the two dimensional (4-bit source address and 4-bit destination address) search space is segmented in each of the dimension according to two ACL rules. In each segment, there is a unique set of rules denoted by different SA# and DA#. SA and DA are segmentation tables of source and destination IP addresses, respectively. Sub-spaces associated with the same rules are then **aggregated** and labeled with the same subspace ID. Example of space aggregation are shown in Figure 8, where AMT is the address mapping table, which aggregates the sub-spaces denoted by the <SA#,DA#> pairs to a sub-space containing only the rule with the highest priority. SP and DP are segmentation

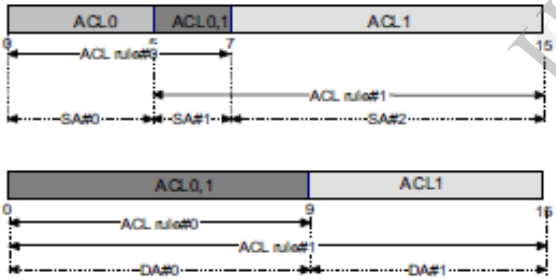

Figure 7.space segmentation

AMT	SA#0	SA#1	SA#2
DA#0	1 (ACL0)	2(ACL1)	3(ACL1)
DA#1	0(N/A)	3(ACL1)	3(ACL1)

Figure 8.space aggregation

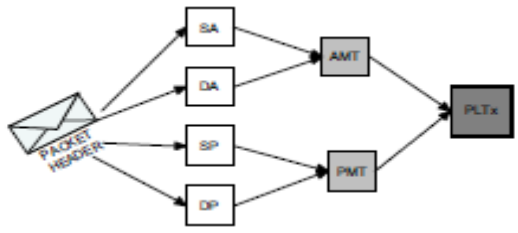

Figure 9.hierarchical space mapping

tables of source and destination ports, respectively. In subsequent phases, earlier sub-space IDs obtained from one dimensional segmentation are recursively crossproducted with the sub-spaces obtained from other dimensions. In Figure 9, AMT, PMT and PLT are hierarchical cross-producting tables for space aggregation. An incoming packet first gets indexes by looking up the four segmentation tables, and then using these indexes to trace the aggregation tables to finally yield classification results. In the final phase, the memory yields the action. More detail of HSM can be found in [16].

4.2. Integrated Protocol Processing

Note that the most time-consuming part of HSM is the binary search because of its $\Theta(\log N)$ complexity. Thus, if we have two independent rulesets, each having M and N rules respectively, the original HSM algorithm requires two independent searches for each ruleset and hence has $\Theta(\log(M) + \log(N))$ temporal complexity. To improve the performance of the binary search, the new algorithm should be able to handle multiple independent rulesets effectively. We propose to integrate multiple rulesets in the space segmentation step while remain independent tables for space aggregation. Specifically, there are two main steps:

Integrated space segmentation: Different from the original HSM, space segmentation for each dimension is carried out not only according to ACL rules but also to the IDS rules. Because 16 bits can support up to 32K rules, we use 16-bit index for ACL classification and 16-bit for IDS protocol analysis to store the segmentation number. The number of segments then increases from $\Theta(M)$ to $\Theta(M+N)$, where M is the number of ACL rules, and N is the number of IDS rules. Because the binary search is now taken on the integrated segmentations, the overall complexity becomes $\Theta(\log(M + N))$. Figure 10 illustrates how to implement the integrated segmentation scheme: The two-dimensional (4-bit source address and 4-bit destination address) search space is segmented in each of the dimension according to 2 ACL rules and 2 IDS rules. In each segment, there is two independent unique set of rules. SA#x and DA#x denote the unique set of ACL rules, while SA#y and DA#y denote the unique set of IDS rules.

Independent space aggregation: The space complexity of space aggregation tables are $\Theta(M^F)$ for M F-dimensional rules [16]. Thus, if we also integrate space aggregation tables, the overall space complexity will be $\Theta((M + N)^F)$. In practice, such an exponential increase of memory is not acceptable, so we retain the

IJERT

independency of each aggregation table and thus reduce the space complexity to $\Theta (M^F + N^F)$. Because the number of memory accesses to these aggregation tables is small compared to the binary search on segmentation tables, the independent tables for multiple rulesets will not significantly impact the overall search time. Figure 5 illustrates the implement action of the independent space aggregation: Use #x indexes to traverse AMTx, PMTx and PLTx table to get the classification results for the ACL rule set, which are commonly ACCEPT/DROP/EXEPTION. Use #y indexes to traverse AMTy, PMTy and PLTy table to get the finally IDS protocol analysis results, which may be a pointer to a specified rule set for deep payload inspection.

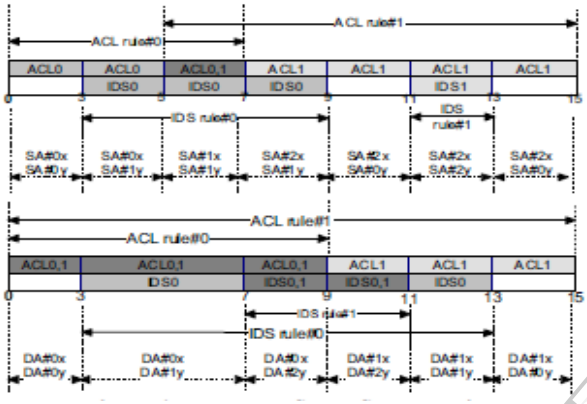

Figure 10. Integrated space segmentation

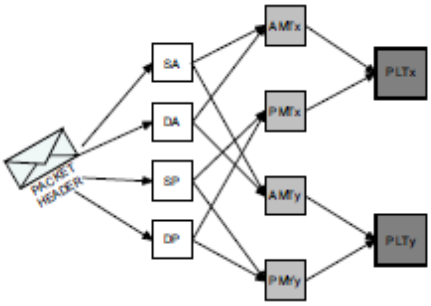

Figure 11. Independent space aggregation

5. Conclusion

To reach high-performance Unified Threat Management, security applications should be optimized at system-level rather than simply stringed together a number of security applications. An Integrated Protocol Processing scheme to resolve the key problems in existing UTMs: Redundant Packet Classification and Unnecessary Deep Inspection.

6. References

[1] Richard Alimi, Ye Wang, Y. Richard Yang, "Shadow Configuration as a Network Management Primitive", *ACM Sigcomm'08*, August 17–22, 2008, Seattle, Washington, USA, pp. 111-122.
[2] Donald Caldwell, SeungjoonLee , Yitzhak andelbaum, "Adaptive Parsing of Router Configuration Languages", *IEEE Internet Network Management Workshop, 2008*.
[3] Peter Drake, "USING SNMP TO MANAGE NETWORKS".
[4] WeldsonQueiroz de Lima, Rodrigo Sanger Alves, Ricardo LemosVianna, Maria Janilce ,etc."Evaluating the Performance of SNMP and Web Services Notifications", *NOMS2006*, 3-7 April 2006, pp.546 – 556.
[5] Ji Huang, Zhang Bin, Li Guohui, GaoXuesong, Li Yan, "Challenges to the New Network Management Protocol: NETCONF", *ETCS '09*, 7-8 March 2009.
[6] K.Claffy, H.-W.Braun, and G.Polyzos, "A parametrizable methodology for Internet traffic flow profiling", IEEE JSAC
[7] Traffic control and congestion control in IP based networks. ITU-T Y.1221, 2002.3
[8] Kwangjin Choi, Jun-kyun Choi, Sangyong Ha1,SeYun Ban. Content-based Pattern Matching for Classification of Network Application [J]. ICACT2006
[9] U.R.Naik and P.R.Chandra, "Designing Highperformance Networking Applications," Intel Press, 2004.
[10] Y. Qi and J. Li, "Performance Evaluation and Improvement of Algorithmic Approaches for Packet Classification," *Proc.Of International Conference on Networking and Services*, 2005.
[11] P. Gupta and N. McKeown, "Packet classification sing hierarchical intelligent cuttings," *Proc. of Hot Interconnects*, 1999.
[12] S. Singh, F. Baboescu, G. Varghese, and J. Wang, "Packet classification Using Multidimensional Cutting," *Proc. of ACM SIGCOMM*, 2003.
[13] P. Gupta and N. McKeown, "Packet classification on multiple fields," *Proc. ACM SIGCOMM*, 1999.
[14] B. Xu, D. Y. Jiang, and J. Li, "HSM: A Fast Packet Classification Algorithm", *Proc. of the 19th Advanced Information Networking and Applications (AINA 2005)*, 2005.

IJERT